Osvaldo Gervasi · Beniamino Murgante ·
Chiara Garau · Ana Maria A. C. Rocha ·
M. Noelia Faginas Lago · David Taniar ·
Yeliz Karaca (Eds.)

LNCS 16767

Computational Science and Its Applications – ICCSA 2026

26th International Conference
Braga, Portugal, June 30–July 3, 2026
Proceedings, Part I

Part I

Springer

Lecture Notes in Computer Science **16767**

Founding Editors

Gerhard Goos
Juris Hartmanis

Editorial Board Members

Elisa Bertino, *Purdue University, West Lafayette, IN, USA*
Wen Gao, *Peking University, Beijing, China*
Bernhard Steffen, *TU Dortmund University, Dortmund, Germany*
Moti Yung, *Columbia University, New York, NY, USA*

The series Lecture Notes in Computer Science (LNCS), including its subseries Lecture Notes in Artificial Intelligence (LNAI) and Lecture Notes in Bioinformatics (LNBI), has established itself as a medium for the publication of new developments in computer science and information technology research, teaching, and education.

LNCS enjoys close cooperation with the computer science R & D community, the series counts many renowned academics among its volume editors and paper authors, and collaborates with prestigious societies. Its mission is to serve this international community by providing an invaluable service, mainly focused on the publication of conference and workshop proceedings and postproceedings. LNCS commenced publication in 1973.

Osvaldo Gervasi · Beniamino Murgante ·
Chiara Garau · Ana Maria A. C. Rocha ·
M. Noelia Faginas Lago · David Taniar ·
Yeliz Karaca
Editors

Computational Science and Its Applications – ICCSA 2026

26th International Conference
Braga, Portugal, June 30 – July 3, 2026
Proceedings, Part I

Springer

Editors
Osvaldo Gervasi
University of Perugia
Perugia, Italy

Chiara Garau
University of Cagliari
Cagliari, Italy

M. Noelia Faginas Lago
University of Perugia
Perugia, Italy

Yeliz Karaca
University of Massachusetts
Amherst, MA, USA

Beniamino Murgante
University of Basilicata
Potenza, Italy

Ana Maria A. C. Rocha
University of Minho
Braga, Portugal

David Taniar
Monash University
Melbourne, VIC, Australia

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-032-30490-2 ISBN 978-3-032-30491-9 (eBook)
https://doi.org/10.1007/978-3-032-30491-9

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2026

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

If disposing of this product, please recycle the paper.

Preface

These 3 volumes (LNCS volumes 16767–16769) comprise the peer-reviewed papers from the 6 tracks of the 2026 International Conference on Computational Science and Its Applications (ICCSA 2026), held between June 30 and July 3, 2026, in Braga (Portugal). The peer-reviewed papers from the 57 Workshops are published in a separate set of 9 volumes (LNCS 16758–16766).

The conference was held in a hybrid format, with the vast majority of participants attending in person, hosted by the University of Minho in Braga, Portugal. We enabled virtual participation for those who did not attend the event in person due to logistical, political, or economic constraints by adopting a technological infrastructure based on open-source software (Jitsi + Riot) and a commercial Cloud infrastructure. ICCSA 2026 was organized by the University of Minho (Portugal), the University of Perugia (Italy), the University of Basilicata (Italy), the University of Cagliari (Italy), Monash University (Australia), and Kyushu Sangyo University (Japan).

With the 2026 edition, ICCSA celebrated its 26th anniversary, marking a new chapter in its longstanding history after its memorable 25th anniversary celebrated in Istanbul, Türkiye. Hosted in Braga, Portugal, the conference found a particularly meaningful setting in one of Europe's oldest and most vibrant academic cities, a place where historical depth, cultural continuity, and contemporary scientific exchange meet. Braga offers an inspiring European context for renewing the international and intercultural dialogue that has always characterized ICCSA, while giving new visibility to the role of computational science in addressing the complex challenges of cities, territories, technologies, and societies.

ICCSA 2026 marked another fruitful and thought-provoking academic event in the International Conferences on Computational Science and Its Applications (ICCSA) conference series, previously held in Istanbul, Türkiye (2025), Hanoi, Vietnam (2024), Athens, Greece (2023), Málaga, Spain (2022), Cagliari, Italy (hybrid with few participants in presence in 2021 and completely online in 2020), whilst earlier editions took place in Saint Petersburg, Russia (2019), Melbourne, Australia (2018), Trieste, Italy (2017), Beijing, China (2016), Banff, Canada (2015), Guimarães, Portugal (2014), Ho Chi Minh City, Vietnam (2013), Salvador, Brazil (2012), Santander, Spain (2011), Fukuoka, Japan (2010), Suwon, South Korea (2009), Perugia, Italy (2008), Kuala Lumpur, Malaysia (2007), Glasgow, UK (2006), Singapore (2005), Assisi, Italy (2004), Montreal, Canada (2003), and (as ICCS) Amsterdam, the Netherlands (2002) and San Francisco, USA (2001).

Computational Science constitutes the main pillar of most present research, industrial and commercial applications, and plays a unique role in exploiting innovative ICT technologies, and the ICCSA conference series has, accordingly, provided ample opportunities to researchers and industry practitioners to discuss new ideas, to share complex problems and their solutions, and to shape new trends in Computational Science. As the conference mirrors society from a scientific point of view, this year's undoubtedly

dominant theme was large language models, machine learning, and Artificial Intelligence (AI), and their applications across the most diverse technological, economic, and industrial fields, amongst others.

The ICCSA 2026 conference was structured into six general tracks covering computational science and its applications: Computational Methods, Algorithms and Scientific Applications; High-Performance Computing and Networks; Geometric Modelling, Graphics and Visualisation; Advanced and Emerging Applications; Information Systems and Technologies; and Urban and Regional Planning. In addition, the conference included 57 workshops, focusing on topical issues of utmost importance to science, technology, and society: from new computational approaches for earth science, to mathematical methods for image processing, new statistical and optimisation methods, several Artificial Intelligence approaches, sustainability issues, smart cities and related technologies, to name some.

In the Main Conference Proceedings, we accepted 119 full papers, 3 short papers and 2 Ph.D. Showcase papers on 405 submissions to the General Tracks of the Conference (with an acceptance rate of 29.9%). Reviews were single blind, and submissions received 2.6 reviews each, on average. In the Workshops proceedings, we accepted 321 full papers, 23 short papers and 3 Ph.D. Showcase papers from a total of 1184 submissions (Acceptance rate 29.9%). We would like to convey our sincere appreciation to the workshop chairs and co-chairs, and program committee members, for their diligent work, commitment, and dedication.

The success and consistent maintenance of the ICCSA conference series in general, and of ICCSA 2026 in particular, rely upon the support of many people: authors, presenters, participants, keynote speakers, workshop chairs, session chairs, Organizing Committee members, student volunteers, Program Committee members, Advisory Committee members, International Liaison chairs, reviewers, and other individuals in various roles. Thus, we take this opportunity to wholeheartedly thank each and every one.

We also wish to thank the publisher, Springer, for accepting the proceedings, for sponsoring part of the best paper awards, and for their kind assistance and cooperation during the editing process. We would cordially like to invite you to refer to the ICCSA website https://iccsa.org, where you can find the relevant details regarding this academic endeavour and our event.

June 2026

Chiara Garau
Ana Maria A. C. Rocha
Osvaldo Gervasi
Beniamino Murgante

Welcome Message from the Organizers

On behalf of the Local Organizing Committee of ICCSA 2026, it is our great pleasure to welcome you to the 26th International Conference on Computational Science and Its Applications, held from June 30 to July 3, 2026, at the University of Minho in Braga, Portugal.

We are deeply honoured by the trust placed in us by the ICCSA General Chairs in organizing another edition of this successful conference series. We hope that ICCSA 2026 provided an inspiring environment for scientific exchange, collaboration, and discussion among researchers from all over the world.

The conference took place at the University of Minho, one of the most prestigious higher education institutions in Portugal. Founded in 1973, the University of Minho has campuses in Braga and Guimarães and is widely recognized for its academic excellence, research activity, and strong international outlook.

Braga, one of the oldest cities in Portugal, is renowned for its historical heritage, cultural vitality, and welcoming atmosphere. Located in the heart of the Minho region, the city combines tradition with modernity, offering visitors renowned gastronomy, beautiful landscapes, and the unique character of *Vinho Verde* wine.

The scientific program of ICCSA 2026 included plenary lectures delivered by distinguished scientists, together with a broad range of workshops covering several areas of Computational Science and its Applications. We believe that the conference provided excellent opportunities to share ideas, discuss emerging challenges, and foster new collaborations.

In addition to the scientific sessions, participants had the opportunity to enjoy a rich social program. Planned activities included a Welcome Reception at Largo do Paço, the Rectorate of the University of Minho, located in the former Archbishop's Palace - one of the most emblematic historical buildings in the city center; a *Vinho Verde* wine tasting experience at Quinta d'Amares, where you can discover the art of winemaking; and a guided visit to the coastal city of Viana do Castelo, renowned for its maritime heritage, traditional architecture, and beautiful Atlantic scenery. The conference gala dinner took place at Colunata Eventos, located within the Bom Jesus do Monte sanctuary, a UNESCO World Heritage Site, where ICCSA participants enjoyed delicious recipes and a wonderful view over the city of Braga.

The success of ICCSA 2026 would not have been possible without the dedication and enthusiasm of many collaborators and volunteers, whose hard work and commitment greatly contributed to the organization of the conference. On behalf of the Local

Organizing Committee, it was our honour to warmly welcome you to the beautiful city of Braga and we hope you had an enjoyable and productive conference.

Sincerely yours,
On behalf of the Local Organizing Committee

Ana Maria A. C. Rocha

Organization

Honorary General Chairs

Bernady O. Apduhan	Kyushu Sangyo University, Japan
Kenneth C. J. Tan	Sardina Systems, UK

General Chairs

Ana Maria A. C. Rocha	University of Minho, Portugal
David Taniar	Monash University, Australia
Osvaldo Gervasi	University of Perugia, Italy

Program Committee Chairs

Beniamino Murgante	University of Basilicata, Italy
Chiara Garau	University of Cagliari, Italy

International Advisory Committee

Jemal Abawajy	Deakin University, Australia
Dharma P. Agarwal	University of Cincinnati, USA
Jagannath Aryal	University of Melbourne, Australia
Rajkumar Buyya	University of Melbourne, Australia
Claudia Bauzer Medeiros	University of Campinas, Brazil
Carl K. Chang	Iowa State University, USA
Elizabeth Chang	University of the Sunshine Coast, Australia
Manfred M. Fisher	Vienna University of Economics and Business, Austria
Pierre Frankhauser	University of Franche-Comté/CNRS, France
Marina L. Gavrilova	University of Calgary, Canada
Sumi Helal	University of Florida, USA & Lancaster University, UK
Bin Jiang	University of Gävle, Sweden
Yee Leung	Chinese University of Hong Kong, China
Biswajeet Pradhan	University of Technology Sydney, Australia
My Thai	University of Florida, USA

International Liaison Chairs

Giuseppe Borruso	University of Trieste, Italy
Elise De Donker	Western Michigan University, USA
Maria Noelia Faginas Lago	University of Perugia, Italy
Maria Irene Falcão	University of Minho, Portugal
Robert C. H. Hsu	Chung Hua University, Taiwan
Yeliz Karaca	UMass Chan Medical School, USA
Tae-Hoon Kim	Zhejiang University of Science and Technology, China
Vladimir Korkhov	Saint Petersburg University, Russia
Takashi Naka	Kyushu Sangyo University, Japan
Rafael D. C. Santos	National Institute for Space Research, Brazil
Maribel Yasmina Santos	University of Minho, Portugal
Anastasia Stratigea	National Technical University of Athens, Greece
Nguyen Canh Thai	Thuyloi University, Vietnam
Kh Md Nahiduzzaman	Mohammed VI Polytechnic University, Rabat, Morocco

Workshop and Session Organizing Chairs

Beniamino Murgante	University of Basilicata, Italy
Chiara Garau	University of Cagliari, Italy

Award Chair

Wenny Rahayu	La Trobe University, Australia

Publicity Committee Chairs

Elmer Dadios	De La Salle University, Philippines
Nataliia Kulabukhova	Saint Petersburg University, Russia
Daisuke Takahashi	Tsukuba University, Japan
Doan Quang Tu	Academy of Military Science and Technology, Vietnam
Shangwang Wang	Beijing University of Posts and Telecommunications, China

Local Organizing Committee Chairs

Maria Irene Falcão	University of Minho, Portugal
Ana Cristina Braga	University of Minho, Portugal
Rui José	University of Minho, Portugal
José Manuel Machado	University of Minho, Portugal
Jorge G. Rocha	University of Minho, Portugal

Technology Chair

Damiano Perri	University of Perugia, Italy

Program Committee

Vera Afreixo	University of Aveiro, Portugal
Vladimir Alarcon	Northern Gulf Institute, USA
Filipe Alvelos	University of Minho, Portugal
Debora Anelli	Polytechnic University of Bari, Italy
Muhammad Waqas Arshad	University of Bologna, Italy
Vittorio Astarita	University of Calabria, Italy
Hartmut Asche	Hasso-Plattner-Institut für Digital Engineering Ggmbh, Germany
Nizamettin Aydın	İstanbul Technical University, Türkiye
Ginevra Balletto	University of Cagliari, Italy
Nadia Balucani	University of Perugia, Italy
Socrates Basbas	Aristotle University of Thessaloniki, Greece
David Berti	ART SpA, Italy
Sandro Bimonte	CEMAGREF, TSCF, France
Ana Cristina Braga	University of Minho, Portugal
Tiziana Campisi	University of Enna KORE, Italy
Alessandra Capolupo	Polytechnic of Bari, Italy
Leocadio G. Casado	University of Almeria, Spain
Ilaria Catapano	National Research Council, IREA, Italy
Maria Cerreta	University of Naples Federico II, Italy
Rachel Chien-Sing Lee	Sunway University, Malaysia
Birol Ciloglugil	Ege University, Turkey
Florbela Maria da Cruz Domingues Correia	Polytechnic Institute of Viana do Castelo, Portugal
Alessandro Costantini	INFN, Italy
Roberto De Lotto	University of Pavia, Italy
Luiza De Macedo Mourelle	State University of Rio De Janeiro, Brazil

Marcelo De Paiva Guimaraes	Federal University of São Paulo, Brazil
Marco De Santis	University of Salerno, Italy
Frank Devai	London South Bank University, UK
Joana Matos Dias	University of Coimbra, Portugal
Aziz Dursun	Virginia Tech University, USA
Laila El Ghandour	Heriot Watt University, UK
Maria Irene Falcão	University of Minho, Portugal
Florbela P. Fernandes	Polytechnic Institute of Bragança, Portugal
Paula Odete Fernandes	Polytechnic Institute of Bragança, Portugal
Adelaide de Fátima Baptista Valente Freitas	University of Aveiro, Portugal
Andreas Fricke	University of Potsdam, Germany
Raffaele Garrisi	Centro Operativo per la Sicurezza Cibernetica, Italy
Ivan Gerace	University of Perugia, Italy
Maria Giaoutzi	National Technical University of Athens, Greece
Salvatore Giuffrida	University of Catania, Italy
Teresa Guarda	Universidad Estatal Península de Santa Elena, Ecuador
Gabriel Guerrero Contreras	University of Cádiz, Spain
Malgorzata Hanzl	Technical University of Lodz, Poland
Maulana Adhinugraha Kiki	Telkom University, Indonesia
JungYoon Kim	Gachon University, South Korea
Clement Ho Cheung Leung	Chinese University of Hong Kong, China
Andrea Lombardi	University of Perugia, Italy
Marcos Mandado Alonso	University of Vigo, Spain
Ernesto Marcheggiani	Katholieke Universiteit Leuven, Belgium
Barbara Martelli	INFN, Italy
Antonino Marvuglia	Luxembourg Institute of Science and Technology, Luxembourg
Michele Mastroianni	University of Salerno, Italy
Hideo Matsufuru	High Energy Accelerator Research Organization, Japan
Fernando Miranda	Universidade do Minho, Portugal
Giuseppe Modica	University of Reggio Calabria, Italy
Majaz Moonis	UMass Chan Medical School, USA
Leila Moradi	University of Salerno, Italy
Nadia Nedjah	State University of Rio de Janeiro, Brazil
Paolo Nesi	University of Florence, Italy
Suzan Obaiys	University of Malaya, Malaysia
Marcin Paprzycki	Polish Academy of Sciences, Poland
Eric Pardede	La Trobe University, Australia

Ana Isabel Pereira	Polytechnic Institute of Bragança, Portugal
Damiano Perri	University of Perugia, Italy
Massimiliano Petri	University of Pisa, Italy
Telmo Pinto	University of Coimbra, Portugal
Maurizio Pollino	ENEA, Italy
Marcos Quiles	Federal University of São Paulo, Brazil
Nguyen Huu Quynh	Thuyloi University, Vietnam
Albert Rimola	Universitat Autònoma de Barcelona, Spain
Humberto Rocha	University of Coimbra, Portugal
Marzio Rosi	University of Perugia, Italy
Lucia Saganeiti	University of L'Aquila, Italy
Francesco Scorza	University of Basilicata, Italy
Marco Paulo Seabra dos Reis	University of Coimbra, Portugal
Jie Shen	University of Michigan, USA
Bruno Sousa	University of Coimbra, Portugal
Francesco Tajani	Sapienza University of Rome, Italy
Daniela Tavano	University of Calabria, Italy
Ho Sy Tam	Thuyloi University, Vietnam
Deodato Tapete	Italian Space Agency ASI, Italy
Rodrigo Tapia-McClung	Centro de Investigación en Ciencias de Información Geoespacial, Mexico
Tarek Teba	University of Portsmouth, UK
Ana Paula Teixeira	Universidade do Minho, Portugal
Luis Teodoro	University of Oslo, Norway
Yiota Theodora	National Technical University of Athens, Greece
Elena Todella	Polytechnic University of Turin, Italy
Graca Tomaz	Instituto Politécnico da Guarda, Portugal
Anna Tonazzini	National Research Council, Italy
Francesca Torrieri	Polytechnic University of Milan, Italy
Vincenza Torrisi	University of Catania, Italy
Vincenzo Totaro	Polytechnic University of Bari, Italy
Arianna Travaglini	University of Basilicata, Italy
Stefanos Tsigdinos	National Technical University of Athens, Greece
Aime Tsinda	University of Rwanda, Rwanda
Elen Twrdy	University of Ljubljana, Slovenia
Doan Quang Tu	Academy of Military Science and Technology, Vietnam
Le Quang Tuan	Thuyloi University, Vietnam
Toshihiro Uchibayashi	Kyushu University, Japan
Fukuko Yuasa	High Energy Accelerator Research Organization, Japan
Ljiljana Zivkovic	Republic Geodetic Authority, Serbia

Workshop on Advancements in Applied Machine-Learning and Data Analytics (AAMDA 2026)

Workshop Organizers:

Alessandro Costantini	INFN, Italy
Elisabetta Ronchieri	INFN CNAF, Italy

Artificial Intelligence Tools In Managing Future Communities (AiMFutCOM 2026)

Workshop Organizers:

Marco Vona	University of Basilicata, Italy
Francesco Scorza	University of Basilicata, Italy
Donatello Cardone	University of Basilicata, Italy
Maurizio Pollino	ENEA, Italy
Gregorio DAgostino	ENEA, Italy

Advanced and Innovative Web Apps 2026 (AIWA 2026)

Workshop Organizers:

Damiano Perri	University of Perugia, Italy
Osvaldo Gervasi	University of Perugia, Italy
Luciano Alessandro Ipsaro Palesi	University of Florence, Italy

Advanced Mathematics and Computing Models in Complex Data-Intensive Computational Systems (AMCM 2026)

Workshop Organizers:

Yeliz Karaca	UMass Chan Medical School and Massachusetts Institute of Technology, USA
Dumitru Baleanu	Lebanese American University, Lebanon
Osvaldo Gervasi	University of Perugia, Italy
Yudong Zhang	University of Leicester, UK
Majaz Moonis	UMass Chan Medical School, USA

Advanced Numerical Approaches for Assessment and Design of No-Tension Masonry Structures (ANAMS 2026)

Workshop Organizers:

Antonino Iannuzzo	Universitá degli studi del Sannio, Italy
Natalia Pingaro	Politecnico di Milano, Italy
Andrea Montanino	International Centre for Numerical Methods in Engineering, Spain
Elham Mousavian	University of Edinburgh, UK
Stefano Galassi	University of Florence, Italy

Artificialization of land, Urban Transformation, and Territorial Resilience in a Changing Climate (AUTTREC 2026)

Workshop Organizers:

Celestina Fazia	Università di Enna Kore, Italy
Kh Md Nahiduzzaman	Mohammed VI Polytechnic University, Morocco
Hassan Radoine	Mohammed VI Polytechnic University, Morocco
Cezar Morar	University of Oradea, Romania
Antonio García Martínez	Universidad de Sevilla, Spain
Clara Stella Vicari Aversa	Università degli Studi Mediterranea di Reggio Calabria, Italy

Blockchain and Distributed Ledgers: Technologies and Applications (BDLTA 2026)

Workshop Organizers:

Vladimir Korkhov	Saint Petersburg State University, Russia
Elena Stankova	Saint Petersburg State University, Russia
Nataliia Kulabukhova	AutoTech, Russia

Bio and Neuro inspired Computing and Applications (BIONCA 2026)

Workshop Organizers:

Nadia Nedjah	State University of Rio de Janeiro, Brazil
Luiza de Macedo Mourelle	State University de Rio de Janeiro, Brazil

Computational and Applied Mathematics (CAM 2026)

Workshop Organizers:

Maria Irene Falcao	University of Minho, Portugal
Fernando Miranda	Universidade do Minho, Portugal

Computational and Applied Statistics for Data and AI (CAS 2026)

Workshop Organizers:

Ana Cristina Braga	University of Minho, Portugal
Ines Sena	University of Minho, Portugal
Joao Mendes	Polytechnic Institute of Bragança, Portugal

Circular Economy and Territorial Competitiveness (CETC 2026)

Workshop Organizers:

Gerardo Carpentieri	University of Naples Federico II, Italy
Carmen Guida	Università degli Studi di Napoli Federico II, Italy
Floriana Zucaro	University of Naples Federico II, Italy
Zhadyra Konurbaeva	D. Serikbayev East Kazakhstan Technical University, Kazakhstan
Saule Kumargazhanova	D.Serikbayev East Kazakhstan Technical University, Kazakhstan

Circular Ecological Heritage Communities: Regeneration and Evaluation Strategies for Historic Villages, Landscapes and Rural Territories (CHEC 2026)

Workshop Organizers:

Mariarosaria Angrisano	Pegaso University, Italy
Antonia Gravagnuolo	CNR Institute of Heritage Science, Italy
Francesca Buglione	CNR ISPC, Italy
Piera Della Morte	Pegaso Telematic University, Italy

Cyber Intelligence and Applications (CIA 2026)

Workshop Organizers:

Gianni DAngelo	University of Salerno, Italy
Francesco Palmieri	University of Salerno, Italy
Massimo Ficco	University of Salerno, Italy

Computational Methods, Statistics and Industrial Mathematics (CMSIM 2026)

Workshop Organizers:

Maria Filomena Teodoro	CINAV, Escola Naval, Instituto Universitário Militar, Portugal
Paula Simões	Academia Militar, Portugal
Teresa A. Oliveira	University of Lisbon and Universidade Aberta, Portugal

Computational Optimization and Applications (COA 2026)

Workshop Organizers:

Ana Maria A. C. Rocha	University of Minho, Portugal
Humberto Rocha	University of Coimbra, Portugal

Computational Methods in Advanced Networking Technologies (CoMANT 2026)

Workshop Organizers:

Eslam Farsimadan	University of Salerno, Italy
Leila Moradi	University of Salerno, Italy
Francesco Palmieri	University of Salerno, Italy

Computational Astrochemistry 2026 (CompAstro 2026)

Workshop Organizers:

Marzio Rosi	University of Perugia, Italy
Daniela Ascenzi	Università degli Studi di Trento, Italy
Nadia Balucani	University of Perugia, Italy
Stefano Falcinelli	University of Perugia, Italy

Computational Methods for Porous Materials (CompPor 2026)

Workshop Organizers:

Vadim Lisitsa	Institute of Petroleum Geology and Geophysics SB RAS, Russia
Evgeniy Romenski	Sobolev Institute of Mathematics, Russia

Workshop on Computational Science and HPC (CSHPC 2026)

Workshop Organizers:

Elise de Doncker	Western Michigan University, USA
Hideo Matsufuru	High Energy Accelerator Research Organization, Japan
Robert Makin	Western Michigan University, USA

Cities, Technologies and Planning (CTP 2026)

Workshop Organizers:

Giuseppe Borruso	University of Trieste, Italy
Beniamino Murgante	University of Basilicata, Italy
Malgorzata Hanzl	Lodz University of Technology, Poland
Anastasia Stratigea	National Technical University of Athens, Greece
Ljiljana Zivkovic	Republic Geodetic Authority, Serbia
Ginevra Balletto	University of Cagliari, Italy

Evaluating Inner Areas Potentials (EIAP 2026)

Workshop Organizers:

Diana Rolando	Politecnico di Torino, Italy
Manuela Rebaudengo	Politecnico di Torino, Italy
Alice Barreca	Politecnico di Torino, Italy
Giorgia Malavasi	Politecnico di Torino, Italy

sustainablE mobiLity and Last mIle lOgisTic in Urban Context (ELLIOT 2026)

Workshop Organizers:

Tiziana Campisi	University of Enna Kore, Italy
Socrates Basbas	Aristotle University of Thessaloniki, Greece
Florin Rusca	National University of Science and Technology Politehnica Bucharest, Romania
Grigorios Fountas	Aristotle University of Thessaloniki, Greece
Aleksandra Deluka-Tibljas	University of Rijeka, Croatia

Econometrics and Multidimensional Evaluation in Urban Environment (EMEUE 2026)

Workshop Organizers:

Maria Cerreta	University of Naples Federico II, Italy
Giuseppe Ciciriello	University of Naples Federico II, Italy
Simona Panaro	University of Sussex Business School, UK
Piero Zizzania	Università of Naples Federico II, Italy
Pierluigi Morano	Polytechnic University of Bari, Italy
F rancesco Tajani	Sapienza University of Rome, Italy

Ecosystem Services in Spacial Planning for Climate Neutral Urban and Rural Areas (ESSP 2026)

Workshop Organizers:

Sabrina Lai	University of Cagliari. Italy
Carmela Gargiulo	Università Federico II di Napoli, Italy
Beniamino Murgante	University of Basilicata, Italy
Francesco Scorza	University of Basilicata, Italy
Corrado Zoppi	University of Cagliari, Italy
Floriana Zucaro	University of Naples Federico II, Italy

The 16th International Workshop on Future Information System Technologies and Applications (FiSTA 2026)

Workshop Organizers:

Bernady O. Apduhan	Kyushu Sangyo University, Japan
Rafael Santos	National Institute for Space Research (INPE), Brazil

Geographical Analysis, Urban Modeling, Spatial Statistics (Geog-And-Mod 2026)

Workshop Organizers:

Giuseppe Borruso	University of Trieste, Italy
Beniamino Murgante	University of Basilicata, Italy
Hartmut Asche	Hasso Plattner Institute for Digital Engineering, Germany
Rodrigo Tapia McClung	Centro de Investigación en Ciencias de Información Geoespacial, Mexico
Andreas Fricke	Hasso Plattner Institute for Digital Engineering, University of Potsdam, Germany

Generative and Neuro-Symbolic Artificial Intelligence for Real-World Systems (GNS-AI 2026)

Workshop Organizers:

Hugo Peixoto	University of Minho, Portugal
Manuel Rodrigues	University of Minho, Portugal
Regina Sousa	University of Minho, Portugal

Geomatics for Resource Monitoring and Management (GRMM 2026)

Workshop Organizers:

Alberico Sonnessa	Politecnico di Bari, Italy
Eufemia Tarantino	Politecnico di Bari, Italy
Marco Locurcio	Polytechnic University of Bari, Italy

International Workshop on Information and Knowledge in the Internet of Things (IKIT 2026)

Workshop Organizers:

Teresa Guarda	Universidad Estatal Península de Santa Elena, Ecuador

Intelligent Sustainable Development: from Models to Real-World Impact (ISD 2026)

Workshop Organizers:

Diego Brandão (	Federal Center for Technological Education of Rio de Janeiro, Brazil
Angela Ferreira	Polytechnic University of Bragança, Portugal
Eduardo Bezerra	Federal Center for Technological Education of Rio de Janeiro, Brazil
Raphael Guerra	Universidade Federal Fluminense, Brazil
Douglas Cardoso	University of Coimbra, Portugal
Rui Pedro Lopes	Polytechnic Institute of Bragança, Portugal

Multidimensional Evolutionary Evaluations for Transformative Approaches (MEETA 2026)

Workshop Organizers:

Maria Cerreta	University of Naples Federico II, Italy
Giuliano Poli	Università Federico II di Napoli, Italy
Gaia Daldanise	CNR Institute for Research on Innovation and Services for Development, Italy
Ludovica La Rocca	University of Federico II of Naples, Italy
Maria Somma	University of Naples Federico II, Italy
Sabrina Sacco	University of Naples Federico II, Italy

Building Multi-dimensional Models for Assessing Complex Environmental Systems (MES 2026)

Workshop Organizers:

Vanessa Assumma	University of Bologna, Italy
Caterina Caprioli	Politecnico di Torino, Italy
Giulia Datola	Politecnico di Milano, Italy

Federico DellAnna	Politecnico di Torino, Italy
Marta DellOvo	Politecnico di Milano, Italy
Marco Rossitti	Politecnico di Milano, Italy

Computational Mechanics, Intelligence, and Data (MIND 2026)

Workshop Organizers:

Lelio Campanile	Università degli Studi della Campania Luigi Vanvitelli, Italy
Renato Zona	Università della Campania Vanvitelli, Italy

Modelling Liveable Cities: Techniques, Methods, Challenges, and Perspectives Behind the X-Minute City (MLC 2026)

Workshop Organizers:

Federico Mara	University of Pisa, Italy
Valerio Cutini	University of Pisa, Italy

Mathematical Methods for Image Processing and Understanding 2026 (MMIPU 2026)

Workshop Organizers:

Ivan Gerace	Università degli Studi di Perugia, Italy
Gianluca Vinti	Università degli Studi di Perugia, Italy
Arianna Travaglini	Lucian Blaga University of Sibiu, Romania

MOdels and Indicators for Assessing and Measuring the Urban Settlement deVElopment in the View of NET ZERO by 2050 (MOVEto0 2026)

Workshop Organizers:

Lorena Fiorini	University of L'Aquila, Italy
Lucia Saganeiti	Institute of Methodologies for Environmental Analysis (IMAA) CNR, Italy
Angela Pilogallo	Institute of Methodologies for Environmental Analysis (IMAA), Italy
Francesco Zullo	University of L'Aquila, Italy
Alessandro Marucci	University of L'Aquila, Italy

Numerical Aspects in Machine Learning for Dynamical Systems (NAMLYS 2026)

Workshop Organizers:

Pasquale De Luca	Parthenope University of Naples, Italy
Livia Marcellino	University of Naples Parthenope, Italy
Valeria Mele	University of Naples Federico II, Italy

Numerical Methods and Machine Learning in Science and Engineering (NMLSE 2026)

Workshop Organizers:

Cecília Coelho	Helmut Schmidt University, Germany
M. Fernanda P. Costa	University of Minho, Portugal
Luís Ferrás	University of Minho, Portugal
Eliana Costa e Silva	Polytechnic of Porto, Portugal

6th Workshop on Privacy in the Cloud/Edge/IoT World 2026 (PCEIoT 2026)

Workshop Organizers:

Lelio Campanile	Università degli Studi della Campania Luigi Vanvitelli, Italy
Mauro Iacono	Università degli Studi della Campania Luigi Vanvitelli, Italy
Michele Mastroianni	University of Foggia, Italy

Preserving Our Past: Spatial and Remote Sensing Technologies for Cultural Heritage in a Changing Climate (POP 2026)

Workshop Organizers:

Maria Danese	Institute of Heritage Science of the National Research Council of Italy (CNR ISPC), Italy
Rosa Lasaponara	Institute of Methodologies for Environmental Analysis (CNR IMAA), Italy
Nicola Masini	Institute of Heritage Science of the National Research Council of Italy (CNR ISPC), Italy

Scientific Computing Infrastructure (SCI 2026)

Workshop Organizers:

Vladimir Korkhov	Saint Petersburg State University, Russia
Elena Stankova	Saint Petersburg State University, Russia

Socio-Economic and Environmental Models for Land Use Management (SEMLUM 2026)

Workshop Organizers:

Debora Anelli	Sapienza University of Rome, Italy
Pierluigi Morano	Polytechnic University of Bari, Italy
Francesco Tajani	Sapienza University of Rome, Italy
Felicia Di Liddo	Polytechnic University of Bari, Italy

Ports and Logistics of the Future - Smartness and Sustainability (SmartPorts 2026)

Workshop Organizers:

Andrea Gallo	Università degli Studi di Trieste, Italy
Giuseppe Borruso	University of Trieste, Italy
Gianfranco Fancello	University of Cagliari, Italy
Veronica Camerada	Università degli studi di Sassari, Italy
Clara Di Fazio	University of Trieste, Italy

Sustainable Evolution of Long-Distance frEight and paSsenger Transport (SOLIDEST 2026)

Workshop Organizers:

Antonio Russo	University of Enna Kore, Italy
Giuseppe Musolino	Università Mediterranea di Reggio Calabria, Italy
Socrates Basbas	Aristotle University of Thessaloniki, Greece
Oana Dinu	National University of Science and Technology Politehnica Bucharest, Romania
Sanja Šurdonja	University of Rijeka, Croatia
Elen Twrdy	University of Ljubljana, Slovenia

Sustainability Performance Assessment: Models, Approaches, and Applications Toward Interdisciplinary and Integrated Solutions. (SPA 2026)

Workshop Organizers:

Francesco Scorza	University of Basilicata, Italy
Sabrina Lai	University of Cagliari, Italy
Jolanta Dvarioniene	Kaunas University of Technology, Lithuania
Michele Campagna	University of Cagliari, Italy

Specifics of Smart Cities Development in Europe (SPEED 2026)

Workshop Organizers:

Chiara Garau	University of Cagliari, Italy
Katarína Vitálišová	Matej Bel University, Slovak Republic
Marco Fanfani	University of Florence, Italy
Anna Vaňová	Matej Bel University, Slovak Republic
Kamila Borsekova	Matej Bel University, Slovak Republic
Paola Zamperlin	University of Pisa, Italy

Smart, Safe, and Health Cities (SSHC 2026)

Workshop Organizers:

Chiara Garau	University of Cagliari, Italy
Gerardo Carpentieri	University of Naples Federico II, Italy
Carmen Guida	Università degli Studi di Napoli Federico II, Italy
Martina Carra	University of Brescia, Italy
Vincenza Torrisi	University of Catania, Italy
Silvia Rossetti	Università degli Studi di Parma, Italy

Smart and Sustainable Island Communities (SSIC 2026)

Workshop Organizers:

Chiara Garau	University of Cagliari, Italy
Anastasia Stratigea	National Technical University of Athens, Greece
Yiota Theodora	National Technical University of Athens, Greece
Giovanna Concu	University of Cagliari, Italy
Tiziana Campisi	University of Enna Kore, Italy
Tanja Congiu	University of Sassari, Italy

Sustainable Tourism Economy, Silk Road, Reuse and Enhancement of Cultural Heritage (STESilkRoad 2026)

Workshop Organizers:

Celestina Fazia	Università di Enna Kore, Italy
Alexandru Ilies	University of Oradea, Romania
Dorina Camelia Ilies	University of Oradea, Romania
Cristina Natoli	Ministero dell'Ambiente e della Sicurezza Energetica, Italy
Bahodirhon Safarov	Samarkand Branch of Tashkent State University of Economics, Uzbekistan
Clara Stella Vicari Aversa	Università degli Studi Mediterranea di Reggio Calabria, Italy

Theoretical and Computational Chemistry and its Applications (TCCM 2026)

Workshop Organizers:

Noelia Faginas Lago	Università di Perugia, Italy
Andrea Lombardi	Università di Perugia, Italy
Marcos Mandado Alonso	University of Vigo, Spain

Transport Infrastructures for Smart Cities (TISC 2026)

Workshop Organizers:

Francesca Maltinti	University of Cagliari, Italy
Mauro Coni	University of Cagliari, Italy
Chiara Garau	University of Cagliari, Italy
Benedetto Barabino	University of Brescia, Italy
Nicoletta Rassu	Università di Cagliari, Italy
James Rombi	University of Cagliari, Italy

Transforming Urban Analytics: The Impact of Crowdsourced Mapping and Advanced AI Techniques on Future Cities (Tr-UrbAna 2026)

Workshop Organizers:

Ayse Giz Gulnerman Gengec	Ankara Hacı Bayram Veli University, Turkey
Müslüm Hacar	Yildiz Technical University, Turkey

From Structural to TRAnsformative-Change of City Environment: Challenges and Solutions and Perspectives (TRACE 2026)

Workshop Organizers:

Maria Rosaria Guarini	Sapienza University of Rome, Italy
Pierluigi Morano	Polytechnic University of Bari, Italy
Francesco Sica	Sapienza University of Rome, Italy
Alejandro Segura	Universidad Politécnica de Madrid, Spain
Yilsy María Nuñez Guerrero	Universidad Politécnica de Madrid, Spain

Urban Artificial Intelligence for Modelling, Analysis and Planning (Urban plAnnIng 2026)

Workshop Organizers:

Chiara Garau	University of Cagliari, Italy
Romano Fistola	University of Naples Federico II, Italy
Beniamino Murgante	University of Basilicata, Italy
Dessislava Petrova Antonova	Sofia University, GATE Institute, Bulgaria
Gianni Pantaleo	University of Florence, Italy
Rosa Anna La Rocca	University of Naples Federico II, Italy

Urban Space Accessibility and Mobilities (USAM 2026)

Workshop Organizers:

Chiara Garau	University of Cagliari, Italy
Alessandro Plaisant	University of Sassari, Italy
Barbara Caselli	Università degli Studi di Parma, Italy
Mauro D. Apuzzo	University of Cassino and Southern Lazio, Italy
Francesco Pinna	University of Cagliari, Italy
Silvio Nocera	IUAV University of Venice, Italy

Virtual Reality, Augmented Reality and Applications (VRA 2026)

Workshop Organizers:

Damiano Perri	University of Perugia, Italy
Osvaldo Gervasi	University of Perugia, Italy
Paolo Nesi	University of Florence, Italy

Workshop on Advanced and Computational Methods for Earth Science Applications (WACM4ES 2026)

Workshop Organizers:

Luca Piroddi	University of Cagliari, Italy
Patrizia Capizzi	University of Palermo, Italy
Marilena Cozzolino	Università del Molise, Italy
Giuseppina Vacca	University of Cagliari, Italy
Luigi Capozzoli	Hydrogeosite Laboratory of CNR-IMAA, Italy
Nikos Papadopoulos	Foundation for Research and Technology Hellas, Greece

Sponsoring Organizations

ICCSA 2026 would not have been possible without the tremendous support of many organizations and institutions, for which all organizers and participants of ICCSA 2026 express their sincere gratitude:

	University of Minho, Portugal (https://www.uminho.pt/)

(*continued*)

(*continued*)

	University of Perugia, Italy (https://www.unipg.it)
	University of Basilicata, Italy (http://www.unibas.it)
	University of Cagliari, Italy (https://unica.it/)
	Monash University, Australia (https://www.monash.edu/)
	Kyushu Sangyo University, Japan (https://www.kyusan-u.ac.jp/)

Plenary Lectures

Digital Twins for Cities – The Rise of the Socio-Technical System

Claudia van der Laag

Professor of Urban Analytics, Department of Built Environment, Oslo Metropolitan University, Oslo

Abstract. Cities are entering a new era in which digital twins are no longer simply technical replicas of urban systems, but evolving socio-technical infrastructures that shape how cities are planned, governed, and experienced. Across the world, urban digital twins are being adopted to address some of the most pressing challenges of our time—climate adaptation, urban resilience, mobility transitions, decarbonisation, infrastructure management, and inclusive governance. Yet as these technologies become increasingly sophisticated, a fundamental question emerges: can digital twins truly support more democratic and human-centred cities, or will they reinforce existing structures of power and expertise? In this keynote, Claudia van der Laag, explores the transformation of urban digital twins from data-driven modelling tools into collaborative platforms for urban decision-making and societal change. It argues that the future of digital twins lies not only in advances in artificial intelligence, real-time data integration, simulation, and predictive analytics, but equally in the ability to connect technological innovation with governance, participation, and human agency. Drawing on emerging international research and practical case studies, the keynote examines the complex ecosystem surrounding urban digital twins, bringing together planners, policymakers, researchers, technologists, and communities. It highlights how digital twins increasingly function as meeting points between multiple forms of knowledge—technical, political, social, and experiential—and why interdisciplinary collaboration is essential for their success. The talk will critically reflect on the tensions between technological ambition and civic inclusion, exploring themes such as transparency, interoperability, participation, and trust. It will also discuss how digital twins can become powerful environments for scenario exploration, urban experimentation, and collective learning, enabling cities to better navigate uncertainty and complexity in an age of rapid urban transformation. Ultimately, this keynote presents a vision of digital twins not as neutral technological systems, but as living socio-technical infrastructures capable of reshaping urban governance itself. The challenge ahead is therefore not only how accurately cities can be modelled, but how inclusively they can be imagined, negotiated, and governed through these emerging digital environments.

Short Bio – Claudia van der Laag (also known as Claudia Yamu) is Professor of Urban Analytics at Oslo Metropolitan University. Her work advances urban analytics and spatial modelling through an integrated multi-method approach, informing sustainable urban planning and policy. She is widely recognised for bridging theory and practice and received the prestigious Michael Breheny Prize for her contributions to multi-scale

modelling. Van der Laag is co-author of the widely recognised textbook ‘Introduction to Space Syntax in Urban Studies’. She also serves on the editorial advisory board of Springer’s The Urban Book Series and the journal Environment and Planning B: Urban Analytics and City Science.

Prescriptive AI in Practice: Bridging Optimisation, Generative AI and Agentic Decision-Making

Bernardo Almada-Lobo

Universidade do Porto, Faculdade de Engenharia, INESC TEC, Portugal

Abstract. Prescriptive AI, viewed through the lens of mathematical optimisation and operations research, addresses complex business decision-making problems involving limited resources, multiple decision options and competing objectives. This talk considers the full decision pipeline: problem framing, model formulation, computational solution, implementation, monitoring and organisational adoption. The practical success of prescriptive AI depends less on algorithmic sophistication alone than on the quality of the translation between real-world decision contexts and formal optimisation models. In practice, this translation remains a major bottleneck. Objectives are often ambiguous, constraints are incomplete or tacit, data may be fragmented, and the operational consequences of a recommendation may be poorly understood. As a result, optimisation projects can fail even when the underlying mathematical model is technically sound. The talk will address the persistent gap between optimisation theory and deployment practice. While optimisation has generated substantial value in domains such as supply chain planning, logistics, production, revenue management, workforce planning and inventory control, many initiatives still face a significant risk of under-delivery. Common failure modes include incorrect problem framing, weak data readiness, brittle assumptions, infeasible or non-actionable recommendations, limited explainability, insufficient ownership by business users and inadequate post-deployment monitoring. Crossing this gap requires moving from one-off model building to an operational lifecycle in which models are stress-tested, embedded in workflows, monitored for drift and revised as the business environment changes. Recent developments in generative and agentic AI create new opportunities to reduce these barriers. Generative AI can support the elicitation of decision-maker preferences, the structuring of unstructured operational knowledge, the drafting of candidate formulations and the explanation of model outputs in business language. Agentic AI can extend this role by orchestrating multi-step workflows that involve model formulation, solver-code generation, execution, debugging, infeasibility diagnosis, sensitivity analysis and human review. However, these capabilities should not be interpreted as replacing optimisation expertise. Rather, they create a new interface between human judgement, analytical modelling and computational decision support. This talk will discuss a cross-fertilisation agenda between optimisation, generative AI and agentic AI, with a focus on how these technologies should be calibrated to the decision at hand.

Short Bio – Full Professor of Industrial Engineering and Management at the Faculty of Engineering of University of Porto (FEUP). Co-founder and Partner at LTPlabs, a Business AI company. Member of the Board of Trustees of the Belmiro de Azevedo

Foundation. Member of the Academy of Engineering. Former Board Member of INESC TEC and Researcher at the MIT Sloan School of Management. Holds a BEng, PhD and Habilitation in Industrial Engineering and Management from FEUP. Completed the Advanced Management Programme at INSEAD, and the Mergers and Acquisitions at London Business School. Certified Analytics Professional (CAP) by INFORMS. His area of activity is Business Analytics and Artificial Intelligence. He researches, develops and implements analytical models and methods to support decision-making, solving management problems across various domains—manufacturing, consumer goods, retail and transport—with a particular focus on Operations Management and the Operations–Marketing interface. He has coordinated and participated in more than 300 development, consultancy and research projects funded by private organisations, science agencies from several countries, the European Commission and regional bodies. He has published more than 100 scientific articles and is co-author of the book Analytics Sandwich: Bringing People and Artificial Intelligence Together to Unlock Business Value. He has supervised more than 20 doctoral theses.

Net-Zero Communities Through Life Cycle Thinking: Canadian Lessons and Global Implications

Kasun Hewage

Professor of Engineering, School of Engineering (Okanagan Campus), The University of British Columbia (UBC), Canada, FortisBC Smart Energy Chair, Associate Director, Clean Energy Research Centre (CERC)

Abstract. This keynote presentation discusses how life cycle thinking supports the transformation of communities toward sustainable, resilient, and net-zero futures. Drawing on Canadian research, practice, and policy experience, the presentation highlights how integrated life cycle approaches can move beyond conventional carbon accounting to inform practical decision-making across buildings, infrastructure, energy systems, waste management, water systems, and industrial processes [1-8]. The keynote features research conducted at the Life Cycle Management Laboratory (LCML) at the University of British Columbia, Canada, an internationally recognized research hub led by Prof. Kasun Hewage. LCML brings together engineers, scientists, policy researchers, postdoctoral fellows, visiting scholars, and graduate researchers to translate data-intensive sustainability research into actionable climate solutions. The laboratory's work is organized around six interconnected research themes: energy systems, urban development, water systems, waste management, construction and asset management, and industrial processes and products. Together, these themes provide an integrated platform for advancing life cycle–based pathways to net-zero communities. Supported by strong partnerships with government, industry, and community stakeholders, LCML has contributed to a substantial body of applied sustainability research, including more than 400 journal articles, 100 conference papers, and numerous technical reports that have informed real-world planning and implementation. This keynote presentation highlights selected success stories that demonstrate how life cycle thinking can be translated into practical strategies for reducing environmental impacts while strengthening community resilience. These insights are especially relevant to the interdisciplinary focus of ICCSA 2026, which will be held in collaboration with the University of Minho in Braga, Portugal, from June 30 to July 3, 2026.

References

1. Weerasinghe, O., Hewage, K., Razi, F., Fatmi, M., Sadiq, R.: Applicability of land use models for sustainable urban planning: a review. J. Urban Plann. Dev. ASCE (2025). Accepted 3 Marc 2025

2. Sultana, S.R., et al.: Looking beyond energy efficiency targets: life cycle sustainability of mechanical and water heating equipment in Canadian homes. J. Clean. Prod. **473**, 143469 (2024). https://doi.org/10.1016/j.jclepro.2024.143469
3. Liyanage D.R., Hewage K., Hussain S.H., Razi F., Sadiq R.: Climate adaptation of existing buildings: a critical review on planning energy retrofit strategies for future climate. Renew. Sustain. Energy Rev. **199**, 114476 (2024). https://doi.org/10.1016/j.rser.2024.114476
4. Liyanage D.R., Hewage K., Karunathilake H., Sadiq R.: Feasibility of carbon-capturing in building heating systems: a life cycle thinking-based approach. Int. J. Greenhouse Gas Control **132** (2023). https://doi.org/10.1016/j.ijggc.2023.104056
5. Moudgil, V., Sadiq, R., Bakhtavar, E., Paudel, A., Hewage, K.: A cloud-oriented data-analysis framework to analyze peak demand dynamics in institutional building clusters. Sustain. Cities Soc. **111**, 105553 (2024). https://doi.org/10.1016/j.scs.2024.105553
6. Liyanage D.R., Hewage K., Ghobadi M., Sadiq R.: Thermal resiliency of single-family housing stock under extreme hot and cold conditions. Energy Build. **323**, 114809 (2024). https://doi.org/10.1016/j.enbuild.2024.114809
7. Wijayasekera S.C., et al.: Sustainability of waste-to-hydrogen conversion pathways: a life cycle thinking-based assessment. Energy Convers. Manage. **270** (2022). https://doi.org/10.1016/j.enconman.2022.116218
8. Jayasena A., Hewage K., Siddiqui O., Sadiq R.: Socio-economic and environmental cost-benefit analysis of passive houses: a life cycle perspective. J. Clean. Prod. **373** (2022). https://doi.org/10.1016/j.jclepro.2022.133718

Short Bio – Prof. Kasun Hewage is a Full Professor and FortisBC Smart Energy Chair in the School of Engineering at the University of British Columbia. He also serves as Associate Director of UBC's Clean Energy Research Centre and Director of the Life Cycle Management Laboratory. Prof. Hewage brings extensive multidisciplinary industrial experience in Canada and internationally. Prior to joining UBC in 2008, he worked as a business management cost specialist in the oil and gas sector and as a civil engineer on hydropower and infrastructure development projects. His research focuses on integrating life-cycle thinking into urban development, energy policy, and infrastructure planning. A distinctive feature of his work is its whole-system perspective, extending beyond conventional assessment methods to evaluate products, processes, and built assets from material extraction through end-of-life. Through this approach, Prof. Hewage and his research team have supported governments, infrastructure developers, and communities in advancing low-impact and net-zero development strategies for both existing and planned communities. Prof. Hewage currently leads many collaborative research projects on life-cycle management of built assets, green construction, smart energy systems, and sustainable community planning. His contributions have helped establish important benchmarks, decision-support methods, and implementation pathways for advancing sustainability in the Canadian and global construction sectors. He has authored or co-authored more than 300 peer-reviewed journal publications, in

addition to numerous book chapters and conference papers presented at national and international forums. Prof. Hewage is registered as a Professional Engineer in British Columbia, Canada. He is a Fellow of the Canadian Society for Civil Engineering, the Canadian Society of Senior Engineers, and the Engineering Institute of Canada.

Machine Learning Based Methodologies for Grape and Wine Production

Pedro Melo-Pinto

University of Trás-os-Montes and Alto Douro, Engineering Department, Centre for the Research and Technology of Agro-Environmental and Biological Sciences, Portugal

Abstract. Wine production is a major area in Portuguese economy. The sector accounts for approximately 2.6% of Portugal's GDP, and is crucial to the development of inland regions. Wine quality depends on grape quality, which is a result of different factors such as harvesting at the optimal point of maturity, or selecting the grapes according to quality features. The growing importance of precision agriculture, demands novel methods to make the most out of the data generated. Machine learning approaches are particularly suitable for complex processes' modelling, where too many arguments are involved. Many of the agricultural processes such as yield estimation or ripening, fall into this category. However, several challenges may arise: even if models are available for particular cases, general models are difficult to come by, largely due to the high variability involved; with such models, there is a need for huge amounts of data, in a way which will not always be easy to obtain. This is particularly true in the case of grape production and the wine industry, where the number of grape varieties, climate, soil, etc., introduce a degree of variability that is uncommon in other sectors. One way of obtaining more information in such cases has been the use of images, whether RGB, multi-spectral or hyperspectral. The use of images introduces additional challenges, such as data size and lighting conditions, which underscores the need for data-driven modelling methodologies. The use of machine learning (ML) techniques combined with images has been establishing itself as a viable alternative predicting yields or quality parameters in grapes, and assist on harvesting critical decisions. Several ML approaches have been proposed to handle such data characteristics, but selecting a suitable methodology that best address the problem under study and make sure it generalizes well, is, most of the times, a cumbersome task. We will address three wine and grape production problems, yield estimation, grape quality assessment and illness detection, presenting different difficulties. Yield estimation is a major aspect in the wine production industry. The number of berries is one of the most relevant yield aspects, and can be estimated using computer vision, namely image segmentation algorithms. Lighting is one of the key factors in computer vision approaches, and occlusions—which are common in this problem—can be minimized, but require estimation models to achieve reliable results. Illness early detection is, of course, of the utmost importance, with an impact on production and quality. Detecting early symptoms and distinguishing them from other conditions is a difficult task, which makes the use of machine learning algorithms advisable, using automatic visual inspection of the leaves, addressed as a pixel-based classification problem. However, the available datasets are limited and complex to compile, as different diseases typically require very

specific timeframes and locations. The use of generative models to overcome this obstacle, in conjunction with robust classification models, lead to models that are intended to be as general as possible. The evaluation of the grape's maturation profile based on the evolution of their oenological parameters over time is one of the most important aspects for ensuring wine quality, playing a decisive role in the definition of the optimal time for harvesting. The use hyperspectral imaging combined with learning algorithms represents a faster and non-destructive grape ripeness assessment method. We focus in two fundamental aspects to address the resulting data variability: a possible wavelength bands selection (with the purpose of reducing the dimensionality of data without losing predictive power) and the generalization ability of the ML model under such demanding conditions.

Short Bio – Pedro Melo-Pinto graduated from Univ. do Porto in 1984 and received a Ph.D. in Computer Science from the Univ. de Trás-os-Montes e Alto Douro (UTAD) in 1998. He is a Full Professor at UTAD where he served as Pro-Rector, as Director and Vice-Director of R&D units, and as member of UTAD's General Council. He also served as a member of The Portuguese Strategic Infrastructures (for the Digital Areas) Monitoring Committee and usually serves as a member (or co-chair) of the evaluation panel for different Portuguese and international R&D projects and grants competitions. He is currently Associate Editor of Computers and Electronics in Agriculture journal. He was Chair of the EuroFuse 2011 conference and the NATO 2001 Advanced Research Workshop, and served as member of the organization committee for more than 40 international conferences/workshops/special sessions. He is the head of the Image-Based Systems Lab from the Centre for the Research and Technology of Agro-Environmental and Biological Sciences.

His scientific interests include Computer Vision (CV) and Machine Learning (ML) and, in the last 15 years, their applications to the Agro-forestry area, namely to grape oenological parameters assessment. He authored/co-authored more than 100 scientific papers. He participated in many Portuguese and international R&D projects, predominantly in the areas mentioned above.

Contents

Computational Methods, Algorithms and Scientific Applications

Securing Corporate Data Exchange: An Efficient Identity-Based Authenticated Signcryption Over Elliptic Curves 3
Minh Tu Le, Van Nghi Nguyen, Hanh Thi Tran, Minh Thang Vu, and Van Duan Nguyen

Algorithmic Complexity Underpins When Path Information Improves Graph Neural Networks Performance on Molecular Graphs 18
Abdulrahman Ibraheem, Narsis Kiani, and Jesper Tegner

A Systematic Performance Study of Parallel Programming Models for Stencil-Based CFD Applications on Heterogeneous Architectures 35
Mariana Padilha, Claudio Schepke, Tailí Petry, and José Rufino

Uncertainty Quantification of Underwater Sound Propagation Using Surrogate Models 51
Mikhail S. Lytaev

Low-Cost Wearable System for Essential Tremor Monitoring with Passive Damping 67
Gustavo Adolpho Bonesso, Carlos Marcelo Gurjão de Godoy, Eliana Chaves Ferretti, and Regina Célia Coelho

Analyzing Internal Sensitivity in Neural Network Representations via Jacobian Norms and Perturbation Analysis 79
Aninda Kumar Sharma

Numerical Algorithm for the Biermann Effect in ICF Self-generated Magnetic Fields and Its Applications 96
Yang Rong, Song Peng, and Wang Qiang

Frequency Domain Prediction of NACA 0012 Self-noise Using Data-Driven Machine Learning 110
Rishav Mishra, Alvis Fong, and Bade Shrestha

Beyond Endpoint Quality: A Phase-Aware Analysis of Search Dynamics in Triangle Reassembly 130
Damir Hasanspahić, Adis Alihodžić, and Damir Hasić

Mesh Optimization for the One-Way Helmholtz Equation in an Inhomogeneous Medium 144
Mikhail S. Lytaev

New Metaheuristic Algorithms to Reduce the Bandwidth of Real-World Graphs 161
Alexandre Augusto Alberto Moreira de Abreu and Sanderson L. Gonzaga de Oliveira

Shelf Space Allocation with Rectangular Arrangement: A Comparative Study of Three MILP Formulations 179
Daniel Martins, Maria João Alves, and Cristela Bairrada

Architecture Foundations and Deployment Scalability for CWMP: A Technical Review 192
Marco Marinello

Less is More: XAI-Driven Feature Selection for Efficient Molecular Property Prediction 204
Jonas Lucas Durão, Luís G. Dias, and Marcos G. Quiles

High Performance Computing and Networks

AMMOF: An Adaptive Multi-metric Objective Function for Enhanced RPL Routing in Industrial IoT Networks 221
Lyes Ibersiene, Badis Djamaa, Mustapha Reda Senouci, Ahmed Yadine Merabet, and Oussama Zahali

Design and Experimental Evaluation of Secure Overlay Networks for Docker Microservices Architectures 236
Damiano Perri, Osvaldo Gervasi, and Francesco Bietolini

Dynamic Regression Algorithm and Multi-Objective Query Optimization in Spark SQL 251
Trung-Dung Le

Analyzing Optimization Trade-Off Between Energy and Model Convergence in Federated Learning 269
Nilo Marchiori Louzada, Renato Elias Nunes de Moraes, Rodolfo S. Villaça, Vinícius F. S. Mota, and Maria Claudia Silva Boeres

2D CNN-Based Link Prediction in Opportunistic Networks 287
Vinícius da Silva Thiago, Ronaldo Moreira Salles, Julio Cesar Duarte, and Gabriela Moutinho de Souza Dias

Evaluating Accuracy–Performance Trade-Offs in Deep Learning Frameworks for Diabetic Retinopathy Detection 303
Bruno Morales, Cristiano Künas, Rodrigo Machado, Pedro Rigon, and Philippe O. A. Navaux

Topology-Aware Distributed Ensemble Learning Using the vCube Hierarchical Hypercube Overlay 321
Charles Giovane Salles, André Luiz Brun, and Luiz Antonio Rodrigues

Automating the Technical Validation of FHIR Implementation Guides: A Data-Oriented Framework Within the HfPT Project 340
Ivo A. Rodrigues, Leopoldo O. Silva, Cátia Pereira, Rute Almeida, Rui C. Ribeiro, and Francisco J. Duarte

Characterizing Lossless GPU Data Compression Across AMD CDNA and RDNA Architectures 357
Cristiano Künas, Gabriel Freytag, Jean Luca Bez, Thiago Araújo, and Philippe Navaux

A Systematic Literature Review of Network-on-Chip Architectures: Topologies, Routing, Reliability, and Emerging Technologies 375
Asrani Lit, Rohana Sapawi, Shamsiah Suhaili, Sharifah Masniah Wan Masra, and Kuryati Kipli

Geometric Modeling, Graphics and Visualization

Adaptive Hyperspectral Image Classification Using a CNN-MAOELM Hybrid Architecture 393
Oleksii Fesenko, Robert Bieliakov, Hryhorii Radzivilov, Andrii Bernatskyi, and Dmytro Koltovskov

A Re-parameterized Multi-scale and Global Context Framework for Waste Detection in Vietnamese Street Scenes 413
Phat Hoang Ho, An Hoang Nguyen, Kien Trang, and Bao Quoc Vuong

LCRO-GAN: A Multi-stage GAN Framework for Image-Based Reconstruction of Historical Monuments 428
Pranav Jigalur, Likhith Gowda, Kripa S Rai, Praveen Koppal, and Sindhu R Pai

Run-Time Assurance for Onboard COTS-Based AI in Safety-Critical LEO Small-Satellite Systems 445
Omer Khalid, Jose-Carlos Gamazo-Real, and Elena Roibás-Millán

Surgical Instrument Tracking: a Lightweight YOLO Approach Across Diverse Domains 461
Nathalia Frederico, Rafael Miranda, Saulo Silva, and Jefferson Morais

Deep Learning-Based Crack Detection on Complex Decorative Surfaces of Cultural Heritage Artifacts 478
HongYi Shih, Debora Verniz, Filiz Bunyak, and Jessica Boldt

StableZero: Improving General Image Generation Through Visualisation Feedback 492
Kazim Laos, Hector D. Menendez, and Gema Bello-Orgaz

From Random to Pre-trained ViT Backbones for Improved Feature Extraction: Application to Diagnosis with Sensor Images 510
Luan B. Guerra, Ana Beatriz S. Zerati, Wallace Casaca, and Lucas C. Ribas

Classification of Dynamic Libras Signs Related to the Hospital Context Using Recurrent Neural Networks 529
Pedro de Araújo Bhering Bittencourt, Luiz Maurílio da Silva Maciel, Marcelo Bernardes Vieira, and Saulo Moraes Villela

A Comparative Study of Prompt Engineering Strategies for Data Visualization Scenarios 544
Camile Maria Cunha Aguiar, Walbert Cunha Monteiro, Thiago Augusto Soares de Sousa, Roberto Yuri da Silva Franco, and Bianchi Serique Meiguins

Detection of Cow Face and Subregions Using a New Annotated Dataset 561
Mathews Edwirds Gomes Almeida, Pedro de Araújo Bhering Bittencourt, Brian Luís Coimbra Maia, Lucas Silva Santana, João Vítor de Castro Martins Ferreira Nogueira, Daniel Muller Rezende, Gabriel Rezende da Silva, Thais de Souza Marins, Luiz Maurílio da Silva Maciel, Marcelo Bernardes Vieira, Saulo Moraes Villela, and Bruno Campos de Carvalho

Enhanced Architecture for Non-destructive Leaf Area Estimation Based on a Semantic Segmentation Network 580
Caio Seixas Duarte, Luiz Maurilio da Silva Maciel, Saulo Moraes Villela, and Marcelo Bernardes Vieira

Advanced and Emerging Applications

Beam Forming and Super-Grouping for Seismic Data Enhancement in 5D Domain 599
Maxim Protasov, Maxim Dmitriev, Young Seo Kim, and Denis Sabitov

Quantile Regression for Soil Organic Carbon: A Distributional Analysis of Soil Variability 613
Edson L. Almeida, Ian Alexander Zahner McIntosh, Joel André Ferreira dos Santos, Ana Carolina Ferreira, Angela Ferreira, Jorge Soares, Marcos Bacis Ceddia, and Diego Brandão

Comparative Evaluation of Filter and Wrapper Feature Selection Methods in Species Distribution Modeling 628
Yousra Cherif and Ali Idri

Provision of Financial Security of the State and Ways of Threat Prevention 644
Nadiia Davydenko, George Abuselidze, Olena Gudz, and Svitlana Onyshko

Bayesian Models to Predict Above Ground-Biomass 660
Ricardo Coelho, Isabel Natário, and Silvia Fraile

Author Index 679

Computational Methods, Algorithms and Scienti c Applications

Securing Corporate Data Exchange: An Efficient Identity-Based Authenticated Signcryption Over Elliptic Curves

Minh Tu Le, Van Nghi Nguyen(✉), Hanh Thi Tran, Minh Thang Vu, and Van Duan Nguyen

Academy of Cryptography Techniques, Hanoi, Vietnam
{nghinv,hanhtt,thangvm,nguyenvanduan}@actvn.edu.vn

Abstract. In this paper, we propose a lightweight signcryption scheme integrating an identity based signature and the Ascon cipher for applications in authentication and data security of enterprise network environments. Our objective is to provide a secure signcryption scheme that combines public key distribution and user authentication with identity information at a reduced implementation cost compared to using PKI system. The proposed scheme is designed for application to the S/MIME protocol to ensure secure email communication in enterprise network environments. We conduct a thorough analysis of the security measures and compare the computational overhead of the proposed scheme with other published works that use PKI system. This will enable us to determine the level of security and efficiency offered by the proposed scheme.

Keywords: Ascon cipher · Elliptic curve · Identity based signature · Signcryption

1 Introduction

Signcryption, a cryptographic technique established by Zheng in 1997 [1], combines the processes of signature and encryption inside a single scheme. In the last 10 years, there have been many publications on signcryption schemes [2–9]. These schemes all include two cryptographic primitives: public key cryptography and a secret key cryptosystem. The public key cryptography used can be on finite fields such as RSA, Elgamal cryptosystem, or elliptic curve cryptosystem. Elliptic curve cryptographic schemes have small key lengths while having the same security as finite field cryptographic schemes based on RSA and Elgamal. Currently, computer networks are used in several aspects of human activity, including employment, education, commerce, and entertainment. Authentication of entities and communication data in computer networks regarding identity and origin is a critical information security concern. In contemporary computer networks, this task is accomplished by cryptographic methods, including digital signatures and digital certificates provided by a Public Key Infrastructure (PKI) from a trusted third party inside the network.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 3–17, 2026.
https://doi.org/10.1007/978-3-032-30491-9_1

In this paper, the problem we are interested in is the way of exchanging public keys between communicating parties. In [2–4, 6], finite-field signcryption schemes are integrated with a method of distributing public keys directly through a public channel and verification based on identification. Paper [5] propose an elliptic curve signcryption scheme based on the theory of linear pairs, and the distribution and authentication of public keys are also based on user identification. The signcryption schemes of [7, 8] are based on using the PKI system to exchange public keys, and the publication of [9] establishes a trusted management server to be able to distribute and authenticate public keys for the elliptic curve signcryption scheme.

For elliptic curve signcryption schemes, the method of exchanging public keys between communicating parties through identification has only been published in the direction of using cryptography on linear pairs; however, there is no published standard on ensuring security for this type of cryptography. The utilization of the PKI system necessitates both parties involved in communication to acquire digital certificates from this infrastructure, resulting in substantial expenses for deployment and installation. This approach is particularly unsuitable for network environments with a large number of users, as it is impractical to compel ordinary network users or IoT devices to obtain and purchase digital certificates. In this paper, we propose a lightweight signcryption scheme for data security in the enterprise network environments. Our proposed scheme has a different approach that employs an identity-based key generation mechanism for generating user key pair, allowing user to directly exchange public keys with each other. Furthermore, it utilizes identity information and public key for user authentication. The deployment of identity-based key pairs for users reduces dependence on PKI, thereby simplifying deployment, reducing operational costs, and improving system scalability. This approach still helps the proposed signcryption scheme, in addition to ensuring the general security goals of a signcryption scheme such as confidentiality, integrity, origin authentication, data tampering, and non-repudiation. It also ensures user authentication, forward security, internal security, and especially low computational overhead, making it easy to install and deploy. The proposed signcryption scheme is applied to email security, specifically by applying it to the S/MIME protocol to secure end-to-end email communication between two parties. The security properties of the proposed scheme will be analyzed in detail in this paper. We also compare the computational cost and implementation performance with published elliptic curve signcryption schemes using trust centers to distribute public keys, such as [9].

This paper consists of 6 main sections. After the introduction in Sect. 1, we present the theoretical foundations of the signcryption scheme, the identity-based signature scheme, and other cryptographic primitives in Sect. 2. These are the foundations for proposed signcryption scheme for application in the data security of enterprise network environments in Sect. 3. The security properties of the proposed scheme are analyzed in Sect. 4. In Sect. 5, we discuss the computational overhead and performance of the proposed scheme with some other publications. Finally, Sect. 6 concludes the paper.

2 Preliminaries

This section provides the theoretical foundation for the scheme presented in this work. It will cover the definition and characteristics of the signcryption scheme, the identity-based signature scheme and elliptic curve cryptography, as well as the Ascon cipher.

2.1 Signcryption Scheme

A typical signcryption scheme consists of four algorithms: Setup (SU), Key generation (KG), Signcryption (SC) and Unsingcryption (USC). In which the SU algorithm declares the domain parameters of the cryptographic primitives used in the signcryption scheme, KG is the algorithm for generating key pairs for participating users, SC is the algorithm for encrypting data and signing probabilistically, and USC is the algorithm for verifying digital signatures and decrypting data. Any signcryption scheme must provide two properties: efficiency, and security.

- Efficiency: The computational and communication overhead of a signcryption scheme should be less than the best alternative of a signature then encryption scheme using similar functionalities.
- Security: Any signcryption scheme must achieve four main security properties: confidentiality, unforgeability, integrity, and non-repudiation. In addition, some signcryption schemes also provide additional security properties such as public verifiability and forward secrecy of message confidentiality.

2.2 Identity-Based Signature

The identity-based signature (IBS) was first introduced by Adi Shamir in 1984 [11], with the objective of reducing reliance on PKI and digital certificates, thereby simplifying deployment and reducing operational costs. In [11], the user's public key can be derived from their identity information, such as the email address, phone number, personal identifier, or IP address. The corresponding secret key is issued by a trusted Key Generation Center (KGC). An IBS scheme typically consists of the following main processes:

a) Key Generation Process
This process takes as input a secret parameter λ (a positive integer). The output is a master key pair (m_{pk}, m_{sk}), where m_{pk} is master public key and m_{sk} is master secret key. The key generation process is performed by the KGC.

b) Key Extraction Process
This process takes as input the user's identity id and the master key pair (m_{pk}, m_{sk}). . The output is a key pair (pk_{id}, sk_{id}), where pk_{id} is user's public key and sk_{id} is user's secret key. The key extraction process is performed by the KGC.

c) Signing Process
This process takes as input the message m and the signer's secret key sk_{id}. The output is a digital signature s. The signing process is performed by the signer.

d) Verification Process
This process takes as input the message m, the digital signature s and the signer's public key pk_{id}. The output is a message about whether the signature is valid or invalid. The verification process is performed by the verifier.

2.3 Elliptic Curves

a) Elliptic Curves in Weierstrass Form
An elliptic curve over the finite field F_p is defined as the set of points $(x, y) \in F_p$ that satisfy the following equation:

$$y^2 \equiv x^3 + ax + bmodp \tag{1}$$

where $a, b \epsilon F_p$ and $4a^3 + 27b^2 modp \neq 0$. The elliptic curve is denoted by $E_p(a, b)$. The point O is the infinity point on the elliptic curve $E_p(a, b)$. Equation (1) defines the elliptic curve in Weierstrass form.

b) Point Addition in Weierstrass Form
Given $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ are two points on $E_p(a, b)$. The rules for point addition on $E_p(a, b)$ are defined as follows:

1) $P_1 + O = O + P_1 = P_1$.
2) If $x_1 = x_2$ and $y_1 = -y_2 modp$ (i.e. $P_1 = -P_2$), then $P_1 + P_2 = O$.
3) If $P_1 \neq -P_2$ then $P_1 + P_2 = P_3 = (x_3, y_3)$:

$$x_3 = \left(\lambda^2 - x_1 - x_2\right) modp.$$

$$y_3 = (\lambda(x_1 - x_3) - y_1) modp.$$

where:

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} incaseofP \neq Q \\ \frac{3{x_1}^2 + a}{2y_1} incaseofP = Q \end{cases}$$

c) Scalar Multiplication in Weierstrass Form

$$Q_1 = t \cdot P_1 = P_1 + P_1 + P_1 + \cdots + P_1(ttimes), \text{ where } t < p.$$

2.4 Ascon-128 Cipher

Ascon cipher was chosen by US National Institute of Standards and Technology (NIST) for future standardization of the lightweight cryptography in 2023.

The ciphers are parameterizable by the key length k (up to 128 bits), "rate" (block size) r, and two numbers of rounds n_1, n_2. All algorithms support authenticated encryption with plaintext P and additional authenticated data A (that remains unencrypted). The encryption input also includes a public nonce N, the output - authentication tag T, size of the ciphertext C is the same as that of P. The decryption uses N, A, C, and T as inputs and produces either P or signals verification failure if the message has been altered. Nonce and tag have the same size as the key K (k bits).

3 Proposed an Efficient and Secure Identity-Based Authenticated Signcryption Scheme Over Elliptic Curve Applied in Enterprise Network Environment

In this section, we first analyze the enterprise network model and then propose a signcryption technique designed for data security in this specific network model, specifically applying it to the S/MIME protocol. Subsequently, we evaluate the correctness of the proposed scheme.

3.1 Application of the S/MIME Protocol for Data Security in Enterprise Network Model

Consider the common enterprise network architecture shown in Fig. 1, which consists of an internal network (LAN), servers (Email and Authentication), a Certificate Authority (CA), switch devices (S1, S2), a router with an integrated firewall, and the Internet. The Authentication Server (AS) will possess a digital certificate that includes both a public key and a private key. This certificate will be issued by a trusted third-party CA and transmitted through a secure channel. For the users to access the Email Server and utilize email services, it is necessary to perform authentication with the AS.

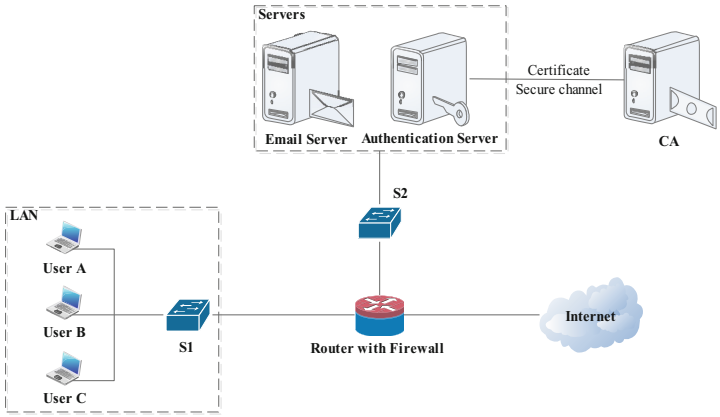


Fig. 1. Enterprise network model

In modern enterprise environments, email serves as a primary communication channel for information exchange among users and departments. Through email systems, various critical data are transmitted, including sensitive information such as financial records, employee and customer personal data, business documents, and contracts,… The exchange of such sensitive information via email necessitates the implementation of appropriate security mechanisms to protect data during transmission. In this context, cryptographic techniques, including encryption and digital signatures are employed to ensure the fundamental security properties of the information: confidentiality, integrity, authentication, and non-repudiation. In a typical enterprise network model, as shown in

Fig. 1, achieving end-to-end email security between users A and C requires the application of the S/MIME protocol. S/MIME provides security mechanism for email messages by combining digital signatures and encryption. This security mechanism is illustrated in Figs. 2 and 3.

In a typical enterprise network model, as shown in Fig. 1, achieving end-to-end email security between users A and C requires the application of the S/MIME protocol. S/MIME provides security mechanism for email by combining digital signatures and encryption. This security mechanism is illustrated in Figs. 2 and 3.

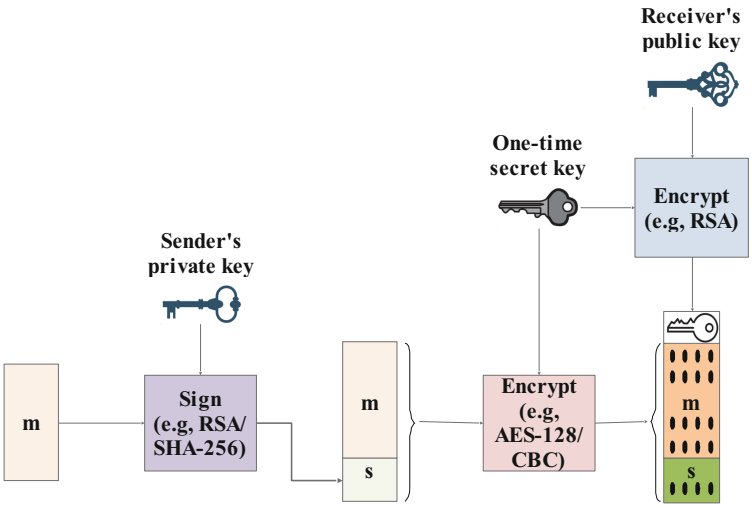


Fig. 2. S/MIME signcryption process

Figure 2 illustrates the signcryption process in the S/MIME protocol. This process consists of the following steps:

1) The Sender creates a message *m*.
2) The Sender uses RSA/ SHA-256 with their private key to generate a digital signature *s* for *m* and then *s* is attached to *m*.
3) The Sender generates a one-time secret key.
4) The message *m* and signature *s* are encrypted as a single block with AES-128/CBC using the one-time secret key.
5) The one-time secret key is encrypted with RSA using the Receiver's public key and the result is appended to the ciphertext of *m* and *s*.

Figure 3 illustrates the unsigncryption process in the S/MIME protocol. This process consists of the following steps:

1) The Receiver uses RSA with its private key to decrypt and recover the one-time secret key.
2) The one-time secret key is used to decrypt and recover the *m* and *s*.
3) The Receiver uses RSA/ SHA-256 with the Sender's public key to verify digital signature *s*. If the signature *s* is valid, the message *m* is accepted.

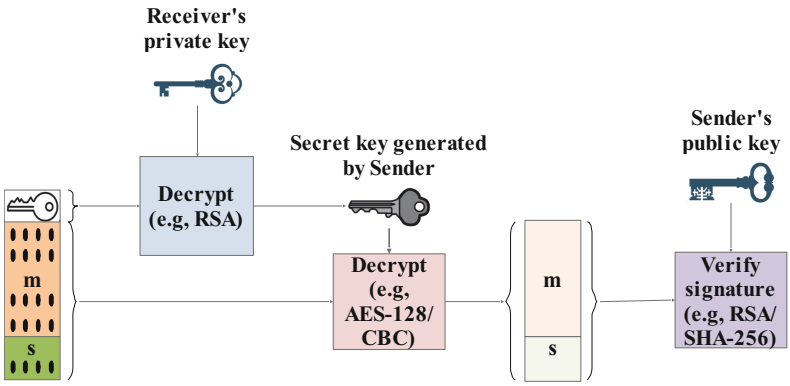


Fig. 3. S/MIME unsigncryption process

In the email security model using S/MIME, each user must possess a public/ private key pair to perform digital signature and email encryption functions. In practice, to ensure the authenticity and reliability of private keys, users typically obtain digital certificates from a trusted third-party CA or deploy an internal PKI to manage and distribute certificates. However, this approach leads to several limitations, including complex deployment and management processes, high operational costs, and limited scalability as the number of users increases.

Therefore, this paper proposes an identity-based authenticated signcryption scheme that reduces reliance on digital certificates and PKI. The scheme is designed for application to the S/MIME protocol in enterprise network environments, offering low cost and ease of deployment, while preserving the essential security properties of email: authentication, confidentiality, integrity, and non-repudiation. The proposed scheme focuses on two main components: a user key generation mechanism based on identity information and a signcryption scheme integrating identity-based authentication.

3.2 Proposed Identity-Based Key Generation Mechanism in Enterprise Network

Firstly, we propose a method for generating a user's key pair based on identity information in the enterprise network system.

Figure 4 illustrates the user key generation process based on identity information. Specifically, the user first sends their identity (ID) to the AS. The server then generates and issues an identity-based key pair to the user over a secure communication channel. In this process, the key pair (mpk, msk) is used as input parameters, where mpk denotes the master public key and msk denotes the master secret key. This master key pair (mpk, msk) is issued by a trusted third-party CA and distributed to the AS via a digital certificate. The certificate is transmitted over a secure channel and this master key pair is owned by the AS. The identity-based key generation mechanism is presented in Fig. 5.

The key generation mechanism utilizes an elliptic curve $E_p(a, b)$ with base point G and degree n as domain parameters. Additionally, a secure hash function H is employed.

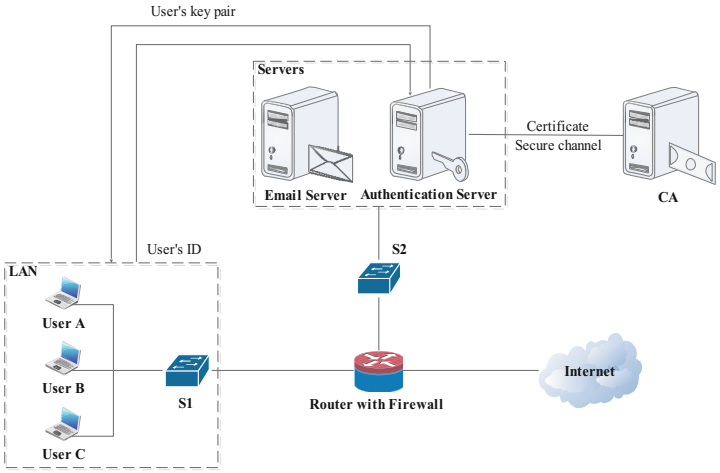


Fig. 4. Key generation process based on user's identity

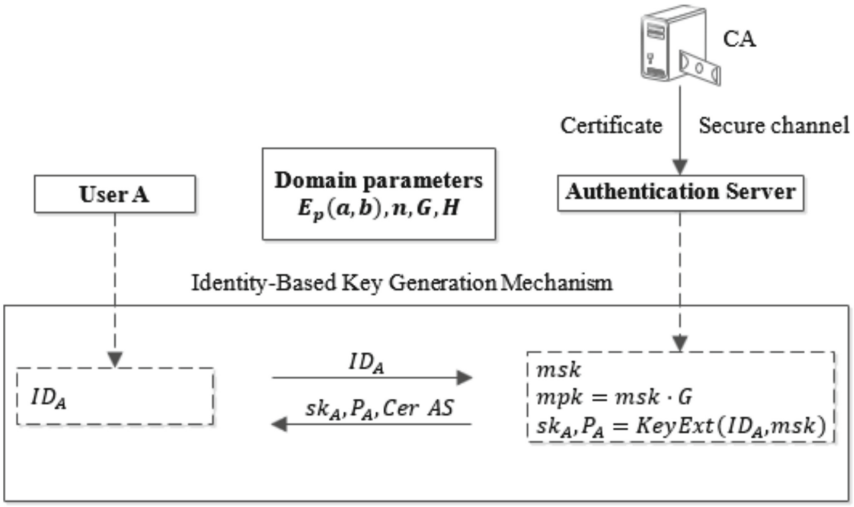


Fig. 5. Identity-based key generation mechanism

First, user A sends its identity ID_A to the AS. Then, the AS generates a key pair for A using the Key Extraction Algorithm.

Key Extraction Algorithm

This algorithm takes as input parameters the identity ID_A and the mater key pair (mpk, msk). The output parameter is the A's key pair (P_A, sk_A), where P_A denotes the public key and sk_A denotes the secret key. This algorithm works as follows:

1) Randomly generate an integer *nonce* such that $0 < nonce < n$
2) Compute $l_A = H(msk \| ID_A \| nonce) mod n$

3) Compute $F_A = l_A \cdot G$ and set $p_A \equiv x_{F_A} modn$, where x_{F_A} is the x coordinate of point F_A. If $p_A = 0$, go back to step 1
4) Compute $t_A = H(mpk \| ID_A) modn$. If $t_A = 0$ then set $t_A = 1$
5) Compute $sk_A \equiv (p_A \cdot msk + l_A \cdot t_A) modn$. If $sk_A = 0$, go back to step 1
6) Compute $P_A = sk_A \cdot G$

The AS transmits the identity-based key pair (P_A, sk_A), the point F_A and the digital certificate *CerAS* to user A via a secure communication channel. The process of issuing identity-based key pairs for other users in the enterprise network model is similar to that for user A described above.

Public Key Verification Process
In the proposed approach, it is not necessary to deploy a PKI infrastructure to authenticate the public key P_A of user A. Instead, the validity of P_A can be verified through the following steps:

1) A sends the parameters ID_A, P_A, F_A to the verifier
2) The AS transmits the master public key mpk to the verifier. The verifier can then authenticate the validity of mpk through a PKI
3) Verifier computes $p_A \equiv x_{F_A} modn$, where x_{F_A} is the x coordinate of point F_A
4) Verifier computes $t_A = H(mpk \| ID_A) modn$. If $t_A = 0$ then set $t_A = 1$
5) Verifier checks the equation $P_A = p_A \cdot mpk + t_A \cdot F_A$. If it is satisfied, then P_A is valid. Otherwise, P_A is invalid.

Under this approach, the public key of user A is verified by the verifier using A's identity information and the master public key of the AS.

3.3 Operation of the Proposed Signcryption Scheme

Assume that two users A and C in the enterprise network model shown in Fig. 1 who intend to perform signcryption and exchange electronic documents. Initially, each user must possess an identity-based key pair. These key pairs are generated and issued by the Authentication Server; with A receiving (P_A, sk_A) and C receiving (P_C, sk_C). The identity-based key generation process is detailed in Sect. 3.2. In this scenario, the IB-SCECC signcryption scheme is employed.

The proposed signcryption scheme (IB-SCECC) utilizes an elliptic curve $E_p(a, b)$ with base point G and degree n. Furthermore, the Ascon cipher and the secure hash function H are employed. The IB-SCECC scheme consists of two main processes: the signcryption process and the unsigncryption process.

a) Signcryption Process
This process is performed by A. Additionally, A also receives the parameters ID_C, P_C and F_C from C, which enable A to verify the validity of C's public key P_C. The public key verification process is detailed in Sect. 3.2.

Using the key pair (P_A, sk_A), A executes the signcryption process. Specifically, A first generates a signature (r, s) for the message m and attaches (r, s) to m. Then, A computes a session key K and employs it to encrypt the plaintext pt consisting of m

and (r, s) using the Ascon cipher. Finally, A obtains the ciphertext ct and transmits it to C along with ID_A, ts and the point I. The signcryption process is described in detail in Fig. 6.

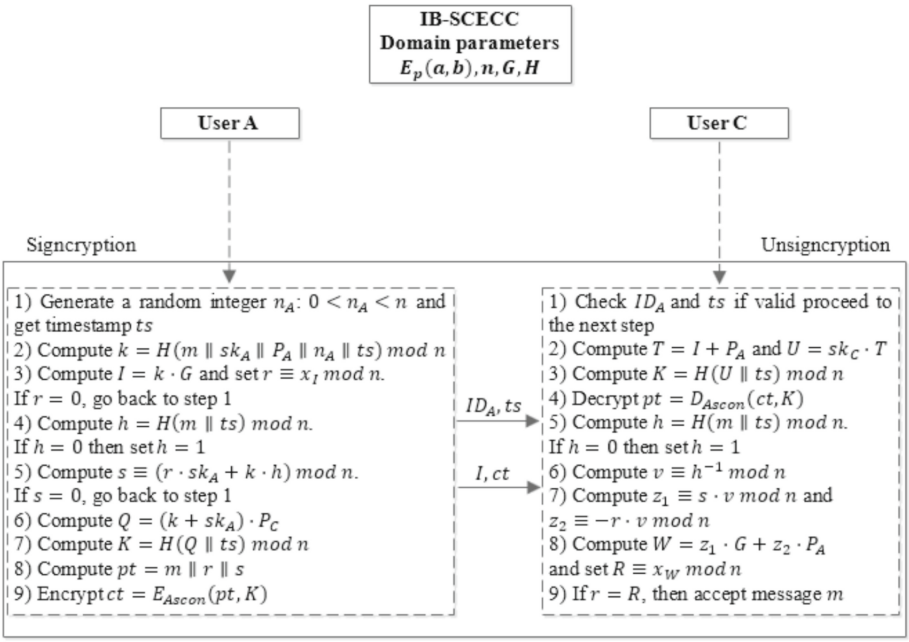


Fig. 6. IB-SCECC signcryption scheme

b) Unsigncryption Process

This process is performed by C. Additionally, C also receives the parameters ID_A, P_A and F_A from A, which enable C to verify the validity of A's public key P_A. The public key verification process is detailed in Sect. 3.2.

User C first verifies the validity of ID_A and ts. Next, C computes a session key K and employs it to decrypt the ciphertext ct using the Ascon cipher to recover the plaintext pt, which includes m and (r, s). Finally, C verifies the validity of the signature (r, s) associated with m. The message is accepted if the signature is valid, otherwise, it is rejected. The unsigncryption process is described in detail in Fig. 6.

3.4 Correctness of the Proposed Signcryption Scheme

For the proposed signcryption scheme, correctness is guaranteed when the following two conditions are satisfied:

- The secret session key computed on both communicating parties is the same.
- The digital signature for message m is valid.

The Correctness of the Session Key Calculated on Both Communicating Sides
To ensure the correctness of the scheme, the session keys computed by both communicating parties must be proven to be the same, i.e., $K = H(Q\|ts)modn = H(U\|ts)modn$. Consider $K = H(Q\|ts)modn$ is calculated at user A side, $K = H(U\|ts)modn$ is calculated at user C side. We only need to show that $Q = U$.

Specifically, $Q = (k + sk_A) \cdot P_C = k \cdot P_C + sk_A \cdot P_C = k \cdot P_C + sk_A \cdot sk_C \cdot G$. On the other hand, $U = sk_C \cdot T = sk_C \cdot I + sk_C \cdot P_A = k \cdot P_C + sk_A \cdot sk_C \cdot G$.

Therefore, we have $Q = U$, that is the session key calculated at both communication sides is the same.

The Correctness of the Signature Verification
User C accepts the message m only if the digital signature is valid i.e., if it can verify that the values r and R are equal. Since $r \equiv x_I modn$ and $R \equiv x_W modn$, we only need to show that the point I equals point W.

Specifically, $I = k \cdot G$. On the other hand, $W = z_1 \cdot G + z_2 \cdot P_A = G \cdot (s \cdot v - r \cdot v \cdot sk_A) = G \cdot ((r \cdot sk_A + k \cdot h) \cdot v - r \cdot sk_A \cdot v) = G \cdot (r \cdot sk_A \cdot v + k \cdot h \cdot v - r \cdot sk_A \cdot v) = G \cdot (k \cdot h \cdot h^{-1}) = k \cdot G$.

So, we have $I = W$ and $r = R$ ensures that C can verify the correctness of user A's signature.

Therefore, our signcryption scheme is correct.

4 Security Analysis

The proposed signcryption scheme employs an Identity-based key generation mechanism to generate a key pair for users and the Ascon cipher for encryption, so the security properties provided by the scheme include user authentication, confidentiality, unforgeability, integrity and non-repudiation, forward security, and internal security. Now, we will examine each of these properties thoroughly.

User Authentication: User C can always authenticate user A by using the Public Key Verification Process with the identity information and public key received from A as inputs. If A's public key is invalid, C terminates the unsigncryption process and discards the ciphertext. On the other hand, A can also authenticate C.

Confidentiality: The proposed scheme provides confidentiality for two objects: the plaintext and the session key. The confidentiality of the plaintext is ensured by the Ascon symmetric encryption algorithm. Currently, there is no theoretically effective attack that can completely break the Ascon cipher [10]. The session key is calculated based on hashing and point multiplication on secp256r1. Therefore, session key secrecy is based on ECDLP, and since secp256r1 is used, solving this ECDLP problem in polynomial time is infeasible when using the domain parameter secp256r1 with modulo p and order n of ~ 256 bits.

Unforgeability: The proposed scheme provides unforgeability for two objects: the identity-based key pair and the digital signature.

The attacker attempts to forge the identity-based key pair of user A by generating the forged parameters $ID_A\prime$, $sk_A\prime$, and $P_A\prime$. However, these parameters cannot be equal to ID_A, sk_A and P_A, since sk_A and P_A are generated by the Authentication Server through

the Key Extraction Algorithm using identity information of A (ID_A) and the master secret key of the Authentication Server (msk) as inputs. As the adversary does not know ID_A or sk_B, it follows that $ID_A\prime \neq ID_A$, $sk_A\prime \neq sk_A$ and $P_A\prime \neq P_A$.

Thus, the forgery of the identity-based key pair is not possible in the proposed signcryption scheme.

The attacker attempts to forge A's signature for message m using the forged parameters $k\prime$, $r\prime$, $h\prime$, $sk_A\prime$, $P_A\prime$. Then the signature $s \equiv (r\prime \cdot sk_A\prime + k\prime \cdot h\prime) mod n$ and $Q = (k\prime + sk_A\prime) \cdot P_B$ leads to $r\prime \neq R$. Furthermore, the session key $K = H(Q \| ts) mod n$ computed by attacker is different from the session key $K = H(sk_C \cdot (I + P_A) \| ts) mod n$ computed by user C. Thus, the decryption process is unsuccessful.

Therefore, signature forgery is infeasible in the proposed signcryption scheme.

Integrity and Non-repudiation: The message m is guaranteed to be intact and non-repudiable by the proposed signcryption scheme because it uses the digital signature algorithm incorporating the secure hash function.

Forward Security: The session key of the proposed signcryption scheme will be unique for each communication session because it depends on the plaintext, the key pair of user A, the public key of user C, the timestamp at the time of signing, and a randomly generated nonce value during the Signcryption process. This leads to the fact that if the session key or private key sk_A at time ts_2 is leaked, the attacker cannot find the session key at time ts_1.

Internal Security: Internal security is known in two cases: one is that an attacker knowing the user A's secret key cannot recover the message m from the ciphertext, and two is that an attacker knowing the user C's secret key cannot forge the previously received ciphertext.

In the proposed signcryption scheme, if the attacker knows the secret key sk_A, recovering m from the ciphertext ct requires knowing the session key K. However, the session key K is generated by the SHA-256 hash function of point Q, and Q is calculated from the generated value k, which depends on the pseudorandom nonce value n_A generated once for the communication session. Therefore, based on the collision resistance of SHA-256 and the one-time use of the nonce, the adversary cannot recover the original message m from the ciphertext ct.

Even if the adversary obtains the C's secret key sk_C, they cannot fabricate the valid ciphertext $ct\prime$ for message $m\prime$, since generating a valid signature for this ciphertext $ct\prime$ requires the sender's secret key sk_A, which is unavailable to the adversary.

Therefore, the proposed signcryption scheme is internally secure.

5 Comparison of Security and Performance Property of the Proposed Scheme with Other Publications

5.1 Comparison of Security Properties

In this section, we compare the security properties of the proposed signcryption scheme with some other methods published in the past 10 years. The compared signcryption schemes must use the same elliptic curve digital signature schemes. The comparison

results in terms of security properties between these schemes and our proposed scheme are described in Table 1. The security properties compared include user authentication (UA), confidentiality (CF), unforgeability (UF), integrity and non-repudiation (IT&NR), forward security (FS), and internal security (IS). All the signcryption systems listed in Table 1 include support for user authentication. However, the authentication mechanism is presented in a different manner here, for example, using the PKI system, using a Management Server (MS), and using Identity and public key (ID&PK). For other properties, there are only two options: "Y" which means provided, or "N," which means not provided. Table 1 demonstrates that our suggested signcryption scheme and Zhang's scheme successfully fulfills all six security characteristics.

Table 1. Comparison of security properties of the signcryption schemes

Scheme	UA	CF	UF	IT&NR	FS	IS
Tsai's scheme [7]	PKI	Y	Y	Y	N	N
Malik's scheme [8]	PKI	Y	Y	Y	N	N
Zhang's scheme [9]	MS	Y	Y	Y	Y	Y
Our scheme	ID&PK	Y	Y	Y	Y	Y

5.2 Performance Comparison and Evaluation

We compare and evaluate the performance of the proposed scheme with other publications according computational overhead. The computational overhead of the schemes will be measured by the number of calculations such as Point Multiplication (PM), Point Addition (PA), Modular Multiplication (MM), Modular Addition (MA), Hash Function (H), Inversion Operation (IO), Encryption (EN), and Decryption (DE) with specific measured data in Table 2.

Note that PKI, MS, and Identity are the methods of establishing and registering the signcryption scheme, in which MS is the trusted management server used to distribute session keys. Based on experimental results of calculations on secp256r1 curve and averaged over about 100 runs on the same computer Laptop Dell Precision 3520 Intel Core i7-8750H 2.20GHz processor, 16 GB of RAM, and Windows 11. We give the experimental implementation time of the calculations on secp256r1 as PM = 7.498 ms, PA = 2.760 ms, MM = 1.609 ms, MA = 1.301 ms, H = 1.938 ms, IO = 4.995 ms.

Table 2. Comparison of security properties of the signcryption schemes

Scheme	Method for registration	Signcryption	Unsigncryption
Tsai's scheme [7]	PKI	4PM + 2MM + 1MA + 2H + 1EN	3PM + 3PA + 3MM + 2MA + 1H + 1DE

(*continued*)

Table 2. (*continued*)

Scheme	Method for registration	Signcryption	Unsigncryption
Malik's scheme [8]	PKI	6PM + 1PA + 3MM + 5MA + 2H + 2IO + 1EN	3PM + 2PA + 2MM + 2H + 1IO + 1DE
Zhang's scheme [9]	MS	1PM + 1MM + 1MA + 2H + 1IO + 1EN	2PM + 1PA + 2MM + 2H + 1IO + 1DE
Our scheme	ID	2PM + 2MM + 2MA + 3H + 1EN	3PM + 2PA + 2MM + 2H + 1IO + 1DE

Based on the information shown in Table 2, it is evident that our proposed scheme and Zhang's system have similar computing costs, making them the two schemes with the most favorable computational costs. However, when evaluating the computational overhead of the KU/Registration process, the utilization of a PKI or MS system necessitates a complicated server installation procedure, a multi-step key calculation/checking procedure, and a database for device logins with a secure transmission using TLS. This results in a significant increase in computational overhead when compared to our suggested scheme.

6 Conclusion

In this paper, we have proposed a lightweight identity-based authenticated signcryption scheme (IB-SCECC) over elliptic curves that integrates an efficient identity-based key generation mechanism with the Ascon cipher. The scheme effectively eliminates the heavy reliance on traditional PKI systems by enabling direct public key exchange and verification through users' identity information. Security analysis demonstrates that our proposed scheme fully satisfies essential security requirements including confidentiality, integrity, user authentication, unforgeability, non-repudiation, forward security, and internal security. Performance evaluation shows that the computational overhead of the scheme is competitive and even lower than several existing elliptic curve signcryption schemes while offering much simpler key management. The proposed IB-SCECC scheme is designed to be easily applied to the S/MIME protocol, providing secure end-to-end email communication in enterprise network environments. In future work, we plan to further optimize the implementation and extend the scheme to other network services and IoT devices.

References

1. Zheng, Y.L.: Digital signcryption or how to achieve cost (signature) + cost (encryption). In: CRYPTO 1997. LNCS, vol. 1294, pp. 165–179, Springer, Berlin (1997). https://doi.org/10.1007/BFb0052234

2. Pandit, T., et al.: Attribute-based signcryption: signer privacy, strong unforgeability and IND-CCA2 security. In: ProvSec 2014. LNCS, vol. 8782, pp. 274–290 (2014). https://doi.org/10.1007/978-3-319-12475-9_19
3. Zhang, X., et al.: Revocable identity-based signcryption scheme without random oracles. J. Network Secur. **17**(2), 110–122 (2015)
4. Datta, P., et al.: Compact attribute-based encryption and signcryption for general circuits from multilinear maps. In: INDOCRYPT 2015, LNCS, vol. 9462, pp. 3–24 (2015). https://doi.org/10.1007/978-3-319-26617-6_1
5. Kar, J.: Provably secure identity-based aggregate signcryption scheme in random oracles. Int. J. Network Secur. **17**(5), 580–587 (2015)
6. Yi, M.: Efficient identity-based online and offline signcryption with short ciphertext. Int. J. Inf. Secur. **16**, 299–311 (2017)
7. Tsai, C., Su, P.: An ECC-based blind signcryption scheme for multiple digital documents, security and communication networks. Secur. Commun. Netw. **2017**, 1–14 (2017)
8. Bashir, M.Z.U.: Analysis and improvement of some signcryption schemes based on elliptic curve. Doctor of Philosophy, Department of Mathematics, Capital University of Science and Technology Islamabad, p.159 (2022)
9. Zhang, P., Li, Y., Chi, H.: An elliptic curve signcryption scheme and its application. Wirel. Commun. Mob. Comput. **2022**, 7499836 (2022)
10. Dobraunig, C., et al.: Ascon v1.2: lightweight authenticated encryption and hashing. J. Cryptology **34**(3) (2021)
11. Shamir, A.: Identity based cryptosystems and signature schemes. In: Advances in Cryptology–Crypto' 84, LNCS 196, pp. 47–53, Springer, Berlin, Heidelberg (1984). https://doi.org/10.1007/3-540-39568-7_5

Algorithmic Complexity Underpins When Path Information Improves Graph Neural Networks Performance on Molecular Graphs

Abdulrahman Ibraheem[1(✉)], Narsis Kiani[3,4,5], and Jesper Tegner[1,2,5,6,7]

[1] Living Systems Lab, BESE and CEMSE, KAUST, Thuwal, Saudi Arabia
rahmanoladi@yahoo.com, jesper.tegner@kaust.edu.sa
[2] SDAIA-KAUST Center of Excellence in Data Science and AI, Thuwal, Saudi Arabia
[3] Algorithmic Dynamics Lab, Karolinska Institutet, Solna, Sweden
narsis.kiani@ki.se
[4] Department of Oncology-Pathology, Karolinska Institutet, Solna, Sweden
[5] Center for Molecular Medicine, Karolinska Institutet, Solna, Sweden
[6] Karolinska University Hospital, Stockholm, Sweden
[7] Science for Life Laboratory, Solna, Sweden

Abstract. Graph Neural Networks (GNNs) are designed to process irregular relational data in recommendation systems, protein networks, social networks, and molecules. GNNs typically rely on message passing and aggregation, with some architectures incorporating graph path information in a bid to improve accuracy. However, it is unclear whether such incorporation of path information truly improves GNN accuracy in all cases. As a first step, we herein shed light on this issue for the case of molecular graphs. We evaluated Graphormer and Mix-Hop models, with and without path information on 36 molecular datasets, derived from six MoleculeNet benchmark datasets. Path information improved performance in some cases but not in other cases. This finding is important, because these two models always incorporate path information in practice, whereas the finding shows such incorporation of path information might actually be detrimental to the models' accuracies in some cases. To more deeply probe this observation, we developed a graph representation model called T-hop which allows us to further highlight the use, versus non-use, of path information. On one hand, we formulate the Path Usefulness Measure (PUM) to quantify the benefit of path information. On the other hand, we quantified the randomness of the different datasets via their algorithmic complexities, using the Block Decomposition Method (BDM). We hypothesized, and confirmed our hypothesis, that: GNN models trained on molecular datasets with less random structures (i.e. lower algorithmic complexity) should benefit more from path information (i.e. larger PUM), compared to datasets with more random structures. Put in other words, we found an inverse relationship between algorithmic complexity and the PUMs.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 18–34, 2026.
https://doi.org/10.1007/978-3-032-30491-9_2

Keywords: GNN · Algorithmic Complexity · Molecular Graphs

1 Introduction

Currently, a prominent class of models for processing graphs comprises graph neural networks (GNN) [1–4]. While most GNNs follow a message passing paradigm [13,14], some GNNs, such as graphormer [5] and Mix-Hop [7], incorporate path information, in a bid to further improve accuracy. In these models, the underlying assumption is that incorporation of graph path information should always improve models' accuracies. However, as we shall see in this work, this assumption can be wrong, because our results show that path information improves accuracies in some cases, while lowering accuracies in other cases.

In this work, we take the first steps towards characterizing conditions under which path information can improve GNN accuracies. To this end, we first examined two established GNN models: Graphormer and Mix-Hop. For each model, we designed two modes: a first mode that jettisons path information altogether and a second mode that utilizes path information. We then compared the two modes on six MoleculeNet graph datasets [11]. We found that path information did not consistently improve performance in these models. To study this phenomenon more closely, we designed a novel model, which we call T-Hop, which more aggressively amplifies the dichotomy between the use of path information on the one hand and its non-use on the other hand. Again, we found that using path information with the T-Hop model also did not improve accuracy in all cases. This prompted us to search for a deeper understanding of the reason behind these counter-intuitive results. We conjectured that the extent to which a graph dataset benefits from path information in GNN models might be inversely correlated with the level of randomness in the dataset's graphs' path connectivity patterns. To test this conjecture, we created six dataset families. Each data set family comprises one of six original MoleculeNet datasets along with five synthetic variants of the original data set. On one hand, we quantified the randomness in the dataset families' graph structures via algorithmic complexity [8,9], using the BDM [8,10]. On the other, we quantified the benefit of path information in the GNN models via the PUM score, which we introduce later in this work. We found that GNN models trained on dataset families with lower algorithmic complexities benefited more from path information and hence have higher PUMs. We underscored this observation by showcasing negative Pearson correlations between algorithmic complexities and the PUMs, thereby corroborating our conjecture. This was further consolidated by our clustering analysis, which showed good matches between the dataset clusters created using the PUMs and those created using the algorithmic complexities.

2 Related Work

To the best of our knowledge, there is no existing work in the literature that characterizes conditions under which path information is expected to improve

GNN accuracies, albeit a plethora of GNN models that incorporate path information in the same literature [5,7,16,17]. For example, the popular graphormer model [5] incorporates path information by utilizing information along the shortest path between every pair of nodes in the input graph and augmenting its attention matrices with that information. Another GNN model that incorporates path information is the Mix-Hop model [7]. It does so by utilizing a set of powered adjacency matrices whose entries encode path information. Yet another model that uses path information, although to a lesser extent than Mix-Hop, is the IGCN (Improved Graph Convolutional Network) model [16]. IGCN uses a powered symmetrically-normalised identity-shifted adjacency matrix, which can ultimately be expressed in terms of the powered adjacency matrix. Furthermore, similarly to Mix-Hop, Power-Up [17] is also a GNN model that utilizes path information. It does so by using a set of matrices that encode information about the path length of the shortest distance between all pairs of nodes in the graph in question. In addition, to incorporate path information, a line of work [18–20], explicitly considers the sequence of paths leading to the nodes of a graph. Other works incorporating path information include NBFNet (Neural Bellman-Ford Networks) [21] and methods that can be considered as its special cases [23–25].

Some earlier works already indicate that adding path information to GNNs does not always improve accuracy [16,20,27]. For example, for PathNN [20], while increasing path length improved results on the NCI1 data set of the TUDataset collection [26], it failed to improve results using the PROTEINS and ENZYMES data sets of the same collection. Likewise, for IGCN [16], experiments on the AWA2 dataset [28] showed worse performance as k increased from 1 to 3, where k can be viewed as a parameter that controls the maximum path length used in the model. Moreover, even for the more recent GraphMixer [27], it was found that using a larger receptive field, and hence path length, diminished accuracies. However, these earlier works neither asked nor answered the question of under which conditions graph path information can be expected to increase GNN accuracies. Hence, this work.

3 Methods

In this section, we outline our methodology vis-à-vis descriptions of: utilized datasets, algorithmic complexity and employed GNN models.

3.1 Datasets

We used a total of thirty six molecular datasets – six original, non-synthetic MoleculeNet datasets [11] along with thirty additional synthetic datasets derived from the original six. Each MoleculeNet dataset comprises a list of molecules, represented via SMILES (Simplified Molecular Input Line Entry System) [31], along with corresponding properties of interest, such as solubility, blood-brain permeability, toxicity, etc. Table 1 summarizes the six original datasets employed.

Table 1. Description of datasets

DATASET	TOTAL NO.OF GRAPHS	PROPERTY	TASK TYPE	METRIC
FreeSolv	643	hydration free energy	regression	RMSE
ESOL	1128	solubility in water	regression	RMSE
Lipophilicity	4200	octanol/water distribution coeff.	regression	RMSE
BACE	1522	β-secretase binding	classification	ROC-AUC
BBBP	2053	blood-brain barrier crossing	classification	ROC-AUC
ClinTox	1491	clinical trial toxicity and FDA approval	classification	ROC-AUC

For processing the graphs in our datasets, we harnessed Deep Graph Library (DGL) [30] and DGL-LifeSci [29]. DGL-LifeSci helped convert MoleculeNet's SMILES (Simplified Molecular Input Line Entry System) [31] representations into DGL graph representations, which are based on a more intuitive list of edges graph representation (LOEGR). In the LOEGR, the graph's nodes correspond to the atoms of the underlying molecule, while the edges of the graph correspond to the bonds between the atoms. This conversion is illustrated in Fig. 1 for the case of methanoic acid. As illustrated by the top-most matrix in the figure, DGL's LOEGR of a graph is implemented as a Pytorch 2-d matrix, wherein each column of the 2-d matrix represents an edge of the graph. Furthermore, the figure also shows that the duo of DGL-LifeSci and DGL makes it possible to easily compute node and edge features for the graph. These are respectively represented by the middle and bottom matrices in the figure. The fact that DGL's graph representations are implemented as Pytorch matrices makes them especially suitable as inputs to the Pytorch implementations of our models.

For our experiments, we required a collection of datasets that share the same path connectivity information, but with different node and edge features. To this end, we defined *dataset families*. A dataset family comprises an original non-synthetic graph dataset, along with a collection of synthetic graph datasets derived from the original dataset. This derivation is done via the addition of pre-defined levels of Gaussian noise to the node and edge features of the graphs in the original dataset, without mutating the path connectivity patterns within those graphs. Note that this means that **datasets within the same dataset family share the same set of path connectivity patterns**. To create the dataset families, we started with the six original non-synthetic MoleculeNet datasets mentioned in the preceding paragraph. We then introduced a noise-level parameter, $\gamma \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. Further, we let σ_0 denote the standard deviation of an original non-synthetic dataset's features. Then for each original dataset, we computed the set, $S_{\sigma_0} = \{0.0\sigma_0, 0.1\sigma_0, 0.2\sigma_0, 0.3\sigma_0, 0.4\sigma_0, 0.5\sigma_0\}$. To each original dataset, we then added zero-mean Gaussian noises with standard deviations $\sigma \in S_{\sigma_0}$ to the dataset's features. This yielded five synthetic variants of each original dataset. Here, note that $\sigma = 0.0\sigma_0$ is the noise-less case, which actually corresponds to the original non-synthetic dataset. In all, the above procedure allowed us to obtain a total of thirty six datasets: thirty synthetic datasets along with the six original non-synthetic datasets. We used scaffold splitting [11]

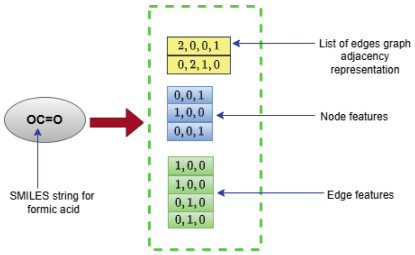


Fig. 1. DGL/DGL-LifeSci conversion of methanoic acid's SMILES representation to a graph representation comprising: 1). a list of edges representation using a Pytorch 2-d matrix with each column representing an edge (top matrix in the figure); 2) a Pytorch 2-d matrix containing the graph's nodes' features (middle matrix in the figure); 3) a Pytorch 2-d matrix containing the graph's edges' features (bottom matrix in figure).

to split the datasets into training:validation:test segments in the ratio 80:10:10. Finally, we point out that each of the 30 synthetic datasets inherit properties listed in Table 1 from its parent non-synthetic dataset.

3.2 Algorithmic Complexity and BDM

We required a way to measure the randomness (i.e. lack of regularity) of the connectivity patterns within the graphs in our datasets. We used algorithmic complexity [8,9] as a measure of this randomness, and employed the BDM [8,10] for approximating it. We favored algorithmic complexity over traditional measures of randomness, such as Shanon entropy, because the former has been demonstrated to be better at capturing the intrinsic information content of a graph's structure, beyond sheer statistical patterns [8]. In BDM [8,10], the adjacency matrix, A, of the input graph, G, is first partitioned into a set of non-overlapping $d \times d$ sub-arrays. The j-th sub-array is denoted r_j. Then, the Coding Theorem Method (CTM) [8] is applied, to compute the algorithmic complexity, $K_{CTM}(r_j)$, of r_j. This involves computing the algorithmic probability of r_j. To compute the algorithmic probability, a large collection of 2-d Turing machines (i.e. turmites) is run. The frequency with which r_j is output by the collection is then computed, and used to approximate $K_{CTM}(r_j)$. In symbols, the BDM algorithmic complexity of G, denoted $K_{BDM}(G, d)$, is expressed in terms of $K_{CTM}(r_j)$ as:

$$K_{BDM}(G, d) = \sum_{(r_j, n_j) \in A} K_{CTM}(r_j) + \log_2(n_j) \tag{1}$$

Above, n_j is the number of times r_j occurs within A. The notation $\sum_{(r_j,n_j)\in A}$, as opposed to $\sum_{r_j \in A}$, serves to emphasize that we count $K_{CTM}(r_j)$ separately for each of the n_j occurrences of r_j within A. To eschew re-inventing the wheel, we leveraged the publicly available implementation of BDM at this URL[1]

[1] https://pybdm-docs.readthedocs.io/en/latest/pybdm.html.

3.3 Models

This section describes the three models employed in this study, namely Graphormer [5], Mix-Hop [7] and T-Hop. The T-Hop model is a tensor-based graph representation model which we introduce in this work. We introduce it as a more effective model for highlighting the dichotomy between the use of path information versus the non-use of path information in GNNs.

Synopsis of the Graphormer Model. Here, we describe how to use the graphormer model for the goals of this work. For a full account of the graphormer model, the interested reader is referred to the graphormer paper [5]. Essentially, Graphormer can be viewed as an analog of the popular transformer model [6] applied to graph datasets. At each layer of graphormer, the transformer-style softmax attention mechanism is applied to every pair of nodes in the input graph; followed by a skip-connection and the LayerNorm; followed by a feedforward MLP; followed again by a skip-connection and LayerNorm. In addition, the graphormer model super-imposes the structure of the underlying graph into the model by augmenting the softmax attention matrix with path information. The path information here is derived by considering the distance along the shortest path between every pair of nodes, as well as by considering the flow of edge features along those shortest paths. Consequently, for our experiments in this paper, we implemented two modes of the graphormer model: a first mode, wherein the softmax attention matrix is not augmented with any path information; and a second mode wherein the softmax attention matrix is augmented with path information as above described. Our Pytorch implementation is at this URL[2].

Synopsis of the Mix-Hop Model. The Mix-Hop model [7] harnesses powered adjacency matrices, $A, A^2, \ldots, A^{L_m}$, with L_m being the maximum path length considered in the model. Specifically, where $\|_{L=1}^{L_m}$ denotes the vector concatenation operation; where H^l denotes input feature matrix to the l-th layer; where W_L^l denotes a learnable matrix; and where σ denotes an activation function, then each layer of Mix-Hop is underpinned by the following formula:

$$H^{l+1} = \Big\|_{L=1}^{L_m} \sigma(A^L H^l W_L^l) \tag{2}$$

Since L_m denotes maximum path length, it follows that we can implement a mode that does not incorporate path information into the model by setting $L_m = 1$. Also, we can implement a mode that incorporates path information into the model by setting $L_m > 1$. Our Pytorch implementations are at this URL[3]

[2] https://github.com/rahmanoladi/kaust_path_project/tree/main/path_project/graphormer.

[3] https://github.com/rahmanoladi/kaust_path_project/tree/main/path_project/mix_hop.

The T-Hop Model: A Novel Tensor-Based Approach. This section introduces T-Hop, a model that sharply contrasts performance with and without path information. Let $G = (V, E)$ represent a graph, where $V = \{v_1, ..., v_n\}$ and $E = \{e_1, ..., e_m\}$ are the nodes and edges of G, as usual. The adjacency matrix of G is A. We consider two arbitrary nodes, v_i and v_j in G. Let $L \geq 2$, and let $\mathcal{B}^L_{i,j,k}$ be the number of paths of length L between v_i and v_j that contain v_k. We may arrange the values, $\mathcal{B}^L_{i,j,k}$, in an $n \times n \times n$ 3-d tensor denoted $\mathcal{B}^L \in \mathbb{R}^{n \times n \times n}$, such that the entry on the i-th row, j-th column and k-th depth of $\mathcal{B}^L$ is $\mathcal{B}^L_{i,j,k}$. Note that the tensor, $\mathcal{B}^L$, encapsulates path information contained in the input graph, G. Further, we scale $\mathcal{B}^L$ by $L + 1$, and denote the result as a new tensor, $\mathcal{T}^L \in \mathbb{R}^{n \times n \times n}$:

$$\mathcal{T}^L_{i,j,k} = \frac{\mathcal{B}^L_{i,j,k}}{(L+1)} \tag{3}$$

Note that, like the tensor, $\mathcal{B}^L$, the tensor $\mathcal{T}^L$ also encapsulates path information contained in the input graph, G. To proceed, we define a new $n \times n$ matrix $\mathcal{M}$ as follows:

$$\mathcal{M} = \alpha_0 A + \sum_{L=2}^{L_m} \sum_{k=0}^{n-1} \alpha_{L,k} \mathcal{T}^L_{:,:,k} \tag{4}$$

Above, A is adjacency matrix, while α_0 and $\alpha_{L,k}$ are learnable scalars. Further, L_m is the maximum path length considered in the model. Also, $\mathcal{T}^L_{:,:,k} \in \mathbb{R}^{n \times n}$ is the 2-d matrix slice associated with depth-k of $\mathcal{T}^L$. Using $\mathcal{M}$, the l-th layer of our T-Hop model can now be described as:

$$H^{l+1} = \sigma(\mathcal{M}\, H^l W^l) \tag{5}$$

Above, H^l denotes input features to the l-th layer, W^l denotes a learnable matrix of weights, and $\sigma(.)$ is a non-linear activation function. Going back to Eq. 4, we can implement a mode of the model that does not use path information by setting $\alpha_{L,k} = 0$ for all $L = 2, \ldots, L_m$ and all $k = 0, \ldots, n-1$. Under this mode, Eq. 4 distills to $\mathcal{M} = \alpha_0 A$, which shows that the model is left only with information contained in the adjacency matrix, A, to the exclusion of all the path information contained in the tensor, $\mathcal{T}^L$. On the other hand, to implement a mode that uses path information, we simply use Eq. 4 as it is. We juxtapose the two modes pictorially in Figs. 2a and 2b, for the simple case of $L_m = 2$. In Fig. 2a, we see that the diagram includes a cube representing $\mathcal{T}^L$, which encapsulates path information; whereas in Fig. 2b, the cube is not included. A Pytorch implementation of the models is at this URL.[4]

3.4 Path Usefulness Measure: A Quantifier of the Usefulness of Incorporating Path Information in Our Models

This subsection introduces the PUM, a metric designed to *quantify the extent to which incorporating path information into a given GNN model enhances its*

[4] https://github.com/rahmanoladi/kaust_path_project/tree/main/path_project/t_hop.

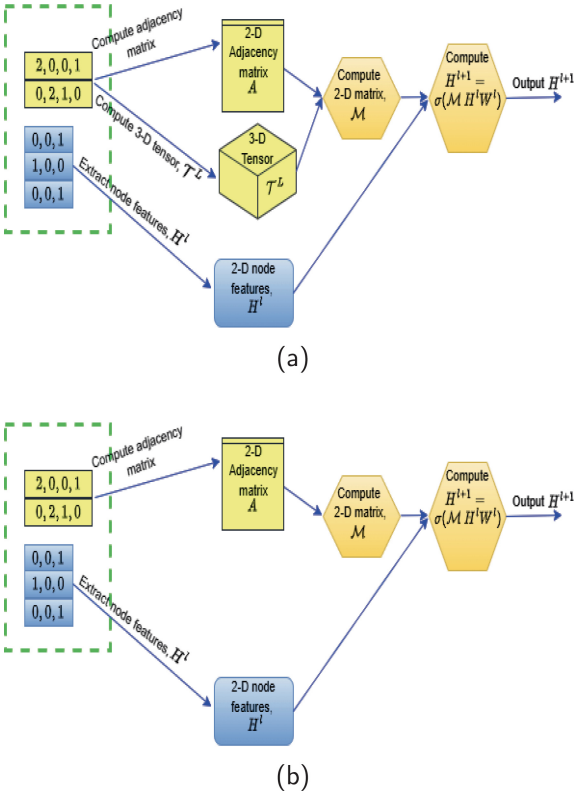


Fig. 2. Schematic depiction of the two modes of T-Hop model for the case of $L_m = 2$: (a) using path information contained in tensor $\mathcal{T}^L$ as represented by the cube in the diagram (b) *without* using path information contained in tensor $\mathcal{T}^L$ (so no cube is included in this case).

accuracy on a specific dataset family. Consider a set of K GNN models, $\mathcal{M} = M_1, M_2, \ldots, M_K$, and a collection of N dataset families, $\mathcal{D} = D_1, D_2, \ldots, D_N$. Each dataset family D_j includes an original dataset along with several synthetic variants generated by adding noise. For each variant $D_j^\gamma \in D_j$, associated with a noise level γ, we evaluate whether the inclusion of path information improves the model's accuracy. Define an indicator function $\mathcal{F}(M_i, D_j^\gamma)$ such that:

$$\mathcal{F}(M_i, D_j^\gamma) = \begin{cases} 1, & \text{if incorporating path information improves accuracy} \\ 0, & \text{otherwise} \end{cases}$$

PUM for a given model M_i and dataset family D_j is then:

$$U_{ij}(M_i, D_j) = \frac{\sum_{D_j^\gamma \in D_j} \mathcal{F}(M_i, D_j^\gamma)}{|D_j|} \tag{6}$$

where $|D_j|$ is the total number of variants in the family. This score represents the fraction of variants within the dataset family for which path information led to improved accuracy. In essence, PUM estimates the probability that incorporating path information enhances model performance for a particular model-dataset pair, providing a normalized measure to compare across datasets and models.

4 Experiments

Having established the theoretical foundations, we now experimentally explore how the use of graph path information impacts the accuracies of GNNs. Our results show that path information does not consistently boost GNN accuracies. Moreover, our results show an inverse correlation between the benefit of using path information and the level of randomness/irregularities in the graphs.

4.1 Incorporating Path Information Does Not Yield Consistent Gains in Accuracies on the Original Six Datasets

In this subsection, we report our experimental results assessing how useful including graph path information is for enhancing GNN performance on the six non-synthetic molecular datasets introduced in Sect. 3.1. We consider three model types discussed in Sect. 3 and two modes per model—one without path information and one with—resulting in a total of $6 \times 3 \times 2 = 36$ experimental configurations. For each setup, we used the Optuna library [32] to perform hyper-parameter sweeps on the validation subset to identify optimal parameters. For each of the three model types, details of the Optuna hyper-parameters associated with the model can be found under the directory associated with the model on this project's github[5]. Using these hyper-parameters, we trained each model for 200 epochs on the corresponding training set, then evaluated performance on the test set, recording the relevant metric: RMSE for regression tasks and ROC-AUC for classification tasks. To ensure robustness, each experiment was run three times, and we report the mean and standard deviation of the results. These are shown in Table 2, with the standard deviations placed in parentheses beneath the associated means. Analyzing these results across all configurations, it becomes clear that adding path information does not yield consistent improvements. The additional computational cost therefore is not always justified. Also, this shows that a section of the users of the Graphormer and Mix-Hop models might be able to improve the accuracies of their models simply by switching off the use of path information in those models.

4.2 Measuring PUM on the Models and Dataset Families

Experiments from the previous sub-section showed that incorporating path information does not consistently lead to gains in performance on the three models

[5] https://github.com/rahmanoladi/kaust_path_project.

Table 2. Comparison of Models with and without path information across all six original datasets. Path information does not help in all Cases.

DATASET	RMSE (Lower is better)						ROC-AUC (Higher is better)					
	FREESOLV		ESOL		LIPOPHILICITY		BACE		BBBP		CLINTOX	
	WITHOUT PATH	WITH PATH	WITHOUT PATH	WITH PATH	WITHOUT PATH	WITH PATH	WITHOUT PATH	WITH PATH	WITHOUT PATH	WITH PATH	WITHOUT PATH	WITH PATH
Graphormer												
	3.55 (0.4539)	**2.91** (0.1556)	1.20 (0.0762)	**1.01** (0.0018)	**0.91** (0.0147)	0.99 (0.0357)	**0.782** (0.0002)	0.735 (0.0283)	0.686 (0.0172)	**0.709** (0.0089)	0.602 (0.0128)	**0.779** (0.0065)
Mix-Hop												
	2.45 (0.0714)	2.61 (0.1790)	1.01 (0.0138)	**0.97** (0.0373)	0.96 (0.0024)	**0.95** (0.0105)	**0.799** (0.0047)	0.798 (0.0018)	**0.707** (0.0201)	0.678 (0.0088)	**0.596** (0.0098)	0.591 (0.0228)
T-Hop												
	2.87 (0.3241)	**2.73** (0.0385)	**1.00** (0.0330)	1.00 (0.0367)	**0.78** (0.0210)	0.88 (0.0210)	**0.852** (0.0087)	0.746 (0.0346)	**0.649** (0.0032)	0.623 (0.0363)	**0.785** (0.0079)	0.752 (0.1003)

and on the six original molecular datasets considered. Yet, the experiments did not provide any quantification of the extent to which path information helped. Recall from Sect. 3.4 that the PUM gives us a criterion for quantifying the extent to which path information helps on a given dataset family, with respect to a given model. Consequently, in this section, we will describe experiments performed to obtain PUMs for the various combinations of (model, dataset family) pairs relevant to this work. Specifically, for our experiments here, we used all the thirty six synthetic and non-synthetic molecular datasets within all the six dataset families listed in the earlier sub-section on datasets (i.e. Section 3.1). However, since we already dealt with the case of the six original non-synthetic datasets in the preceding sub-section, then for those cases, we simply copied the relevant results from the preceding sub-section. The remaining thirty synthetic datasets spawned a total of $30 \times 3 \times 2 = 180$ experimental cases, because we trained three different model types (i.e. Graphormer, Mix-Hop and T-hop) at two different modes (i.e. without path information and with path information) on each of the thirty synthetic datasets. Ideally, we should perform a separate hyper-parameter sweep for each of the 180 cases, but that would be too computationally expensive. Hence, we reused the hyper-parameters from the original datasets for the synthetic variants to reduce computation. Specifically, suppose e_n represents an experimental case associated with a non-synthetic dataset, and e_s represents an experimental case associated with a synthetic dataset. We applied the hyper-parameters associated with e_n to e_s as long as e_s and e_n share the same model type, dataset family and mode (mode here means whether the experimental case involves the use of path information or not). Based on this, for each of the 180 synthetic-dataset experimental cases, we trained the pertinent model associated with the case for 200 epochs on the training segment of the associated dataset, and used the trained model to obtain scores of a relevant accuracy metric (R.M.S.E for regression datasets and ROC-AUC for classification datasets) on the test segment of the dataset. To address statistical fluctuations, we actually ran each experimental case thrice, and computed the mean and standard deviation of the accuracies of the three runs. We present the means and standard deviations in Table 3, where the standard deviations are shown in parentheses

beneath the associated means; for ease of reference, the table also copies results for the cases associated with the non-synthetic datasets from Table 2. In addition to the R.M.S.E and ROC-AUC for each experimental case, the table also displays the PUMs associated with each dataset family and model. Because there are six datasets within each dataset family, the PUMs lie in the range $[\frac{0}{6}, \frac{6}{6}]$. A PUM of $\frac{6}{6}$ indicates complete certainty that the given model always yields better results with path information on the given dataset family, while a PUM of $\frac{0}{6}$ indicates complete certainty that the given model yields better results without path information on the given dataset family. Now, we see from the table that, most of the times, the PUMs are neither equal to $\frac{6}{6}$ nor equal to $\frac{0}{6}$, which shows that, even within a single dataset family, it is difficult to predict whether a given model will benefit from the incorporation of path information. Nonetheless, a closer look at the PUMs for each model type shows that the values associated with the T-Hop model are either very close to $\frac{6}{6}$ (i.e. $\frac{5}{6}$ on FreeSolv dataset family) or very close to $\frac{0}{6}$ (e.g. $\frac{1}{6}$ on ESOL family and $\frac{0}{6}$ on Lipophilicity family). This shows that, unlike the other models, T-Hop allows us to more easily distinguish between when path information is helpful versus when it is not helpful. In other words, T-Hop is more effective at highlighting the dichotomy between dataset families that benefit from path information versus those that do not. To quantify this observation, we may compute a *dichotomy score*, Φ_i, for the i-th model according to $\Phi_i = \frac{1}{6}\sum_{D_j \in \mathcal{D}} \max\{U_{ij}, 1 - U_{ij}\}$. Here, U_{ij} is PUM of applying model M_i on dataset family D_j, and $\mathcal{D}$ denotes our collection of six dataset families. Using this confidence score, we obtained scores of $\frac{29}{36}$, $\frac{24}{36}$ and $\frac{33}{36}$ for the Graphormer, Mix-Hop and T-Hop models respectively. We see that T-Hop very closely approaches a perfect score of 1.

4.3 PUM Shows Inverse Correlation with Algorithmic Complexity

A centerpiece of the results from the preceding sub-sections is that it is difficult to predict with certainty when path information can be expected to help for a given model and molecular dataset family. Nonetheless, having a means of making such a prediction would be truly beneficial for deploying GNNs in a more resource-efficient manner. Towards this, we hypothesized that the less random the connectivity pattern of a molecular graph is, the more the path information contained in that graph can be expected to boost the accuracies of GNN models. Since algorithmic complexity is a measure of randomness in graphs [9], this hypothesis implies that we expect an inverse correlation between the algorithmic complexity of our dataset families, and the PUMs associated with the families. In this sub-section, we will empirically validate this hypothesis from two perspectives. The first perspective will employ Pearson correlations between PUM and algorithmic complexity, while the second will use clustering analysis. From our work in the previous sub-section, we already have all the required PUMs. Thus, we now turn to how to obtain the algorithmic complexities for our six dataset families. To obtain them, we used BDM [8,10]. Here, we point out that all members of the same dataset family share similar values of algorithmic complexity. This is because algorithmic complexity only examines the connectivity patterns

Table 3. PUMs for all model types and dataset families. It is difficult to use the PUMs to predict whether or not path information will boost accuracies.

MODEL	RMSE (Lower is better)						ROC-AUC (Higher is better)					
	FREESOLV		ESOL		LIPOPHILICITY		BACE		BBBP		CLINTOX	
	WITHOUT PATH	WITH PATH	WITHOUT PATH	WITH PATH	WITHOUT PATH	WITH PATH	WITHOUT PATH	WITH PATH	WITHOUT PATH	WITH PATH	WITHOUT PATH	WITH PATH
Graphormer												
$\gamma = 0$	3.55 (0.4539)	**2.91** (0.1556)	1.20 (0.0762)	**1.01** (0.0018)	**0.91** (0.0147)	0.99 (0.0357)	**0.782** (0.0002)	0.735 (0.0283)	0.686 (0.0172)	**0.709** (0.0089)	0.602 (0.0128)	**0.779** (0.0065)
$\gamma = 0.1$	4.219 (0.3890)	**3.54** (0.0853)	1.23 (0.1130)	**1.19** (0.1868)	**1.02** (0.0474)	1.05 (0.0132)	**0.779** (0.0033)	0.761 (0.0340)	**0.693** (0.0160)	0.661 (0.0321)	**0.594** (0.0094)	0.582 (0.0064)
$\gamma = 0.2$	4.16 (0.3905)	**3.42** (0.0912)	1.31 (0.0743)	**1.16** (0.0750)	**1.03** (0.0559)	1.04 (0.0203)	**0.781** (0.0028)	0.767 (0.0283)	**0.687** (0.0122)	0.653 (0.0264)	**0.594** (0.0100)	0.575 (0.0080)
$\gamma = 0.3$	4.24 (0.3831)	**3.44** (0.0721)	1.36 (0.0956)	**1.22** (0.0803)	**1.02** (0.0262)	1.06 (0.0298)	**0.780** (0.0013)	0.768 (0.0267)	**0.688** (0.0148)	0.663 (0.0341)	**0.580** (0.0141)	0.578 (0.0046)
$\gamma = 0.4$	4.02 (0.4484)	**3.47** (0.0434)	**1.34** (0.1136)	1.51 (0.2614)	**1.03** (0.0312)	1.06 (0.0165)	0.776 (0.0047)	**0.783** (0.0089)	**0.684** (0.0157)	0.636 (0.0245)	0.579 (0.0240)	**0.580** (0.0053)
$\gamma = 0.5$	4.20 (0.7253)	**3.49** (0.0349)	**1.41** (0.2020)	1.62 (0.1524)	**1.00** (0.0391)	1.07 (0.0122)	0.777 (0.0037)	**0.781** (0.0059)	**0.670** (0.0122)	0.641 (0.0206)	**0.581** (0.0176)	0.577 (0.0052)
PUM		PUM= $\frac{6}{6}$		PUM= $\frac{4}{6}$		PUM= $\frac{0}{6}$		PUM= $\frac{2}{6}$		PUM= $\frac{1}{6}$		PUM= $\frac{2}{6}$
Mix-Hop												
$\gamma = 0$	**2.45** (0.0714)	2.61 (0.1790)	1.01 (0.0138)	**0.97** (0.0373)	0.96 (0.0024)	**0.95** (0.0105)	**0.799** (0.0047)	0.798 (0.0018)	**0.707** (0.0201)	0.678 (0.0088)	**0.596** (0.0098)	0.591 (0.0228)
$\gamma = 0.1\%$	**2.43** (0.0838)	2.82 (0.2120)	**1.19** (0.1675)	1.27 (0.1040)	1.05 (0.0052)	**1.02** (0.0117)	0.737 (0.0195)	**0.780** (0.0066)	**0.671** (0.0094)	0.656 (0.0215)	**0.581** (0.0690)	0.559 (0.0440)
$\gamma = 0.2\%$	2.87 (0.3401)	**2.73** (0.2040)	1.28 (0.1640)	**1.23** (0.0140)	1.16 (0.0377)	**1.11** (0.0416)	0.671 (0.0833)	**0.691** (0.0734)	**0.657** (0.0274)	0.629 (0.1393)	**0.589** (0.0148)	0.539 (0.0329)
$\gamma = 0.3\%$	**3.16** (0.2525)	3.23 (0.5875)	1.48 (0.2173)	**1.41** (0.0625)	1.23 (0.0448)	1.23 (0.0763)	0.674 (0.0400)	**0.752** (0.0417)	0.602 (0.0198)	**0.614** (0.0230)	**0.578** (0.0367)	0.555 (0.0263)
$\gamma = 0.4\%$	3.74 (0.1838)	**3.18** (0.2319)	1.68 (0.2307)	**1.57** (0.0652)	**1.33** (0.0829)	1.39 (0.1044)	0.630 (0.1005)	**0.700** (0.0621)	0.592 (0.0259)	**0.599** (0.0303)	**0.627** (0.0057)	0.463 (0.0334)
$\gamma = 0.5\%$	3.89 (0.1518)	**3.32** (0.2235)	1.83 (0.3599)	**1.63** (0.1241)	**1.45** (0.1081)	1.51 (0.1752)	**0.676** (0.0268)	0.663 (0.0423)	0.567 (0.0310)	**0.603** (0.0269)	**0.550** (0.0619)	0.493 (0.0641)
PUM		PUM= $\frac{3}{6}$		PUM= $\frac{5}{6}$		PUM= $\frac{3}{6}$		PUM= $\frac{4}{6}$		PUM= $\frac{3}{6}$		PUM= $\frac{0}{6}$
T-Hop												
$\gamma = 0$	2.87 (0.3241)	**2.73** (0.0385)	**1.00** (0.0330)	1.00 (0.0367)	**0.78** (0.0210)	0.88 (0.0210)	**0.852** (0.0087)	0.746 (0.0346)	**0.649** (0.0032)	0.623 (0.0363)	**0.785** (0.0079)	0.752 (0.1003)
$\gamma = 0.1$	3.68 (0.7068)	**3.07** (0.0387)	**1.09** (0.0307)	1.32 (0.0524)	**0.82** (0.0052)	1.01 (0.0090)	**0.819** (0.0167)	0.764 (0.0136)	**0.645** (0.0066)	0.618 (0.0077)	**0.826** (0.0061)	0.681 (0.0367)
$\gamma = 0.2$	**3.34** (0.1859)	3.35 (0.2472)	**1.26** (0.1582)	1.30 (0.0215)	**0.88** (0.0383)	1.00 (0.0078)	**0.827** (0.0228)	0.770 (0.0033)	**0.642** (0.01160)	0.620 (0.0262)	**0.806** (0.0023)	0.591 (0.0518)
$\gamma = 0.3$	3.80 (0.8851)	**3.34** (0.3717)	**1.22** (0.0781)	1.34 (0.0678)	**0.92** (0.0241)	1.01 (0.0525)	**0.804** (0.0251)	0.687 (0.0718)	**0.632** (0.0195)	0.612 (0.0201)	**0.799** (0.0151)	0.627 (0.0567)
$\gamma = 0.4$	3.41 (0.4146)	**3.34** (0.3023)	**1.50** (0.3625)	1.52 (0.1281)	**0.93** (0.0286)	1.02 (0.0142)	**0.804** (0.0219)	0.774 (0.0359)	**0.631** (0.0222)	0.611 (0.0202)	**0.784** (0.0212)	0.667 (0.0466)
$\gamma = 0.5$	3.91 (0.1905)	**3.55** (0.1852)	1.62 (0.1876)	**1.42** (0.1401)	**0.93** (0.0333)	1.04 (0.0137)	**0.789** (0.0044)	0.763 (0.0223)	0.624 (0.0154)	**0.634** (0.0242)	**0.773** (0.0095)	0.538 (0.0673)
PUM		PUM= $\frac{5}{6}$		PUM= $\frac{1}{6}$		PUM= $\frac{0}{6}$		PUM= $\frac{0}{6}$		PUM= $\frac{1}{6}$		PUM= $\frac{0}{6}$

of the underlying graphs, without looking at the node and edge features. Hence, to obtain the algorithmic complexity of a dataset family, D_j, it suffices to compute the algorithmic complexity of its original non-synthetic member, $D_j^0 \in D_j$. Therefore, for each of the six dataset families, we computed the algorithmic complexity of every single molecular graph in its non-synthetic dataset member, D_j^0. We averaged the algorithmic complexity values over all molecular graphs in D_j^0. We display the results in the first row of Table 4, wherein we have arranged the results in ascending order from left to right, thereby obtaining ascending order algorithmic complexities (AOAC). Further, Table 4 also contains the PUMs associated with all three models and all six datasets, copied from Table 3.

For each dataset family, we also computed the average PUM accross all model types; we placed those average PUMs in the *across all models* row of the table. To

Table 4. Exhibition of inverse correlation between AOAC and PUM

	DATASETS' AOAC AND MODELS' PUMS						CORRELATION BETWEEN AOAC and PUM
	FREESOLV	ESOL	BBBP	CLINTOX	LIPOPHILICITY	BACE	
DATASETS' AOAC:	105.61	216.63	488.75	510.71	577.11	717.38	
MODELS' PUMs:							
Graphormer	$\frac{6}{6}$	$\frac{4}{6}$	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{0}{6}$	$\frac{2}{6}$	-0.84
Mix-Hop	$\frac{3}{6}$	$\frac{5}{6}$	$\frac{3}{6}$	$\frac{0}{6}$	$\frac{3}{6}$	$\frac{4}{6}$	-0.19
T-Hop	$\frac{5}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{0}{6}$	$\frac{0}{6}$	$\frac{0}{6}$	-0.81
Across all models	$\frac{14}{18}$	$\frac{10}{18}$	$\frac{5}{18}$	$\frac{2}{18}$	$\frac{3}{18}$	$\frac{6}{18}$	-0.82

validate our hypothesis from a first perspective, we computed the Pearson correlation between the PUMs, listed in Table 4, and the corresponding algorithmic complexities. We display the computed Pearson correlations in the last column of the table. First, according to the correlations in the table, we note that each model showed negative correlation. The graphormer model showed the strongest negative correlation, followed closely by the T-Hop model, with the Mix-Hop model coming third. Moreover, from the table, we also see that the correlation between the average PUMs and their associated algorithmic complexities is also strongly negative (See the *across all models* row of the table). We illustrate these observations graphically in Fig. 3. As shown in the figures's legend, three of the plots are for the three individual models, while the fourth plot is for the average across all models. All four plots in the figure exhibit negative correlations between PUM and algorithmic complexity. Hence, it becomes obvious that low algorithmic complexity of a molecular dataset family is an indicator that GNN models are more likely to benefit from path information on the datasets within the family. This result is reasonable, because if the path connectivity pattern in a graph is to be useful for machine learning prediction tasks, then we should want the pattern to be as far as possible from being random/unpredictable. That is, we should want it to have low algorithmic complexity.

To validate our hypothesis from a second perspective, we performed binary clustering analysis. We first clustered the dataset families into two categories based on their associated algorithmic complexities. We then repeated the clustering based on PUMs instead of algorithmic complexities. The results are shown in Table 5. In the table, we display the algorithmic complexities (and PUMs), and place the cluster labels, in parentheses, next to the algorithmic complexities (and PUMs). For the algorithmic-complexities-based clustering, we used 0 to label dataset families corresponding to low algorithmic complexities, and used 1 to label those corresponding to high algorithmic complexities. However, since we

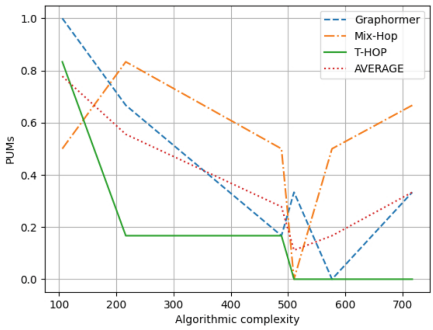


Fig. 3. Plots of PUMs versus algorithmic complexity.

hypothesize an inverse relationship between algorithmic complexity and PUM, we inverted the labelling scheme in the case of the PUM-based clustering. For the PUM-based clustering, we used 0 to label dataset families corresponding to high PUMs, and used 1 to label those corresponding to low PUMs. Our hypothesis is that the dataset-family clustering produced via the algorithmic complexity values, should match with the dataset-family clustering produced via the PUM. By looking at the *across all models* row in Table 5, we find that the cluster labels there perfectly match the cluster labels in the algorithmic complexities row of the table, thereby validating our hypothesis. For individual models, we found that the Graphormer model also agrees with the hypothesis, with the T-Hop model coming next in agreement, and the Mix-Hop model coming third in agreement. The observed disparities in the levels of agreement with the hypothesis are not far fetched, since the models incorporate path information in different ways. In the table, we also included a column that contains the Silhouette scores for the clustering produced by the three models. We found that the T-Hop model got the highest Silhouette score, which is likely to be due to its heavy and aggressive use of all the available path information contained in the graphs, whereas the graphormer model, for example, only considers path information along shortest paths between graph nodes.

4.4 Computational Run-Times with and Without Path Information

We juxtapose the compute times required for our models without using path information versus using path information. The single-epoch training times, in seconds, for all six non-synthetic datasets, across all three models are shown in Table 6. As can be clearly seen from the table, in all cases, running the models without path information is more efficient, as expected. For each model type, the model sizes were kept constant across all datasets. Table 7 shows the pertinent hyper-paramaters defining the model sizes. We listed "None" as the maximum path length of the Graphormer model because the model considers all pair-wise node interactions in its attention mechanism, regardless of how far apart the nodes are in the underlying graph topology.

Table 5. Two ways of clustering dataset families: 1) based on algorithmic complexities and 2). based on PUMs. Clusterings produced via the two ways match remarkably.

	DATASETS' AOAC, MODELS' PUMS AND CLUSTER LABELS IN PARENTHESES						SILHOUETTE SCORES
	FREESOLV	ESOL	BBBP	CLINTOX	LIPOPHILICITY	BACE	
DATASETS' AOAC & CLUSTER LABELS:	105.61 **(0)**	216.63 **(0)**	488.75 **(1)**	510.71 **(1)**	577.11 **(1)**	717.38 **(1)**	0.71
MODELS' PUMs & CLUSTER LABELS:							
Graphormer	$\frac{6}{6}$ **(0)**	$\frac{4}{6}$ **(0)**	$\frac{1}{6}$ **(1)**	$\frac{2}{6}$ **(1)**	$\frac{0}{6}$ **(1)**	$\frac{2}{6}$ **(1)**	0.60
Mix-Hop	$\frac{3}{6}$ **(0)**	$\frac{5}{6}$ **(0)**	$\frac{3}{6}$ **(0)**	$\frac{0}{6}$ **(1)**	$\frac{3}{6}$ **(0)**	$\frac{4}{6}$ **(0)**	0.61
T-Hop	$\frac{5}{6}$ **(0)**	$\frac{1}{6}$ **(1)**	$\frac{1}{6}$ **(1)**	$\frac{0}{6}$ **(1)**	$\frac{0}{6}$ **(1)**	$\frac{0}{6}$ **(1)**	0.72
Across all models	$\frac{14}{18}$ **(0)**	$\frac{10}{18}$ **(0)**	$\frac{5}{18}$ **(1)**	$\frac{2}{18}$ **(1)**	$\frac{3}{18}$ **(1)**	$\frac{6}{18}$ **(1)**	0.62

Table 6. Comparison of computational run-times with and without path information across all six original datasets.

	FREESOLV		ESOL		LIPOPHILICITY		BACE		BBBP		CLINTOX	
	WITHOUT PATH	WITH PATH	WITHOUT PATH	WITH PATH	WITHOUT PATH	WITH PATH	WITHOUT PATH	WITH PATH	WITHOUT PATH	WITH PATH	WITHOUT PATH	WITH PATH
T-Hop	**0.46**	0.69	**0.70**	6.94	**2.42**	172.43	**0.95**	38.76	**1.52**	133.37	**1.23**	100.32
Graphormer	**0.93**	4.88	**1.80**	25.96	**10.69**	547.71	**3.39**	301.83	**6.23**	191.69	**4.70**	275.44
Mix-Hop	**0.56**	0.65	**0.88**	0.95	**3.23**	3.56	**1.22**	1.35	**2.14**	2.33	**1.69**	1.90

Table 7. Comparison of computational run-times with and without path information across all six original datasets.

	LAYERS	HIDDEN DIM	ATTENTION HEADS	MAX. PATH LENGTH(L_m)	
				WITHOUT PATH	WITH PATH
T-Hop	5	600	None	1	4
Graphormer	5	600	6	None	None
Mix-Hop	5	600	None	1	3

5 Conclusion and Future Work

This work advances our understanding of conditions under which the incorporation of path information into graph neural networks improves accuracies on molecular graph datasets. Specifically, our results demonstrate that molecular graphs with higher algorithmic complexity scores, which reflect more random patterns, show minimal or no benefit from the inclusion of path information, thereby questioning the cost-effectiveness of incorporating such information in these cases. Indeed, future work can parlay this finding into a scheme that conserves computational resources, by not expending the extra computation involved in incorporating path information on datasets that are known to have high algorithmic complexity. This is especially important in large-scale or resource-constrained environments, where high computational costs are pro-

hibitive. Finally, while the current study was limited to the domain of molecular graphs, we believe that the lessons learnt here will spark researchers in other domains to test our hypothesis on graphs occurring within their domains.

References

1. Shen, J., Ain, Q.T., Liu, Y., Liang, B., Qiang, X., Kou, Z.: GTAT: empowering graph neural networks with cross attention. Sci. Rep. **15**(1), 4760 (2025)
2. Wang, J., Hou, K., Chen, H., Fang, J., Li, H.: Dual self-attention fusion message neural network for virtual screening in drug discovery by molecular property prediction. J. Bionic Eng. **22**(1), 354–369 (2025)
3. Yu, Z., Huang, W., Liu, Y.: Rigid protein-protein docking via equivariant elliptic-paraboloid interface prediction. arXiv preprint arXiv:2401.08986 (2024)
4. Kipf, T.: and Welling. M. Semi-supervised classification with graphconvolutional networks. ICLR (2017)
5. Ying, C., et al.: Do Transformers really perform bad for graph representation? In: Advances in Neural Information Processing Systems, pp. 28877–28888 (2021)
6. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
7. Abu-El-Haija, S., et al.: Higher-order graph convolutional architectures via sparsified neighborhood mixing. ICML, MixHop (2019)
8. Zenil, H., Kiani, N.A., Tegnér, J.: Algorithmic Information Dynamics: A Computational Approach to Causality with Applications to Living Systems. Cambridge University Press (2023)
9. Zenil, H., Kiani, N.A., Tegnér, J.: A review of graph and network complexity from an algorithmic information perspective. Entropy **20**(8), 551 (2018)
10. Zenil, H., Hernández-Orozco, S., Kiani, N.A., Soler-Toscano, F., Rueda-Toicen, A., Tegnér, J.: A decomposition method for global evaluation of shannon entropy and local estimations of algorithmic complexity. Entropy **20**(8), 605 (2018)
11. Wu, Z., et al.: MoleculeNet: a benchmark for molecular machine learning. Chem. Sci. **9**, 513–530 (2018)
12. Gori, M., Monfardini, G., Scarselli, F.: A new model for learning in graph domains. In: Proceedings. IEEE International Joint Conference on Neural Networks, vol. 2, pp. 729–734 (2005)
13. Gilmer, J., Schoenholz, S., Riley, P., Vinyals, O., Dahl, G.: Neural message passing for quantum chemistry. arXiv preprint arXiv:1704.01212 (2017)
14. Yang, K., Swanson, K., J.W.: Analyzing learned molecular representations for property prediction. J. Chem. Inf. Model. **59**, 3370–3388 (2019)
15. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018)
16. Li, Q., Wu, X., Liu, H., Zhang, X., Guan, Z.: Label efficient semi-supervised learning via graph filtering. In: CVPR, pp. 9582–9591 (2019)
17. Jin, M., Chang, H., Zhu, W., Sojoudi, S.: Power up! robust graph convolutional network via graph powering. In: AAAI (2021)
18. Kong, L., Chen, Y., Zhang, M.: Geodesic graph neural network for efficient graph representation learning. arXiv preprint arXiv:2210.02636 (2022)
19. Sun, Y., et al.: Beyond homophily: structure-aware path aggregation graph neural network. In: Proceedings of the 31st International Joint Conference on Artificial Intelligence, pp. 2233–2240 (2022)

20. Michel, G., Nikolentzos, G., Lutzeyer, J., Vazirgiannis, M.: Path neural networks: expressive and accurate graph neural networks. In: Proceedings of the ICML (2023)
21. Zhu, Z., Zhang, Z., Xhonneux, L.P., Tang, J.: Neural bellman-ford networks: a general graph neural network framework for link prediction. In: Advances in Neural Information Processing Systems, vol. 34 (2021)
22. Bellman, R.: On a routing problem. Q. Appl. Math. **16**(1), 87–90 (1958)
23. Katz, L.: A new status index derived from sociometric analysis. Psychometrika **18**(1), 39–43 (1953)
24. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. J. Am. Soc. Inform. Sci. Technol. **58**(7), 1019–1031 (2007)
25. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web. Technical report, Stanford InfoLab (1999)
26. Morris, C., Kriege, N.M., Bause, F., Kersting, K., Mutzel, P., Neumann, M.: TUDataset: a collection of benchmark datasets for learning with graphs. In: Graph Representation Learning and Beyond Workshop (2020)
27. Cong, W., et al.: Do we really need complicated model architectures for temporal networks? In: ICLR (2023)
28. Xian, Y., Lampert, C., Schiele, B., Akata, Z.: Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. IEEE Trans. Pattern Anal. Mach. Intell. (2018)
29. Li, M., et al.: DGL-LifeSci: an open-source toolkit for deep learning on graphs in life science. ACS Omega (2021)
30. Wang, M., et al.: Deep graph library: a graph-centric, highly-performant package for graph neural networks. arXiv preprint arXiv:1909.01315 (2019)
31. Weininger, D.: Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. J. Chem. Inf. Comput. Sci. **28**(1), 31–36 (1988)
32. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: a next-generation hyperparameter optimization framework. In: KDD '19: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2623–2631 (2019)
33. Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J.E., Stoica, I.: Tune: a research platform for distributed model selection and training. arXiv preprint arXiv:1807.05118

A Systematic Performance Study of Parallel Programming Models for Stencil-Based CFD Applications on Heterogeneous Architectures

Mariana Padilha[1], Claudio Schepke[1(✉)], Tailí Petry[1], and José Rufino[2]

[1] Laboratory of Advanced Studies in Computation (LEA), Federal University of Pampa (UNIPAMPA), Alegrete, Brazil
{marianarp.aluno,claudioschepke,tailipetry.aluno}@unipampa.edu.br
[2] Research Centre in Digitalization and Intelligent Robotics (CeDRI), Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Instituto Politécnico de Bragança, 5300-253 Bragança, Portugal
rufino@ipb.pt
https://sites.unipampa.edu.br/lea/

Abstract. Coupled free-flow and porous media phenomena are widely stencil-based computational fluid dynamics (CFD) applications. Memory-access patterns and parallel programming abstractions strongly influence their performance. Understanding the impact of programming models on stencil computations in heterogeneous architectures, which combine multi-core CPUs and GPUs, is critical. This paper presents a systematic performance study of directive-based (OpenMP and OpenACC) and explicit-kernel (CUDA) parallel programming models applied to a structured-grid porous-media CFD solver. We perform a full refactoring of a legacy Fortran implementation and evaluate task granularity, scalability behavior, and architectural influence across distinct CPUGPU systems. Moreover, we analyze numerical reproducibility under IEEE 754 double precision and characterize the computational and memory-access behavior of the solver. Experimental results demonstrate how the abstraction level and parallel decomposition strategy impact scalability and performance limits in stencil-based CFD applications on heterogeneous architectures.

Keywords: High-Performance Computing · Computational Fluid Dynamics · Porous Media Problem · OpenMP · OpenACC · CUDA

1 Introduction

Devastating climate phenomena and environmental catastrophes have repeatedly occurred in different parts of the world [9], often involving complex interactions between free fluid flow and porous media, such as those observed in heavy rainfall events, dam failures, and mining waste overflows. The simulation of this type of

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 35–50, 2026.
https://doi.org/10.1007/978-3-032-30491-9_3

phenomenon involves the computation of operations based on the mathematical representation of the physical forces involved [2]. These phenomena are commonly modeled using the NavierStokes and DarcyForchheimer equations. Their numerical solution requires repeated stencil evaluations across space and time, making performance sensitive to memory-access patterns and parallel decomposition strategies.

Stencil-based computational fluid dynamics (CFD) solvers are widely used to simulate coupled free-flow and porous-media phenomena in environmental and engineering applications [7]. These solvers typically operate on structured grids and rely on iterative local-neighbor computations. As a result, they generate memory-intensive workloads and require strict convergence control. Due to these characteristics, stencil-based CFD applications are suitable candidates for evaluating parallel programming models on heterogeneous systems that combine multi-core CPUs and GPUs.

A previous study introduced a porous-media flow solver for an inclined plane configuration [10]. That study focused on algorithmic restructuring to improve cache efficiency and limit vectorization. Although modest gains were achieved in a single-core context, the increasing availability of heterogeneous systems demands a deeper investigation into how such legacy scientific codes can be redesigned and evaluated under distinct parallel programming abstractions.

Porting and parallelizing legacy CFD solvers involves more than inserting directives. It requires restructuring memory allocation strategies, adapting data access patterns, controlling task granularity, and ensuring numerical reproducibility across architectures. Directive-based models, such as OpenMP and OpenACC, and explicit-kernel programming approaches, such as CUDA, follow distinct abstraction philosophies, which may lead to different scalability behaviors and performance trends in stencil-dominated workloads.

In this work, we extend a porous-media CFD solver through a complete refactoring and parallel redesign targeting heterogeneous CPUGPU systems. The base legacy code is reorganized to support modular memory management, dynamic domain parameterization, and GPU-ready execution. Rather than focusing solely on raw speedup, we conduct a systematic performance study on how different parallel programming models could influence scalability, task granularity, and architectural behavior in a stencil-based CFD application.

The main contributions of this work are: (1) a structured refactoring of a legacy porous-media CFD solver into a modular and parameterized implementation suitable for heterogeneous architectures; (2) a systematic comparative evaluation of directive-based (OpenMP and OpenACC) and explicit-kernel (CUDA) programming models applied to a structured-grid stencil workload; (3) a detailed analysis of task granularity and scalability behavior across multi-core CPU and GPU; (4) a performance characterization of the computational and memory-access behavior of the solver, providing insight into observed scalability limits; (5) a numerical reproducibility validation under IEEE 754 double-precision arithmetic across CPU and GPU executions; (6) a controlled multi-architecture

experimental study highlighting the influence of hardware characteristics on parallel efficiency.

The remainder of this paper is organized as follows. Section 2 references some related works. Section 3 shows the application's functionalities and the code structure. The description of the methodology of the experiments is in Sect. 4. Section 5 shows the performance results for the different parallel implementation approaches. Finally, Sect. 6 presents the conclusion and future work.

2 Related Works

ALHafez and Kurdi [1] present a systematic analysis of MPI, OpenMP, and CUDA parallel programming interfaces across multiple dimensions: architectural foundations, performance characteristics, domain-specific suitability, programming complexity, and recent advancements. Their study restricts to a theoretical examination of each model's strengths, weaknesses, and optimization techniques.

Khalilov and Timoveev [6] compare the performance of CUDA, OpenMP, and OpenACC on an Nvidia Tesla V100 GPU in various typical scenarios that arise in scientific programming, such as matrix multiplication, regular patterns for memory access (BabelStream benchmark), and evaluate the performance of physical simulation codes implemented using these programming models (Cloverleaf and Lulesh benchmark). Moreover, they study the performance of matrix multiplication implemented in vendor-optimized BLAS libraries for the Nvidia Tesla V100 GPU and the Math Kernel Library (MKL) for Intel Xeon processors.

In [11], Silva et al. provide OpenMP and OpenACC implementations for a grain-drying simulation, a porous media application. They evaluate native `DO CONCURRENT`, OpenMP (loop, sections, tasks, target, and teams approach), and OpenACC parallel loops, maintaining the same code structure.

Karimi and Bhattacharya [5] focuses on an application that addresses a fluid flow and solute transport problem in periodic porous media. The authors select C++ for their implementation, adopting the CUDA interface and using the cuFFT and cuBLAS standard libraries for the GPU version of their code. Data exchange between the CPU and GPU occurs only at initialization and at the end of the execution to store the results for recovery. The rest of the calculation is conducted entirely on the GPU. Two FEM-based methods are solved on a single core of an Intel Skylake CPU (2.1 GHz), while the FFT-based method is solved on both the same CPU and a NVIDIA P100 GPU with 3584 cores (1.3 GHz). The performance of the GPU version is approximately 20× faster.

The approach pursued in our work, and described in this paper, considers an entire application, as proposed in [5,11], instead of conceptual notions [1] or benchmarks [6]. However, we apply the same parallel programming interfaces as in [6]: OpenMP, OpenACC, and CUDA. We also incorporate the notion of restricted data movement between the CPU and GPU only before and after the iterative step, as in [5], and diverge by explicitly implementing all functions called by our application, instead of making calls to external libraries.

3 Numerical Model Application

The original goal of the application we have refactored is to simulate and produce a numerical solution for a coupled flow of a free fluid over a porous medium. The model data are defined using discrete quantities through the Finite Volume Method applied to a heterogeneous system [4]. The application addresses the challenges associated with modeling abrupt transitions between free flow, governed by the Navier–Stokes equations [8], and porous media, described by the Darcy–Forchheimer equation [3]. In particular, the influence of porous matrix properties on flow behavior is investigated, contributing to a better understanding and numerical modeling of such complex systems.

The application source code is organized into three main stages: i) preprocessing, ii) main computation, and iii) data output, as illustrated in Fig. 1. Initially, two data structures are read from input files. These structures contain the physical parameters associated with a given simulation and the values defining the domain decomposition sizes. Subsequently, memory allocation is performed, and the finite volume discretization scheme is applied.

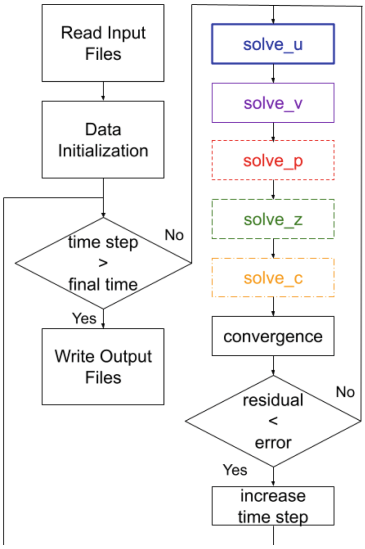


Fig. 1. Main flowchart of the numerical model. Adapted from [10].

The main computation advances over discrete time steps and iterates over a two-dimensional spatial domain. At each time step, the physical variables are updated iteratively until a convergence criterion is satisfied. Within this inner iteration loop, the first stage solves the momentum equations in the x and y directions. Next, the continuity equation is enforced by solving a Poisson equation associated with mass conservation. The third stage computes energy transport (temperature), followed by the calculation of concentration through the

solution of a scalar advection–diffusion equation. At the end of each inner iteration, selected variables are updated and the convergence metric is evaluated. The iterative process terminates either when the convergence criterion is reached or when the maximum number of iterations is exceeded. After each time-step increment in the outer loop, additional variables are updated accordingly.

At the end of the execution, all computed physical properties are written to separate output files, one for each variable. These outputs can subsequently be post-processed and visualized using external tools, such as Python scripts.

Figure 2 illustrates the subroutines invoked during the iterative solution process. For all solver routines—`solve_u`, `solve_v`, `solve_p`, `solve_z`, and `solve_c`—there are associated subroutine calls responsible for data computation and boundary condition treatment. For instance, in the pressure solver `solve_p` (highlighted in red), the `resp` subroutine computes intermediate data, while `bcp` applies the corresponding boundary conditions. The same structure is adopted for the remaining solver routines.

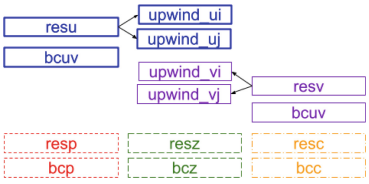


Fig. 2. Subroutines invoked by each solver during the iterative step. Box border types and font colors correspond to the operations shown in Fig. 1.

The routines responsible for data computation and boundary condition evaluation are invoked three times within each solver routine. This behavior results from the adoption of a second-order Runge–Kutta time integration scheme. In this approach, two intermediate estimates are required to compute the final approximation, which is obtained as a weighted average of the evaluated slopes. Additionally, the routines `upwind_ui`, `upwind_uj`, `upwind_vi`, and `upwind_vj` are employed to handle the simplified and distinct computation of grid points located near the domain boundaries, specifically in the second and penultimate rows and columns of the discrete mesh.

The convergence metric is computed from the residual vectors produced by `solve_u`, `solve_v`, and `solve_p`, using the `maxval` operator (denoted as max, below), which returns the maximum absolute value within each residual array:

$$residual_u = \max(|res_u|), \tag{1}$$

$$residual_v = \max(|res_v|), \tag{2}$$

$$residual_p = \max(|res_p|), \tag{3}$$

$$error = \max(residual_u, residual_v, residual_p). \tag{4}$$

The various implementations developed, based on the OpenMP, OpenACC, CUDA, and `DO CONCURRENT` parallel programming interfaces, are public[1].

The code organization is mainly governed by temporal discretizations. Spatial discretizations are controlled externally by the convergence loop, while internally, the operations are divided into five functions based on the physical calculations involved. Each function is organized into three loops. Each loop operates on one of the steps of the second-order Runge-Kutta method over all points.

The computation is distinct for the different types of discrete points. The vast majority of the elements follow a stencil processing pattern, including neighboring point values. For instance,

$$P_{ij} = V_{i-1,j} + V_{i+1,j} + V_{i,j-1} + V_{i,j+1}, \tag{5}$$

where P is a physical property and V abstracts distinct computations involving the processing of neighboring element values. The elements at the edges have simpler calculations according to the definition of boundary conditions:

$$\begin{array}{llll} u(x,y) = 0, & v(x,y) = 0 & \text{within the porous medium} \\ u(x,y) = 0, & v(x,y) = v_i & \text{in the free flow region,} \end{array} \tag{6}$$

where v_i is the injection speed.

To avoid a conditional test per computed point, points are grouped according to the type of processing applied to each group. That is, the internal points are computed in a block of rows and columns. The external points from the first and last rows and columns are computed separately, each in a specific loop.

4 Evaluation Methodology

In this section, we describe the computer architecture, software, domain discretization, and case studies targeted in this work.

4.1 Testbed

The tests were performed on a bare-metal Linux runtime, using three systems (nodes) of different architectures, with distinct numbers of cores, CPU frequencies, cache sizes, and different co-processors (GPUs) attached:

- *Node*1: 2 × Intel Xeon CPU E5-2650 octa-core processors, 2 threads per core, and 20 MB of L3, 256 KB (per core) L2, and 32 KB (per core) L1 caches; 1 × NVIDIA Quadro M5000 GPU with 2048 CUDA cores, 8 GB of global memory, and caches of 2 MB for L2 and 48 KB (per SM) for L1;
- *Node*2: 2 × Intel Xeon CPU E5-2420 hexa-core processors, 2 threads per core, and 15 MB of L3, 256 KB (per core) L2, and 64 KB (per core) L1 cache; 2 × NVIDIA TITAN V GPUs with 5120 CUDA cores, 12 GB of global memory, and caches of 4.5 MB for L2 and 96 KB (per SM) for L1;

[1] https://anonymous.4open.science/r/simulacao-de-meios-porosos-636E.

– *Node*3: 1 × Intel i9-13900HX CPU with 8 performance-cores (clock frequency limited to 4 GHz, 2 threads per core), 16 efficient-cores, and 36 MB of L3, 2 MB (per core) L2, 80 KB (per core) L1, 4 MB (per module) E-Core L2, and 96 KB (per core) E-Core L1 cache; 1 × NVIDIA GeForce RTX 4060 GPU with 3072 CUDA cores, 8 GB of global memory, and caches of 24 MB for L2 and 128 KB (per SM) for L1.

The compilers used were `nvfortran` and `nvcc` from NVIDIA HPC_SDK version 25.5, with the compilation flags `-O3` to allow aggressive compiler optimizations for all environments and input sizes, `-mp` when running in parallel using OpenMP threads, and `-stdpar=gpu -acc -Minfo=accel` for OpenACC.

4.2 Domain and Case Studies

Figure 3 illustrates the problem domain. The simulation considers a surface of 12 × 20 cm. The open fluid flow extends 15 cm in the positive y-axis direction, and the porous media extends 5 cm in the negative y-axis direction. Special operations occur in a transition zone between fluid flow and porous media, such as soil, for example. For the boundary conditions, there are also special operations.

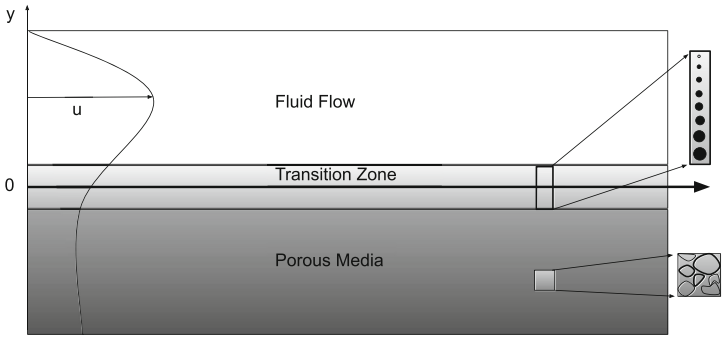


Fig. 3. Domain size scheme for a y-axis direction open fluid flow.

We defined 4 case studies. Each one is four times larger than the previous one. The number of points for each case after the discretization by the finite volume of the domain illustrated in Fig. 3 is: $Case1$ - $51 \times 84 = 4,284$; $Case2$ - $102 \times 169 = 17,238$; $Case3$ - $153 \times 254 = 38,862$; $Case4$ - $204 \times 339 = 69,156$.

The simulation covered the time period from 0 to 1 s using a Δt of 0.005 s, which is equivalent to 200 steps of the external loop. The convergence criteria requires less than 10^{-4} for the residual vectors of `solve_U`, `solve_V`, and `solve_P`. If the stopping criteria are not reached in 800,000 iterations for a specific time step, the algorithm ends, indicating convergence problems.

Each experiment was executed 10 times to collect the mean execution time for initialization, iterative steps, and post-processing. The standard deviation

observed was less than 1.5% of the average execution time. The pre- and post-processing times are low in relation to the total execution time. Table 1 presents the initialization and post iterative step times for $Node1$ for all four case studies.

Table 1. Pre- and Post-Processing execution time in seconds for $Node1$

Time	Case1	Case2	Case3	Case4
Pre-processing	0.03	0.07	0.16	0.32
Post-processing	0.06	0.17	0.48	1.62

The GPU memory usage observed, given by the `nvidia-smi` command for the four case studies, in order of domain size, was 312, 322, 340, and 368 MiB.

We also ensured strict numerical equivalence: all codes produce identical output results. This comparison and verification was done using Python scripts.

5 Evaluation Results

This section presents the experimental results obtained from executing the refactored CFD solver on three distinct heterogeneous architectures. We analyze the convergence behavior of the numerical method, evaluate the performance of multicore parallel implementations (OpenMP, OpenACC multicore, and `DO CONCURRENT`), and compare them against GPU-accelerated versions using OpenACC and CUDA. The analysis focuses on scalability, the impact of domain size on performance gains, and the relative efficiency of each programming model across different hardware configurations.

5.1 Numerical Convergence

Figure 4 presents the number of iterations necessary to attain the values for convergence at each time step (1 to 200) in $Case1$. Each time step converges when all partial residual values for `solve_u`, `solve_v`, and `solve_p` methods are smaller than 10^{-4} (stop criterion). The number of iterations decreases as the time step advances. That occurs because a time step computation takes the results from the previous step. After the 80th time step, the number of iterations reduces to 1 in most steps. A similar behavior appears in the other case studies.

Figure 5 displays the values achieved for the residual of the quantity of movement in x (res_u) and y (res_v), as well as of the continuity (res_p), also for $Case1$. The residual res_p is the last to achieve a value smaller than 10^{-4} in most cases, while res_u and res_v have already reached the 10^{-8} to 10^{-9} range.

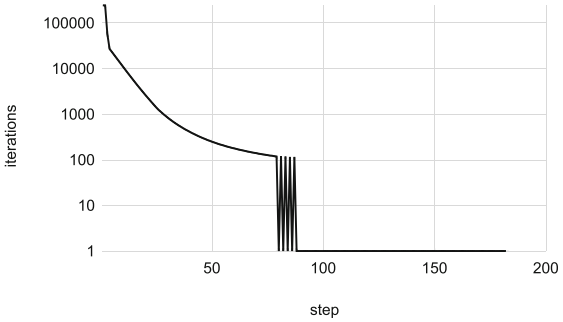


Fig. 4. Iterations needed for the convergence (log. scale, $Case1$).

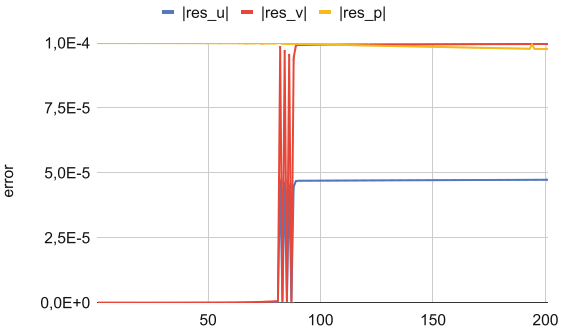


Fig. 5. Values of the residual error at the moment of convergence ($Case1$).

5.2 Multi-core Performance

Figure 6 shows the speedup for the $Case1$ in $Node2$, evaluating multi-core parallel programming interfaces against a sequential code version. We range the number of threads from 2 to 12, in steps of 2. Note that for each parallel programming interface, it is explicitly necessary to set the exact number of threads.

The results are relatively similar for OpenMP and OpenACC. The OpenACC has the worst results, perhaps limited by the multithreading generated by the compiler. The $Case1$ is the smallest test, and because of this, the performance gain is limited. For example, we tested `DO CONCURRENT`, and the speedup was 3.25, which is the maximum speedup of OpenACC.

Figure 7 presents the speedup for the three case studies. As before, we range the number of OpenMP threads from 2 to 12, in steps of 2, now comparing that version with the `DO CONCURRENT` Fortran version for multi-core processing and the OpenACC version for GPU computation. As the case study sizes increase, the speedup improves due to the larger granularity.

The relationship between the performances of the `DO CONCURRENT` and OpenMP variants improves with larger domains. For example, the `DO CONCURRENT` variant in $Case1$ has a speedup equivalent to that of 4 OpenMP

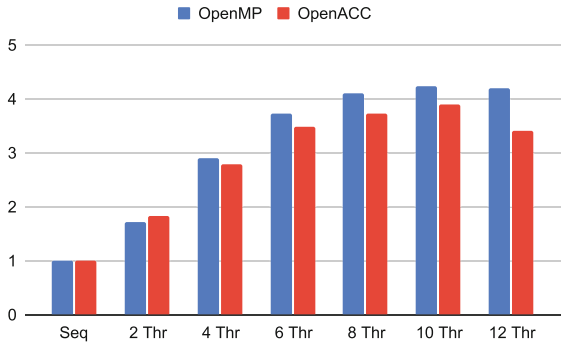


Fig. 6. Speedup of Multicore OpenMP and OpenACC variants ($Case1$ in $Node2$).

threads, while in $Case2$ there is a speedup between 6 and 8 OpenMP threads, and the speedup exceeds 8 OpenMP threads for $Case3$.

The OpenACC variant running on the GPU on $Node1$ achieved a speedup of 9.27 to the sequential single-core implementation, for $Case3$. This result is similar to using OpenMP with 12 threads.

To gain additional insight into the GPU-based OpenACC implementation, we also ran with $Case4$. In this scenario, the sequential version takes $314,241.68s$, while the OpenACC variant demands just $12,807.16s$, yielding a speedup of $24.53\times$, improving over the value ($20\times$) presented in [5].

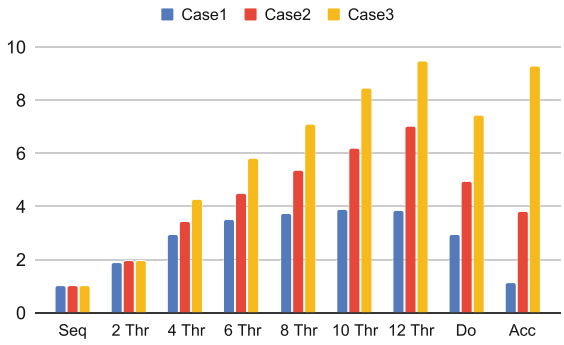


Fig. 7. Speedup of the Multicore OpenMP and `DO CONCURRENT` variants, and of the GPU-based OpenACC variant ($Case1$, $Case2$, and $Case3$ in $Node1$).

5.3 GPU Performance

We also provide a detailed analysis of the GPU-accelerated implementations using OpenACC and CUDA. We evaluate their performance on the $Node3$

architecture, compare the efficiency of directive-based versus explicit kernel programming, and employ the Roofline model to identify performance bottlenecks. The analysis focuses on the results obtained with the RTX 4060 Laptop GPU, which represents a modern and energy-efficient architecture where the trade-offs between programming effort and performance are particularly relevant.

CUDA Implementation and Profiling. The main CUDA implementation choices are summed up in Table 2, and profiling results are provided in Table 3.

We use 2D thread blocks of 16×16 threads to map the structured grid onto the GPU execution model. This configuration provides a good balance between thread occupancy and memory access efficiency for stencil-based computations.

Table 2. CUDA kernel configuration used in the GPU implementation.

Parameter	Value
Thread block size	16×16
Thread mapping	2D structured grid (1 thread per cell)
Kernel organization	27 kernels (one per solver stage)
CUDA streams	Used for boundary condition kernels
Memory access pattern	Strided global memory accesses in 2D arrays

A dedicated CUDA kernel receives each major computation state of the solver. This design enables fine-grained control over GPU execution and different solver stages to be scheduled independently on the device.

We use CUDA streams to enable the concurrent execution of independent kernels whenever data dependencies permit it. However, several solver stages require sequential execution due to dependencies between intermediate fields. As a result, full kernel concurrency is not always possible, and streams are applied only to operations to achieve safe overlap.

The mapping between grid points and threads preserves spatial locality. However, due to the layout of the 2D arrays in memory, the fastest varying thread dimension does not perfectly align with the contiguous memory axis, resulting in partially strided global memory accesses.

The profiling results (see Table 3) revealed that the majority of the runtime is concentrated in the velocity solver kernels. Together, the `solve_u` and `solve_v` kernels account for approximately 70% of the total kernel execution time, reflecting the higher computational complexity of the momentum equations. The scalar transport kernels (`solve_z` and `solve_c`) represent about 26% of the runtime, while the pressure solver (`solve_P`) contributes only 4%. Additionally, only 1.9% of the total execution time corresponds to host-side overhead, indicating that the solver computation is effectively offloaded to the GPU (Fig. 8).

Table 3. Runtime distribution among the main CUDA kernel.

Kernel	Total Time (s)	Percentage
solve_u	2200.7	34.77%
solve_v	2249.3	35.53%
solve_p	255.0	4.03%
solve_z	793.0	12.53%
solve_c	832.2	13.15%
Total kernel time	6330.3	100%
Total execution time	6453.0	—
Overhead	122.8	1.90%

CPU-Based vs. GPU-Based Variants on Node3. Figure 9 presents the speedup achieved by the different parallel implementations (CPU- and GPU-based) on $Node3$ for $Case1$, $Case2$, and $Case3$, all relative to the original sequential CPU version. Several important observations emerge from these results.

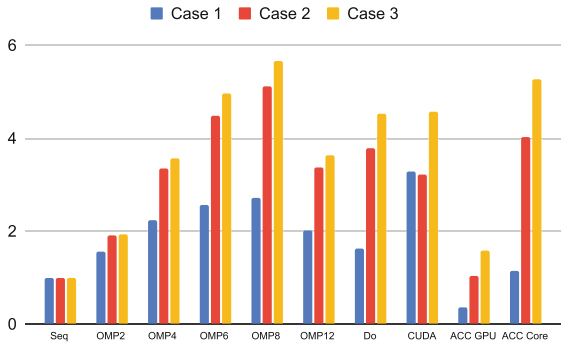


Fig. 8. Speedup of the OpenMP, `DO CONCURRENT` and OpenACC Multicore variants, and of the CUDA and OpenACC GPU variants ($Case1$, $Case2$, and $Case3$ in $Node3$).

First, the absolute speedup values on Node3 are lower than those observed on Node1 and Node2. That is primarily due to the high single-core performance of the Intel $i9-13900HX$ processor. Because the sequential baseline executes faster on this modern CPU, the speedup achieved through parallelization - whether on multiple CPU cores or on the GPU - is naturally reduced.

For the multi-core implementations, the results follow the expected behavior. The OpenMP version shows increasing speedup as the number of threads grows, scaling up to 8 threads. Beyond this point, performance gains begin to plateau due to the limitations of granularity. Even for Case 3, the problem size does not provide sufficient workload to utilize additional threads efficiently. Both the `DO CONCURRENT` and multi-core OpenACC implementations also deliver good

speedup when using the maximum number of threads, achieving performance comparable to OpenMP.

Regarding GPU-accelerated execution, two key observations stand out. First, the CUDA version, evaluated here for the first time, achieves speedup results similar to those obtained with the best OpenMP multi-core configuration. For *Case*3, the maximum OpenMP speedup (using 12 threads) is approximately 9.0×, while CUDA achieves 9.1×. That indicates that, for these domain sizes and on this specific architecture, the GPU does not significantly outperform a fully utilized multi-core CPU. The CPU's high single-thread performance, combined with the GPU's memory bandwidth limitations (as further discussed in the Roofline analysis), contributes to this parity.

Second, the OpenACC GPU implementation produces significantly smaller gains compared to the CUDA version. While CUDA achieves a 9.1× speedup for *Case*3, OpenACC on the GPU reaches only 5.3×. This performance gap of approximately 1.7× across all domain sizes is substantial and warrants further investigation. The consistency of this gap—it does not decrease significantly with larger workloads—suggests that the difference is not merely due to runtime overhead, but rather reflects how effectively each programming model utilizes the GPU's computational and memory resources. The Roofline analysis in the next section provides further insight into this behavior.

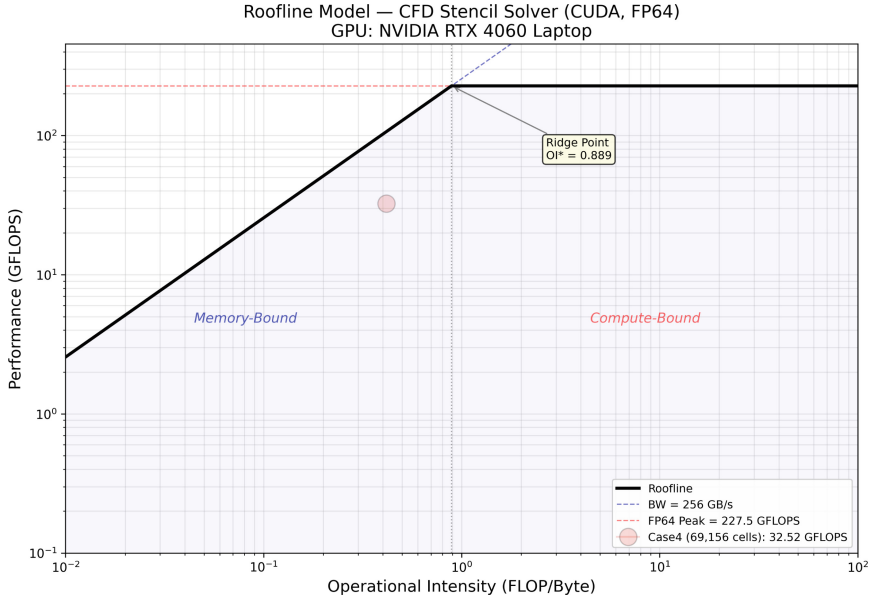


Fig. 9. Roofline model for the CUDA implementation of the CFD solver on the RTX 4060 Laptop GPU (Node3). The solver operates in the memory-bound region with an operational intensity of approximately 0.42 FLOP/Byte.

Roofline Model Analysis on Node3. To quantitatively understand the performance characteristics of the GPU implementation, the Roofline performance model is employed for the largest problem size ($Case4$) on the $Node3$ architecture. Figure 9 presents the Roofline plot for the CFD solver, allowing us to determine whether the application is limited by memory bandwidth or computational throughput.

The RTX 4060 GPU in $Node3$ has a theoretical peak double-precision performance of approximately 227.5 GFlop/s and a peak memory bandwidth of 256 GB/s. This results in a ridge point—where the application transitions from memory-bound to compute-bound—at an arithmetic intensity of approximately 0.89 Flop/Byte. The operational intensity of the solver was estimated by counting the floating-point operations and memory transfers performed in the main stencil kernels. Considering all solver stages, including the QUICK discretization, central-difference computations, and RungeKutta time integration, the overall operational intensity is approximately 0.42 Flop/Byte.

Since this value lies well below the ridge point, the application is clearly memory-bound. This behavior is characteristic of stencil-based CFD solvers, where each grid point update requires loading multiple neighboring values from memory while performing a relatively small number of arithmetic operations.

Based on the Roofline model, the maximum attainable performance for this operational intensity is approximately 106.8 GFlop/s, determined by the memory bandwidth limit (256 GB/s $\times$ 0.42 Flop/Byte). The measured performance of the CUDA implementation is approximately 32.5 GFlop/s, corresponding to about 30% of the memory-bandwidth-limited Roofline performance. This result confirms that the solver is primarily constrained by memory traffic rather than computational throughput. Consequently, further performance improvements are unlikely to be achieved through floating-point optimizations alone.

From an optimization perspective, since the application is memory-bound, further improvements should focus on increasing arithmetic intensity and improving memory access efficiency. Potential strategies include (1) increasing data reuse through techniques such as loop tiling or temporal blocking, (2) ensuring fully coalesced memory accesses in GPU kernels, and (3) leveraging the large L2 cache available on the RTX 4060 to reduce DRAM traffic. Given the intrinsic characteristics of stencil-based CFD workloads, the current performance already reflects the fundamental memory bandwidth limitations of the architecture.

6 Conclusion and Future Work

This paper improves an application for simulating free flow parallel to a porous plane. This application can solve problems such as soil's rain absorption and flood simulation. We described the code organization and proposed changes to parallelize it. An analysis of convergence for the timestep iterations and the relationship between the residual errors, was also presented.

We discussed several implementations using the parallel programming interfaces OpenMP, OpenACC, CUDA, and Fortran `DO CONCURRENT` directive. The

results show performance gains of 9.43× using 12 OpenMP threads for $Case3$, on $Node1$. The multi-core OpenACC version presents results alongside OpenMP. The `DO CONCURRENT` version shows that the results are more distant from OpenMP.

For GPUs, CUDA and OpenACC are good alternatives. CUDA can explicitly specify parallel operations, whereas OpenACC depends on directive injection and compiler decisions. The speedup achieved for $Case4$ with OpenACC was 24.

More running tests are needed to establish other performance relationships. For example, it is relevant to evaluate larger case studies, especially for GPU executions or faster cores. We intend to implement an OpenMP target for GPU in future work to establish a third GPU approach. The target approach allows the selection of the parallel environment at compilation time.

Acknowledgments. This work was supported by FCT - Fundação para a Ciência e Tecnologia, I.P. by projects: CeDRI, UID/05757/2025 (DOI: 10.54499/UID/05757/2025) and UID/PRR/05757/2025 (DOI: 10.54499/UID/PRR/ 05757/2025); SusTEC, LA/P/0007/2020 (DOI: 10.54499/LA/P/0007/2020); by CNPq: 10/2023 Universal (Project N° 407827/2023-4); and by FAPERGS: PROBIC scientific initiation scholarship.

References

1. ALHafez, N., Kurdi, A.: Parallel paradigms in modern HPC: a comparative analysis of MPI, OpenMP, and CUDA. arXiv preprint arXiv:2506.15454 (2025)
2. Civan, F.: Porous Media Transport Phenomena. Wiley, Hoboken, NJ, United States (2011)
3. Cornelissen, P.: Coupled free-flow and porous media flow: a numerical and experimental investigation. Ph. D. thesis, Faculty of Geosciences - Utrecht University (2016)
4. Ferziger, J.H., Perić, M., Street, R.L.: Finite Volume Methods, pp. 81–110. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-319-99693-6_4
5. Karimi, M., Bhattacharya, K.: Accelerated computational micromechanics for solute transport in porous media. Comput. Methods Appl. Mech. Eng. **426**, 116976 (2024)
6. Khalilov, M., Timoveev, A.: Performance analysis of CUDA, OpenACC and OpenMP programming models on TESLA V100 GPU. J. Phys. Conf. Ser. **1740**(1), 012056 (2021)
7. Krishnamoorthy, S., Baskaran, M., Bondhugula, U., Ramanujam, J., Rountev, A., Sadayappan, P.: Effective automatic parallelization of stencil computations. In: Proceedings of the 28th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI '07, pp. 235–244. Association for Computing Machinery, New York, NY, USA (2007)
8. Kundu, P., Cohen, I., Hu, H.: Fluid Mechanics. Academic Press (2002)
9. Liritzis, I.: Disasters and climatic phenomena today and in the past. Proc. Eur. Acad. Sci. Arts **2**(1) (2023). https://doi.org/10.4081/peasa.22, https://www.peasa.eu/site/article/view/22

10. Schepke, C., Spigolon, R.A., Rufino, J., Cristaldo, C.F.D.C., Pizzolato, G.L.: Contributions to accelerating a numerical simulation of free flow parallel to a porous plane. In: 2025 33rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), pp. 226–233 (2025). https://doi.org/10.1109/PDP66500.2025.00038
11. Silva, H.U.D., Lucca, N., Schepke, C., Oliveira, D.P.D., Cristaldo, C.F.D.C.: Parallel OpenMP and OpenACC porous media simulation. J. Supercomput. **79**(8), 8425–8446 (2023).https://doi.org/10.1007/s11227-022-05004-2

Uncertainty Quantification of Underwater Sound Propagation Using Surrogate Models

Mikhail S. Lytaev(✉)

St. Petersburg Federal Research Center of the Russian Academy of Sciences, 14-th Linia, V.I., No. 39, Saint Petersburg 199178, Russia
mikelytaev@gmail.com

Abstract. Accurate prediction of underwater sound propagation is essential for sonar performance assessment, marine ecology studies, and ocean acoustic tomography. However, the inherent uncertainty in environmental parameters, particularly the sound speed profile (SSP), introduces significant variability in transmission loss predictions. In this work, we propose a Kriging-based surrogate modeling framework for uncertainty quantification (UQ) of underwater acoustic propagation. The sound speed variability is parameterized using empirical orthogonal functions (EOFs) derived from Argo float data collected in the Baltic Sea. A wide-angle parabolic equation solver, implemented within the PyWaveProp library, serves as the high-fidelity model, and Kriging surrogate models from the SMT library are employed to construct a computationally efficient approximation that maps uncertain EOF coefficients to transmission loss fields. Latin hypercube sampling is used for experimental design, and the surrogate accuracy is validated against direct Monte Carlo simulations. Numerical experiments demonstrate that the mean transmission loss field under SSP uncertainty can differ significantly from the deterministic prediction, while the qualitative structure of the field remains largely preserved even under large perturbation amplitudes.

Keywords: Underwater acoustics · Uncertainty quantification · Kriging · Gaussian process regression · Surrogate modeling · Parabolic equation method

1 Introduction

Underwater sound propagation modeling plays a central role in a wide range of applications, including sonar performance prediction, underwater communication, marine mammal impact assessment, and ocean acoustic tomography [13]. The acoustic field in the ocean is governed by the wave equation, which depends on environmental parameters such as the sound speed profile (SSP), water depth (bathymetry), and geoacoustic properties of the seabed. In practice, these parameters are known only approximately, introducing uncertainty into the predicted acoustic field.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 51–66, 2026.
https://doi.org/10.1007/978-3-032-30491-9_4

Uncertainty quantification (UQ) aims to characterize the impact of input uncertainties on model outputs. The reference approach is Monte Carlo (MC) simulation: draw N independent realizations of the uncertain inputs, run the acoustic solver for each, and estimate output statistics from the ensemble. While MC converges at a rate $\mathcal{O}(N^{-1/2})$ regardless of input dimension, it requires $N = 10^3$–10^5 solver evaluations to converge–prohibitive when a single parabolic-equation (PE) solve takes seconds to minutes. This has motivated the development of surrogate-based UQ methods, which replace the expensive simulator with a fast-to-evaluate approximation [24].

Several surrogate modeling strategies have been explored for underwater acoustic UQ. Polynomial chaos expansions (PCE) represent the model output as a series of orthogonal polynomials in the uncertain inputs, enabling analytical computation of statistical moments and sensitivity indices [3,22,25]. Finette [9,17] pioneered the application of Karhunen–Loève decomposition combined with polynomial chaos to the stochastic acoustic waveguide problem, demonstrating rapid convergence for the mean field and intensity. Khazaie et al. [16] compared PCE, Kriging, and hybrid PC-Kriging surrogates for shallow-water acoustic UQ, finding that all three methods drastically reduce the required sample size compared to brute-force MC.

Dynamically orthogonal (DO) methods take an intrusive approach, reformulating the stochastic differential equation in a low-rank representation [1,2]. While achieving orders-of-magnitude speedup over MC, DO methods require modification of the acoustic solver code, limiting their applicability to legacy or black-box models. Moreover, existing DO formulations for underwater acoustics have been developed only for the narrow-angle PE, and their extension to wide-angle approximations remains an open problem.

Deep learning surrogates have also been applied to underwater acoustics. Du et al. [8] developed a physics-informed neural network (PINN) for acoustic field prediction by embedding the Helmholtz equation as a loss constraint. Pham and Sarkar [19] trained a neural network to estimate transmission loss uncertainty from a single baseline calculation. Indu Kant Deo et al. [15] combined neural encoders with a sparse variational Gaussian process output layer for three-dimensional TL prediction with calibrated uncertainty bands.

Kriging, also known as Gaussian process (GP) regression, occupies a favorable position among these approaches: it is non-intrusive, provides both a prediction and an associated predictive variance, and supports adaptive sampling strategies [23]. Despite these advantages, Kriging remains underexplored for underwater acoustic propagation problems, particularly in the context of SSP uncertainty parameterized by empirical orthogonal functions (EOFs).

In this paper, we present a Kriging-based surrogate modeling framework for UQ of underwater sound propagation. The high-fidelity model is a wide-angle PE solver implemented in the PyWaveProp library [20]. We propose a parameterization of the uncertain sound speed profile using EOF coefficients derived from Argo float data [14] in the Baltic Sea, which provides a compact, data-driven input space for the surrogate model. We systematically compare

four Kriging variants from the SMT (Surrogate Modeling Toolbox) library [6] and validate the surrogate predictions against direct adaptive MC simulations.

The remainder of the paper is organized as follows. Section 2 reviews the relevant background on underwater acoustic modeling and Kriging. Section 3 describes the proposed methodology. Section 4 presents numerical experiments and results. Section 5 concludes the paper.

2 Background

2.1 Underwater Acoustic Propagation

The acoustic pressure field $p(\mathbf{r}, t)$ in the ocean satisfies the Helmholtz equation in the frequency domain:

$$\nabla^2 p(\mathbf{r}) + k^2(\mathbf{r})\, p(\mathbf{r}) = -\delta(\mathbf{r} - \mathbf{r}_s),$$

where $k(\mathbf{r}) = \omega / c(\mathbf{r})$ is the wavenumber, ω is the angular frequency, $c(\mathbf{r})$ is the spatially varying sound speed, and $\mathbf{r}_s$ denotes the source position.

The parabolic equation (PE) method provides an efficient numerical solution for range-dependent environments by factoring the Helmholtz equation into a one-way marching equation [7]:

$$\frac{\partial \psi}{\partial r} = \mathrm{i} k_0 \left(\sqrt{1 + X} - 1 \right) \psi,$$

where ψ is the reduced pressure field, r is the range, k_0 is a reference wavenumber, and X is an operator involving depth derivatives and the index of refraction. Transmission loss (TL) is the standard metric for characterizing acoustic propagation:

$$\mathrm{TL}(r, z) = -20 \log_{10} \left| \frac{p(r, z)}{p_0} \right|,$$

where p_0 is a reference pressure.

2.2 Kriging (Gaussian Process Regression)

Kriging models the response $y = f(\mathbf{x})$ as a realization of a Gaussian process:

$$Y(\mathbf{x}) = \boldsymbol{\beta}^T \mathbf{f}(\mathbf{x}) + Z(\mathbf{x}),$$

where $\boldsymbol{\beta}^T \mathbf{f}(\mathbf{x})$ is a trend (regression) term and $Z(\mathbf{x})$ is a zero-mean GP with covariance

$$\mathrm{Cov}[Z(\mathbf{x}), Z(\mathbf{x}')] = \sigma^2\, R(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}),$$

defined for any two points $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, where σ^2 is the process variance, R is a correlation function (e.g., Matérn or squared exponential), and $\boldsymbol{\theta}$ are hyperparameters estimated by maximum likelihood.

Given a training set $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N$, let $\mathbf{R} \in \mathbb{R}^{N \times N}$ denote the correlation matrix with entries $R_{ij} = R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}; \boldsymbol{\theta})$, let $\mathbf{F} \in \mathbb{R}^{N \times p}$ be the regression matrix

with rows $\mathbf{f}(\mathbf{x}^{(i)})^T$, and let $\mathbf{r}(\mathbf{x}^*) = [R(\mathbf{x}^{(1)}, \mathbf{x}^*; \boldsymbol{\theta}), \ldots, R(\mathbf{x}^{(N)}, \mathbf{x}^*; \boldsymbol{\theta})]^T$ be the vector of cross-correlations between the training points and a new point $\mathbf{x}^*$. The Kriging predictor is then:

$$\hat{y}(\mathbf{x}^*) = \mathbf{f}(\mathbf{x}^*)^T \hat{\boldsymbol{\beta}} + \mathbf{r}(\mathbf{x}^*)^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\beta}}),$$

with an associated prediction variance $\hat{\sigma}^2(\mathbf{x}^*)$ that quantifies surrogate uncertainty.

The hyperparameters $\boldsymbol{\theta}$ and the process variance σ^2 are estimated by maximizing the concentrated log-likelihood function [23]:

$$\ln L(\boldsymbol{\theta}) = -\frac{N}{2} \ln(2\pi) - \frac{N}{2} \ln(\hat{\sigma}^2) - \frac{1}{2} \ln |\mathbf{R}|,$$

where $\hat{\sigma}^2 = \frac{1}{N} (\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\beta}})^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\beta}})$ is the maximum likelihood estimate of the process variance and $|\mathbf{R}|$ is the determinant of the correlation matrix. Equivalently, the hyperparameters are found by minimizing the negative log-likelihood $-\ln L(\boldsymbol{\theta})$. This optimization problem is typically non-convex and is solved using gradient-based methods with multiple restarts.

In this work, we compare four Kriging variants from the SMT library [6], which differ primarily in how they handle the correlation matrix and the hyperparameter optimization:

- **KRG** (full Kriging) estimates a separate length-scale parameter θ_j for each input dimension $j = 1, \ldots, d$, so that the correlation function takes the product form $R(\mathbf{x}_i, \mathbf{x}_k; \boldsymbol{\theta}) = \prod_{j=1}^{d} r(|x_{ij} - x_{kj}|; \theta_j)$, where $r(\cdot\,; \theta_j)$ is a one-dimensional kernel (e.g., squared exponential or Matérn). The number of hyperparameters scales linearly with d, making the likelihood optimization increasingly expensive in high dimensions.
- **KPLS** [5] (Kriging with partial least squares) addresses the curse of dimensionality by first applying PLS regression to identify a small number $h \ll d$ of latent components $\mathbf{t}_l = \mathbf{X}\mathbf{w}_l$ $(l = 1, \ldots, h)$ that maximize the covariance between inputs and outputs. The correlation function is then parameterized in the reduced space, requiring only h hyperparameters instead of d. This dramatically reduces the cost of likelihood optimization while preserving accuracy when the response varies primarily along a low-dimensional subspace.
- **KPLSK** [4] is a two-stage hybrid: it first fits a KPLS model to obtain an inexpensive initial estimate of the hyperparameters, then uses these as starting values for a full Kriging optimization in the original d-dimensional space. This combines the fast convergence of KPLS with the flexibility of KRG, often achieving accuracy close to full Kriging at a fraction of the cost.
- **GPX** is a Kriging implementation based on the `egobox` library [18], a Gaussian process toolbox written in Rust. It uses the same mathematical formulation as KPLS but benefits from optimized linear algebra routines, yielding significantly faster training and prediction times.

3 Methodology

3.1 Problem Formulation

We consider a set of uncertain environmental parameters $\mathbf{x} = (x_1, \ldots, x_d) \in \mathcal{X} \subset \mathbb{R}^d$ with a known joint probability distribution $\pi(\mathbf{x})$. The quantity of interest (QoI) is the transmission loss field $\mathrm{TL}(r, z; \mathbf{x})$ computed by the PE solver at specified receiver locations.

The goal of UQ is to estimate statistical moments and probability distributions of the QoI:

$$\mathbb{E}[\mathrm{TL}] = \int_{\mathcal{X}} \mathrm{TL}(\mathbf{x})\, \pi(\mathbf{x})\, d\mathbf{x}, \quad \mathrm{SD}[\mathrm{TL}] = \sqrt{\mathbb{E}[\mathrm{TL}^2] - (\mathbb{E}[\mathrm{TL}])^2}.$$

3.2 EOF Parameterization of Sound Speed Uncertainty

Empirical orthogonal function (EOF) analysis [11,21], also known as principal component analysis (PCA), provides a data-driven decomposition of a spatially distributed random field into orthogonal modes ordered by explained variance. It has been widely used in physical oceanography to characterize the variability of temperature, salinity, and sound speed profiles.

Let $\{c_k(z)\}_{k=1}^K$ denote a set of K observed sound speed profiles, each sampled (or interpolated) onto a common depth grid $\{z_1, \ldots, z_J\}$. Define the ensemble mean profile

$$\bar{c}(z_j) = \frac{1}{K} \sum_{k=1}^{K} c_k(z_j),$$

and the anomaly matrix $\mathbf{A} \in \mathbb{R}^{K \times J}$ with entries $A_{kj} = c_k(z_j) - \bar{c}(z_j)$. The spatial covariance matrix is

$$\mathbf{C} = \frac{1}{K-1} \mathbf{A}^T \mathbf{A} \in \mathbb{R}^{J \times J}.$$

Since $\mathbf{C}$ is symmetric positive semi-definite, it admits an eigendecomposition $\mathbf{C} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^T$, where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_J]$ is an orthogonal matrix of eigenvectors and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_J)$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_J \geq 0$. The eigenvectors $\mathbf{u}_i$ are the EOF modes (spatial basis functions) and the eigenvalues λ_i represent the variance captured by each mode.

In practice, the EOF modes are computed via the singular value decomposition (SVD) of the anomaly matrix, $\mathbf{A}^T = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^T$, which is numerically more stable than forming $\mathbf{C}$ explicitly [10]. The columns of $\mathbf{U}$ are the EOF modes and the singular values satisfy $\sigma_i^2/(K-1) = \lambda_i$. The principal component (PC) coefficients, representing the projection of each profile onto the EOF basis, are given by $\alpha_{ki} = \sum_j A_{kj}\, u_{ij}$. The fraction of total variance explained by the first M modes is

$$\rho_M = \frac{\sum_{i=1}^{M} \lambda_i}{\sum_{i=1}^{J} \lambda_i}.$$

For UQ, we use the truncated EOF expansion to parameterize the sound speed perturbation. The perturbed profile is expressed as:

$$c(z) = \bar{c}(z) + \sum_{i=1}^{M} \xi_i \varphi_i(z),$$

where $\bar{c}(z)$ is the mean profile, $\varphi_i(z)$ are the EOF modes (discretized as $\mathbf{u}_i$), and $\xi_i \sim \mathcal{N}(0, \sigma^2_{\mathrm{EOF}})$ are independent Gaussian random coefficients. The truncation order M is chosen so that ρ_M is close to unity; the rapid spectral decay of the EOF eigenvalues (the first five modes capture 96.1% of the total variance in the Baltic Sea dataset) justifies the use of a low-dimensional parameterization. The EOF basis is optimal in the L^2 sense: no other linear decomposition of the same rank captures more variance [21].

3.3 Surrogate Construction

It is important to note that the surrogate models employed in this work are real-valued: a separate scalar Kriging model is trained independently for each spatial field point (r_j, z_j) of interest, mapping the EOF coefficients $\boldsymbol{\xi}$ to the transmission loss $\mathrm{TL}(r_j, z_j; \boldsymbol{\xi})$ at that location. The surrogate modeling workflow consists of the following steps:

1. **Experimental design.** Generate N sample points $\{\mathbf{x}^{(i)}\}_{i=1}^{N}$ in the input space using Latin hypercube sampling (LHS) [12].
2. **Forward model evaluation.** Run the PE solver at each sample point to obtain the training responses $\{y^{(i)}\}_{i=1}^{N}$.
3. **Kriging fitting.** Train the Kriging model by estimating the trend coefficients $\hat{\boldsymbol{\beta}}$ and the correlation hyperparameters $\hat{\boldsymbol{\theta}}$ via maximum likelihood estimation. We compare four Kriging variants from the SMT library [6]: full Kriging (KRG), Kriging with partial least squares (KPLS), KPLS with additional Kriging refinement (KPLSK), and GPX.
4. **Validation.** Assess surrogate accuracy using RMSE against direct Monte Carlo results.

3.4 Uncertainty Propagation

Once the surrogate is trained and validated, uncertainty propagation is performed by evaluating the Kriging predictor on a large Monte Carlo sample $\{\mathbf{x}^{(j)}\}_{j=1}^{N_{\mathrm{MC}}}$ drawn from $\pi(\mathbf{x})$, where $N_{\mathrm{MC}} \gg N$. This yields empirical estimates of the mean, variance, and full probability distribution of the QoI at negligible computational cost.

4 Numerical Experiments

4.1 Problem Setup

We consider a shallow-water underwater acoustic environment at a frequency of $f = 100$ Hz. The source is located at a depth of $z_s = 20$ m with a beam

width of 2°. The water column depth is $H = 100$ m. The computational domain extends to a range of $r_{\max} = 10$ km and a depth of $z_{\max} = 150$ m (including a 50 m sediment layer). The wide-angle parabolic equation is solved using the PyWaveProp library [20] with a Padé $(2, 3)$ rational approximation, using range and depth steps of $\Delta r = 20$ m and $\Delta z = 1$ m, respectively. The seabed is modeled as a homogeneous half-space with a sound speed of 1700 m/s, density of 1.5 g/cm^3, and attenuation of 1.0 dB/λ.

The sound speed variability is modeled using EOFs derived from Argo float data collected in the Baltic Sea. The mean sound speed profile ranges from approximately 1424 m/s (near the thermocline minimum at ∼57 m depth) to 1468 m/s (near the surface). A total of 82 EOF modes were extracted; however, the variance is strongly concentrated in the leading modes: mode 1 alone explains 70.5% of the total variability, the first two modes capture 88.2%, and the first five modes account for 96.1%. This rapid spectral decay justifies the use of a low-dimensional EOF parameterization for UQ.

Figure 1 illustrates the effect of varying the number of EOF modes M and the perturbation standard deviation (SD)σ on the resulting sound speed profile ensemble. The 3σ spread of the SSP ensemble grows both with M and σ: for $M = 1$, $\sigma = 2.5$, the total spread is only 43.8 m/s (barely exceeding the mean profile range), whereas for $M = 5$, $\sigma = 20$, it reaches 79.1 m/s. The variability is concentrated in the thermocline region (∼9–30 m depth), where the SD peaks at ∼3.5 m/s for $M = 3$, $\sigma = 10$, while below ∼65 m the variability drops below 0.7 m/s. Notably, the EOF modes vanish at the surface and bottom boundaries, so the SSP uncertainty is exactly zero at these extremes regardless of the perturbation amplitude.

In further experiments, the uncertain input parameters are the first $M = 3$ EOF coefficients $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3)$, each drawn independently from a zero-mean Gaussian distribution $\xi_i \sim \mathcal{N}(0, \sigma_{\text{EOF}}^2)$. The perturbed sound speed profile is then $c(z) = \bar{c}(z) + \sum_{i=1}^{M} \xi_i \varphi_i(z)$, where $\bar{c}(z)$ is the mean profile and $\varphi_i(z)$ are the EOF modes. Four levels of SSP uncertainty are considered: $\sigma_{\text{EOF}} = 2.5$, 5, 10, and 20 m/s, spanning mild to extreme environmental variability.

4.2 Monte Carlo Uncertainty Propagation

To provide a ground-truth reference for the surrogate-based UQ, we performed direct (full-model) adaptive Monte Carlo simulations for four levels of SSP uncertainty ($\sigma_{\text{EOF}} = 2.5$, 5, 10, 20 m/s). The adaptive algorithm draws samples in batches of 100, checks convergence of the running mean and SD against a 1% relative tolerance after a minimum of 500 samples, and terminates early if the criterion is met (maximum: 50,000 samples). All four runs converged, requiring 3,000 samples for $\sigma_{\text{EOF}} = 2.5$ and 4,200 samples for the remaining cases. Table 1 summarizes the results at a receiver depth of $z_r = 20$ m.

Several trends are evident. First, the spatially averaged mean TL is remarkably stable across perturbation levels, shifting by only 0.23 dB from $\sigma_{\text{EOF}} = 2.5$ (−54.46 dB) to $\sigma_{\text{EOF}} = 20$ (−54.23 dB). This indicates that the ensemble-averaged propagation loss is insensitive to SSP perturbation amplitude at the

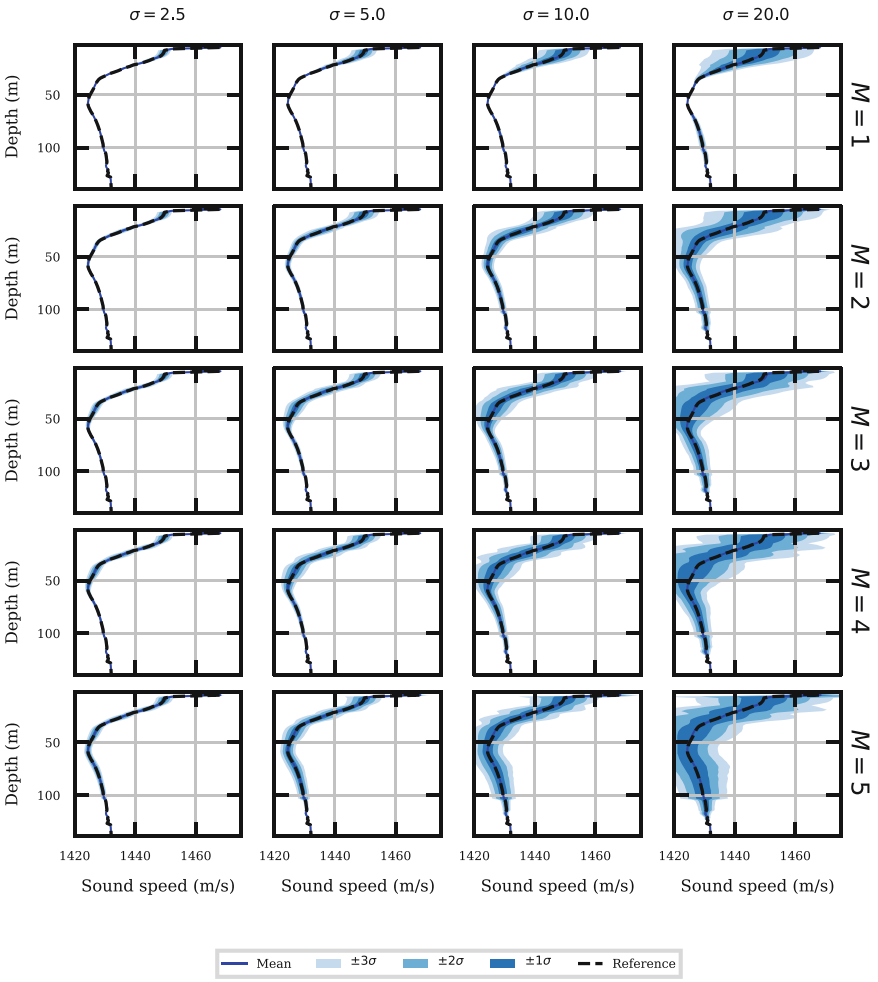


Fig. 1. Sound speed profile ensembles for different numbers of EOF modes M and perturbation SDs σ. Shaded regions indicate $\pm 1\sigma$, $\pm 2\sigma$, and $\pm 3\sigma$ intervals.

global level. However, the pointwise mean TL can differ from the deterministic reference by several decibels (see Fig. 2, bottom panel), which is significant for applications that rely on point-by-point comparisons between measurements and model predictions.

Second, the spatial variability of the mean TL (captured by its spatial SD) decreases monotonically from ± 4.73 dB to ± 2.13 dB as σ_{EOF} increases–the interference pattern is progressively "washed out" by averaging over increasingly diverse SSP realizations.

Third, the pointwise TL uncertainty (SD TL) grows sub-linearly with input uncertainty: an eightfold increase in σ_{EOF} (from 2.5 to 20 m/s) yields only a

Table 1. Direct adaptive Monte Carlo results for different SSP perturbation levels. Mean TL and SD TL are spatial statistics (mean ± spatial SD) of the pointwise mean and SD of the transmission loss field along range at $z_r = 20$ m.

σ_{EOF} (m/s)	Samples	Time (s)	Mean TL (dB)	SD TL (dB)	Max SD TL (dB)
2.5	3 000	137	-54.46 ± 4.73	1.67 ± 1.63	7.90
5	4 200	236	-54.43 ± 4.06	2.76 ± 1.89	8.26
10	4 200	241	-54.36 ± 3.07	3.85 ± 1.70	7.51
20	4 200	235	-54.23 ± 2.13	4.55 ± 1.19	7.83

2.7× increase in mean SD TL (from 1.67 to 4.55 dB), suggesting saturation due to nonlinear mode coupling at large perturbations.

Fourth, the maximum SD TL remains in the range 7.5–8.3 dB regardless of σ_{EOF}, indicating localized sensitivity hotspots, likely near interference nulls and shadow-zone boundaries, where the TL is highly sensitive to SSP changes at any perturbation level.

Figure 2 compares the mean transmission loss, its SD, and the pointwise difference from the deterministic reference for all four perturbation levels. Figure 3 shows the individual $\pm 1\sigma$ uncertainty bands.

4.3 Surrogate Comparison

Four surrogate methods from the SMT library [6] were compared: full Kriging (KRG), Kriging with partial least squares (KPLS), KPLS with additional Kriging refinement (KPLSK), and GPX. Each method was evaluated with $N = 10, 30, 100$, and 300 training samples generated via Latin hypercube sampling [12]. The experiments were repeated for three levels of EOF perturbation SD ($\sigma_{\text{EOF}} = 2.5$, 5, 10), representing increasing environmental uncertainty. Tables 3, 4 and 5 report the training-plus-prediction time per field point (in seconds), the mean RMSE, the SD RMSE (total SD combining aleatory and epistemic uncertainty, compared against direct Monte Carlo), and the epistemic RMSE (uncertainty due to the surrogate model itself). A detailed description of each metric is provided in Table 2.

Several trends emerge from the surrogate comparison. First, full Kriging (KRG) consistently achieves the lowest mean RMSE at every noise level when a sufficient number of training samples is available. This confirms that the full correlation structure is beneficial for this problem. However, KRG is also the most expensive method per field point: at $N = 300$ it requires 6.6–7.6 s/pt, roughly 12–14× slower than GPX at the same sample size.

Second, GPX offers a dramatic speedup, 0.03–0.55 s/pt across all configurations, but its accuracy plateaus or even degrades at $N = 300$. This non-monotonic behavior suggests overfitting or hyperparameter instability in the GPX implementation for this problem class.

Third, KPLSK provides a practical middle ground. Its accuracy at $N = 100$ is close to that of full KRG, while requiring roughly half the time per point. For

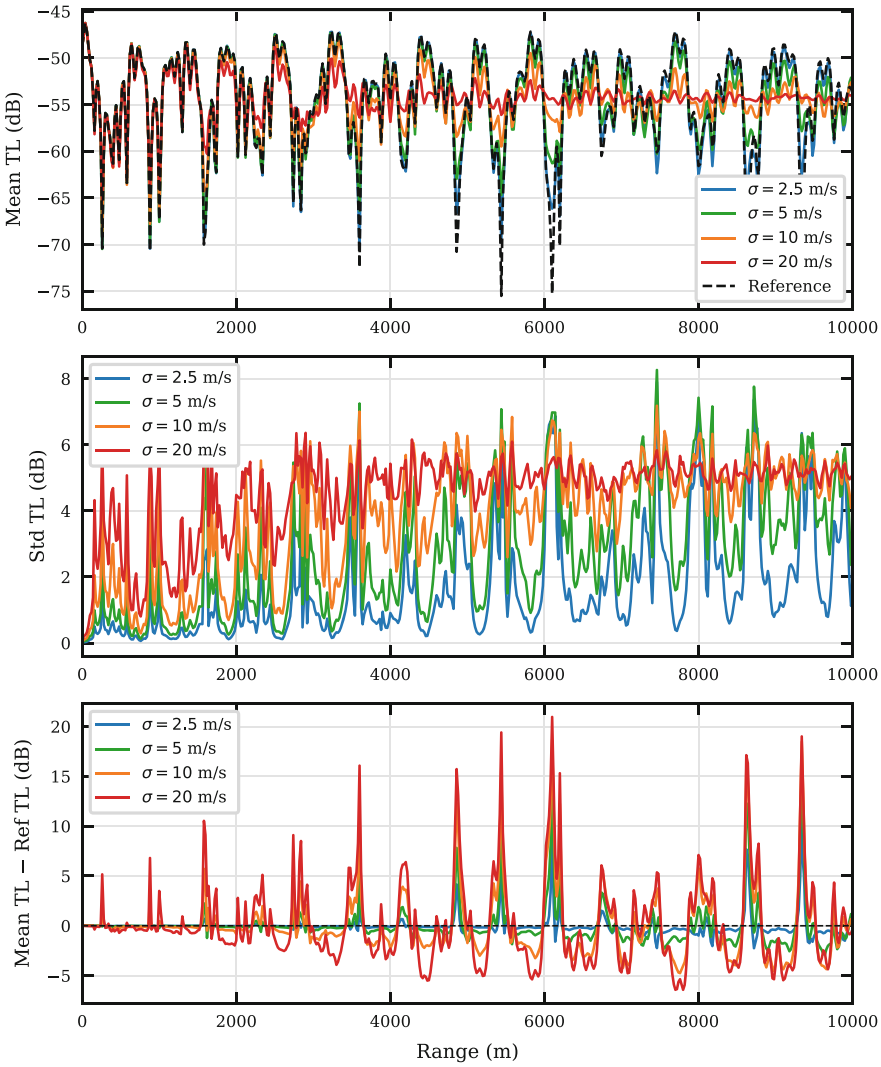


Fig. 2. Direct Monte Carlo results for $\sigma_{\mathrm{EOF}} = 2.5$, 5, 10, and 20 m/s: mean transmission loss (top), SD (middle), and pointwise difference from the deterministic reference (bottom).

moderate uncertainty ($\sigma_{\mathrm{EOF}} \leq 5$) and $N \geq 100$, KPLSK achieves sub-0.32 dB mean RMSE, which is adequate for most practical UQ applications.

The decomposition of prediction uncertainty into total and epistemic components reveals additional insights. The mean epistemic SD (reflecting surrogate model uncertainty) decreases with increasing N for KRG and KPLSK, confirming that these models become more confident as training data grows. In contrast, GPX exhibits elevated epistemic uncertainty at intermediate sample sizes, con-

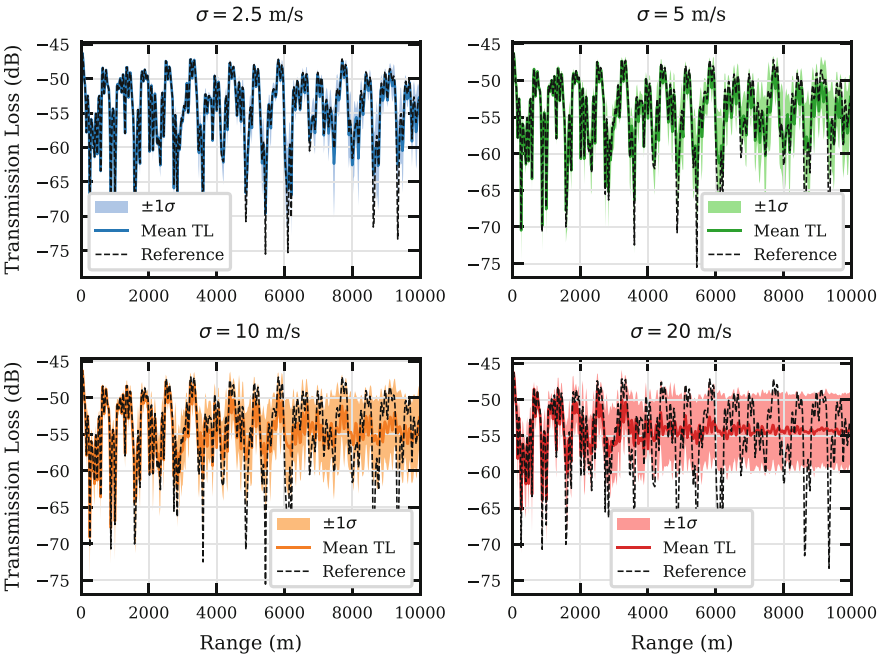


Fig. 3. Transmission loss with $\pm 1\sigma$ uncertainty bands for each EOF perturbation level ($\sigma_{\mathrm{EOF}} = 2.5$, 5, 10, 20 m/s). Dashed line shows the deterministic reference.

Table 2. Description of metrics reported in Tables 3, 4 and 5.

Column	Description
Method	Surrogate model type: KRG (full Kriging), KPLS (Kriging with partial least squares), KPLSK (KPLS with Kriging refinement), or GPX
N	Number of Latin hypercube training samples
Time/pt (s)	Total wall-clock time (training + prediction) per spatial field point, in seconds
Mean RMSE	Root mean square error of the surrogate-predicted mean TL field against the direct Monte Carlo mean, in dB
SD RMSE	Root mean square error of the total (aleatory + epistemic) SD of TL against the direct Monte Carlo SD, in dB. Measures how well the surrogate captures the overall uncertainty, combining both the physical variability due to SSP uncertainty and the surrogate model uncertainty
Mean Epist. SD	Spatial mean of the epistemic (surrogate model) SD of TL, in dB. Quantifies the surrogate's own prediction uncertainty arising from limited training data

Table 3. Surrogate accuracy comparison for $\sigma_{\mathrm{EOF}} = 2.5$.

Method	N	Time/pt (s)	Mean RMSE	SD RMSE	Mean Epist. SD
KRG	10	0.5017	0.7187	0.7641	0.8730
	30	1.0780	0.2488	0.3396	0.7116
	100	1.7031	0.0749	0.1466	0.5104
	300	7.5733	0.1153	0.0764	0.5104
KPLS	10	0.1026	1.2516	0.7761	1.6167
	30	0.2452	0.4188	0.4150	1.2701
	100	0.7021	0.2231	0.3145	1.0868
	300	3.3923	0.2730	0.6042	0.5790
KPLSK	10	0.1506	0.9737	0.7319	1.1323
	30	0.3549	0.2930	0.3341	0.9599
	100	0.8787	0.1140	0.1648	0.5669
	300	3.3725	0.4032	0.5363	0.4989
GPX	10	0.0359	0.7363	0.7511	0.7751
	30	0.0482	0.6519	0.5870	0.8113
	100	0.1187	0.1096	0.2821	0.6822
	300	0.5314	0.4469	0.7200	0.2422

Table 4. Surrogate accuracy comparison for $\sigma_{\mathrm{EOF}} = 5$.

Method	N	Time/pt (s)	Mean RMSE	SD RMSE	Mean Epist. SD
KRG	10	0.4176	1.4256	1.1377	1.4842
	30	0.7153	1.0140	0.7579	1.4319
	100	1.2103	0.2447	0.3208	0.8110
	300	6.8701	0.1341	0.2995	0.5848
KPLS	10	0.0786	1.1715	1.1384	2.4355
	30	0.1437	1.5743	0.8073	2.0606
	100	0.3319	0.4893	0.8173	1.3021
	300	2.2528	0.3508	0.6319	0.9131
KPLSK	10	0.1259	1.1194	1.1541	2.0379
	30	0.2275	1.1781	0.7762	1.4594
	100	0.5135	0.3130	0.6992	0.7605
	300	3.2145	0.2562	0.5584	0.5669
GPX	10	0.0293	1.4840	1.1657	1.3222
	30	0.0421	1.0869	0.6629	2.0065
	100	0.1196	0.2980	0.4665	1.4636
	300	0.5474	0.5254	1.1012	0.6254

Table 5. Surrogate accuracy comparison for $\sigma_{\mathrm{EOF}} = 10$.

Method	N	Time/pt (s)	Mean RMSE	SD RMSE	Mean Epist. SD
KRG	10	0.3310	2.4563	1.9933	1.7990
	30	0.6102	0.8858	0.9423	2.8066
	100	1.1901	0.6842	0.7071	2.1414
	300	6.6497	0.1741	0.3467	1.3658
KPLS	10	0.0645	2.0116	1.7691	3.5809
	30	0.1090	1.1218	1.0079	3.5206
	100	0.2613	0.8275	1.1308	2.7229
	300	1.7783	0.4117	1.6946	1.4364
KPLSK	10	0.1097	2.4120	2.0413	2.0813
	30	0.1908	0.8856	0.9491	2.8347
	100	0.4355	0.6784	0.8009	2.1666
	300	2.5251	0.3153	1.6476	1.0544
GPX	10	0.0285	1.9631	1.9193	2.4935
	30	0.0416	1.1036	1.0867	3.0295
	100	0.1094	0.8129	0.7652	2.6499
	300	0.5407	1.3705	2.4934	0.5575

sistent with its non-monotonic accuracy behavior. The SD RMSE (combining aleatory and epistemic uncertainty, compared against direct Monte Carlo) generally decreases with N, indicating improved overall uncertainty estimation as training data grows. At large N, the epistemic component diminishes and the remaining error is dominated by the aleatory contribution, reflecting the shift from surrogate inadequacy to irreducible input variability.

Finally, increasing σ_{EOF} from 2.5 to 10 roughly triples the mean RMSE across all methods and sample sizes, reflecting the greater complexity of the input–output mapping when the SSP variability spans a wider range. This underscores the importance of choosing an EOF perturbation scale that is consistent with the actual oceanographic uncertainty.

Since each surrogate is a scalar model trained independently per spatial point, the computational advantage of the surrogate approach depends critically on the number of field points at which uncertainty estimates are required. When UQ is needed at a small number of receiver locations–for example, at a fixed sensor array or along a single transect–the one-time cost of training a handful of Kriging models is amortized over a large number of surrogate evaluations, yielding substantial speedups over direct Monte Carlo. However, when the full two-dimensional (or three-dimensional) transmission loss field must be characterized, a separate surrogate must be trained for every grid point, and the aggregate training cost can exceed that of brute-force Monte Carlo, which naturally pro-

duces the entire field in each realization. In such scenarios, direct Monte Carlo may be the more practical choice.

5 Conclusion

We presented a Kriging-based surrogate modeling framework for UQ of underwater sound propagation. The sound speed variability was parameterized using EOFs derived from Argo float data collected in the Baltic Sea; the first five modes capture 96.1% of the observed variance, providing a compact and physically meaningful input space for the surrogate. The wide-angle parabolic equation solver from the PyWaveProp library was used as the high-fidelity forward model, and Kriging implementations from the SMT library served as surrogate approximations.

A systematic comparison of four surrogate methods (KRG, KPLS, KPLSK, GPX) revealed that full Kriging achieves the best accuracy, while KPLSK offers a favorable accuracy–cost trade-off for moderate uncertainty levels. GPX provides orders-of-magnitude speedup per field point but exhibits accuracy limitations for this problem class. The decomposition of prediction uncertainty into aleatory and epistemic components showed that the total (aleatory + epistemic) SD provides a more complete comparison against direct Monte Carlo, and that as the surrogate improves with more training data, the dominant error source shifts from model inadequacy to irreducible input variability. Increasing the EOF perturbation scale from $\sigma_{\mathrm{EOF}} = 2.5$ to 10 roughly triples the surrogate error, highlighting the need for sufficient training data under high environmental uncertainty.

Direct adaptive Monte Carlo simulations with 3,000–4,200 full-model evaluations revealed two key findings. First, the mean transmission loss field under SSP uncertainty can differ significantly from the deterministic prediction on a point-by-point basis (up to several decibels), even though the spatially averaged TL remains stable. This has important practical implications: when comparing simulation results with field measurements, one should account for the bias introduced by environmental uncertainty rather than relying on a single deterministic run. Second, the qualitative structure of the transmission loss field (the overall pattern of convergence and shadow zones) is remarkably robust to SSP uncertainty. The TL uncertainty grows sub-linearly with SSP perturbation amplitude, while localized sensitivity hotspots ($\sim$8 dB) persist regardless of the perturbation level.

Future work will explore several directions. First, vector-valued surrogate models that jointly predict the transmission loss at multiple spatial points could share information across locations, improving accuracy and eliminating the need to train a separate model per field point–thereby making surrogate-based UQ competitive even for full two-dimensional field characterization. Second, adaptive sampling strategies (e.g., expected-improvement or variance-reduction criteria) would allow the training set size to be determined automatically rather than fixed a priori, reducing the number of required forward model evaluations

while ensuring a target surrogate accuracy. Additional extensions include higher-dimensional input spaces incorporating bathymetry and seabed parameters, and validation against measured oceanographic data.

Acknowledgments. This study was supported by the Russian Science Foundation grant No. 25-71-00028.

References

1. Ali, W.H.: Dynamically orthogonal equations for stochastic underwater sound propagation. Ph.D. thesis, Massachusetts Institute of Technology (2019)
2. Ali, W.H., Lermusiaux, P.F.: Dynamically orthogonal narrow-angle parabolic equations for stochastic underwater sound propagation. Part II: applications. J. Acoust. Soc. Am. **155**(1), 656–672 (2024)
3. Blatman, G., Sudret, B.: Adaptive sparse polynomial chaos expansion based on least angle regression. J. Comput. Phys. **230**(6), 2345–2367 (2011)
4. Bouhlel, M.A., Bartoli, N., Otsmane, A., Morlier, J.: An improved approach for estimating the hyperparameters of the kriging model for high-dimensional problems through the partial least squares method. Math. Probl. Eng. **2016**(1), 6723410 (2016)
5. Bouhlel, M.A., Bartoli, N., Otsmane, A., Morlier, J.: Improving kriging surrogates of high-dimensional design models by partial least squares dimension reduction. Struct. Multidiscip. Optim. **53**(5), 935–952 (2016)
6. Bouhlel, M.A., Hwang, J.T., Bartoli, N., Lafage, R., Morlier, J., Martins, J.R.: A python surrogate modeling framework with derivatives. Adv. Eng. Softw. **135**, 102662 (2019)
7. Collins, M.D.: A split-step padé solution for the parabolic equation method. J. Acoust. Soc. Am. **93**(4), 1736–1742 (1993)
8. Du, L., Wang, Z., Lv, Z., Wang, L., Han, D.: Research on underwater acoustic field prediction method based on physics-informed neural network. Front. Mar. Sci. **10**, 1302077 (2023)
9. Finette, S.: A stochastic response surface formulation of acoustic propagation through an uncertain ocean waveguide environment. J. Acoust. Soc. Am. **126**(5), 2242–2247 (2009)
10. Golub, G.H., Van Loan, C.F.: Matrix Computations. JHU Press (2013)
11. Hannachi, A., Jolliffe, I.T., Stephenson, D.B., et al.: Empirical orthogonal functions and related techniques in atmospheric science: a review. Int. J. Climatol. **27**(9), 1119–1152 (2007)
12. Helton, J.C., Davis, F.J.: Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. Reliab. Eng. Syst. Saf. **81**(1), 23–69 (2003)
13. Jensen, F.B., Kuperman, W.A., Porter, M.B., Schmidt, H., Tolstoy, A.: Computational Ocean Acoustics, vol. 2011. Springer, New York, NY (2011). https://doi.org/10.1007/978-1-4419-8678-8
14. Johnson, G.C., et al.: Argo–two decades: global oceanography, revolutionized. Ann. Rev. Mar. Sci. **14**(1), 379–403 (2022)
15. Kant Deo, I., Venkateshwaran, A., Jaiman, R.K.: A physics-guided probabilistic surrogate modeling framework for digital twins of underwater radiated noise. arXiv e-prints arXiv:2509.25730 (2025)

16. Khazaie, S., Wang, X., Komatitsch, D., Sagaut, P.: Uncertainty quantification for acoustic wave propagation in a shallow water environment. Wave Motion **91**, 102390 (2019)
17. Khine, Y.Y., Creamer, D.B., Finette, S.: Acoustic propagation in an uncertain waveguide environment using stochastic basis expansions. J. Comput. Acoust. **18**(04), 397–441 (2010)
18. Lafage, R.: Egobox, a Rust toolbox for efficient global optimization. J. Open Sour. Softw. **7**(78), 4737 (2022)
19. Lee, B.M., Johnson, J.R., Dowling, D.R.: Predicting acoustic transmission loss uncertainty in ocean environments with neural networks. J. Mar. Sci. Eng. **10**(10), 1548 (2022)
20. Lytaev, M.S.: Rational interpolation of the one-way helmholtz propagator. J. Comput. Sci. **58**, 101536 (2022)
21. Preisendorfer, R.W.: Principal Component Analysis in Meteorology and Oceanography. Elsevier, Amsterdam (1988)
22. Sudret, B.: Global sensitivity analysis using polynomial chaos expansions. Reliab. Eng. Syst. Saf. **93**(7), 964–979 (2008)
23. Williams, C.K., Rasmussen, C.E.: Gaussian Processes for Machine Learning, vol. 2. MIT Press, Cambridge, MA (2006)
24. Xiu, D.: Numerical Methods for Stochastic Computations: A Spectral Method Approach. Princeton University Press (2010)
25. Xiu, D., Karniadakis, G.E.: The wiener-askey polynomial chaos for stochastic differential equations. SIAM J. Sci. Comput. **24**(2), 619–644 (2002)

Low-Cost Wearable System for Essential Tremor Monitoring with Passive Damping

Gustavo Adolpho Bonesso[1], Carlos Marcelo Gurjão de Godoy[1(✉)], Eliana Chaves Ferretti[2], and Regina Célia Coelho[1]

[1] Science and Technology Institute, Federal University of São Paulo, São José Dos Campos, SP, Brazil
gurjao.godoy@unifesp.br

[2] Institute of Health and Society, Federal University of São Paulo, Santos, SP, Brazil

Abstract. Essential tremor (ET) is a progressive neurological disorder causing involuntary upper-limb oscillations that impair fine motor control and daily independence. Monitoring and attenuating such biomechanical vibrations represent a significant challenge in wearable device design. Using ET as a practical use case—characterized by oscillations in the 2–10 Hz range—this study presents the design and development of a low-cost, customizable embedded system that helps overcome that challenge. The hardware consists of a passive weighted bracelet fabricated via additive manufacturing, integrated with an ESP32-based inertial sensing module for continuous data acquisition. To reduce the microcontroller's computational load, the system employs an onboard time-domain version of a traditional frequency-domain period extraction algorithm. The sensing architecture was evaluated through a controlled damped-harmonic-oscillator bench test, using a high-precision laboratory rotary motion sensor as the ground-truth reference. When testing the empty bracelet, the embedded accelerometer recorded a frequency of oscillation of 2.99 Hz (SD = 0.044) and decreased to 2.50 Hz (SD = 0.004) with the addition of a 137 g weight to the bracelet. The change in the oscillatory frequency shows the principle of passive inertial damping and the success in using the embedded hardware for ET monitoring.

Keywords: essential tremor · assistive technology · weighted bracelet · inertial sensors · additive manufacturing

1 Introduction

Essential tremor (ET) is a chronic neurological movement disorder characterized by involuntary rhythmic oscillations of the upper limbs. The oscillation frequency range is normally from 2 Hz to 10 Hz [1, 2]. ET affects fine motor control, impacting the patient's autonomy in daily activities like eating, writing, and personal care [3, 4]. It doesn't reduce the patient's life expectancy, but the high prevalence of the disorder in the elderly impacts public health [1].

Conventional treatment options include pharmacological methods involving beta-blockers, anticonvulsants, and botulinum toxin injections, or invasive procedures, such

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 67–78, 2026.
https://doi.org/10.1007/978-3-032-30491-9_5

as deep-brain stimulation (DBS) in severe cases [5]. However, these options have significant limitations that includes side effects, varying effectiveness, high costs, and limited availability, especially in low-resource settings [5]. These issues have motivated increasing interest in alternative or complementary solutions that are non-invasive, affordable, and easy to use.

Assistive technologies that use mechanical concepts are gaining prominence and importance. Weighted gloves and wristbands are passive devices that use mechanical concepts such as mass inertia to reduce tremors by altering the dynamic response of the limb without using active control systems [6, 7]. Some users of these passive devices report functional improvements, but others indicate differences in clinical evidence, showing that many solutions still need to be systematically implemented [8].

Wearable sensor technologies and mobile health systems have advanced, permitting continuous and objective monitoring of movement and related symptoms in real-life environments [9–11]. Inertial measurement units (IMUs), which include accelerometers and gyroscopes, are used to monitor and estimate tremor characteristics such as frequency of oscillation, amplitude, and the variation of these parameters over time [7, 9, 11]. However, these systems are focused on monitoring tremors and are not capable of direct physical intervention or successfully reducing tremors. This shows that there is a gap in combining passive tremor reduction strategies with embedded digital sensing systems. Filling this gap can be beneficial in two ways: reducing tremors and enabling objective evaluation of the effectiveness of these devices, increasing the customization of solutions, and improving clinical validation.

This study suggests the design and development of a low-cost, customizable wearable system, combining a passive wristband with weight recipients and an embedded tremor sensing module. The system utilizes additive manufacturing to allow for design flexibility and rapid iterations, and an embedded sensing system with software optimized for low operating costs.

The main contribution of this work is the development of a device that combines passive inertial damping with an integrated embedded sensing module, enabling real-time tremor monitoring, as validated through controlled bench tests. Below is a list of particular contributions of this work:

1. The design and implementation of a prototype device that combines passive mechanical tremor attenuation with integrated real-time monitoring, filling the identified gap between assistive technologies and embedded sensor monitoring.
2. A method that allows the calculation of frequency of oscillation using raw accelerometer data, based on time-domain analysis, which is computationally optimized and, consequently, suitable for embedded systems having minimal processing capacity and low power consumption requirements.
3. An integrated, low-cost prototype fabricated using additive manufacturing techniques (3D printing) offering a flexible device for customized tremor management.
4. An experimental assessment method for the bracelet's embedded sensing module using a controlled mass-spring system equipped with reference sensors, enabling repeatability and quantitative evaluation of the system's performance.

5. Experimental evidence showing the effectiveness of passive inertial damping, confirming the viability of this solution as a wearable device for tremor reduction in patients.

2 Related Works

This research on assistive technologies for essential tremor (ET) developed a method to reduce tremor using a bracelet with embedded sensors for tremor assessment and monitoring. Methods to reduce ET can be classified into passive and active systems. As examples of passive devices, we can mention weighted gloves, wristbands, and orthopedic supports. Physical principles such as the addition of mass, inertia, and damping are used to reduce oscillatory movement [4, 6]. Passive solutions have low complexity, high safety, and ease of use, making them appropriate for applications in daily life. However, their effectiveness depends heavily on individual factors, such as tremor frequency and amplitude, as well as user habituation to their use. Recent studies and systematic reviews show that clinical evidence from the use of passive devices shows diverse results, with little standardization in assessment protocols and outcome measures [8].

Active and semi-active systems reduce tremor by controlled actuation, using wearable frameworks and stabilization devices that use gyroscopes. These systems produce compensatory forces that suppress the user's involuntary movement [8]. These methods usually result in higher levels of tremor reduction, but they also have some disadvantages in relation to passive devices. The systems are more complex, heavier, consume more energy, and are more expensive. These limitations affect usability and user acceptance, especially for long-term use and in non-clinical settings.

Wearable sensor technologies for the assessment of movement disorders have also been developed. Inertial measurement units (IMUs), composed of accelerometers and gyroscopes, are used to measure tremor characteristics such as frequency, amplitude, and their variation over time [7, 9, 11]. With these sensors, continuous monitoring is possible, supporting both clinical assessment and remote health applications. With these technologies integrated into mobile health platforms, the possible applications increase, allowing data collection under practical conditions [9–11].

Using signal processing, tremor analysis is typically performed using frequency-domain techniques, such as the Fast Fourier Transform (FFT), which allows the identification of dominant oscillatory components [11]. However, these approaches can demand high processing power, making their use unfeasible on low-power embedded platforms. It is necessary to develop alternative methods that can reduce the demand for processing power, but that have good accuracy for real-time use [11].

Many systems that aim to reduce tremors do not have integrated monitoring capabilities, making it difficult to objectively measure their effectiveness. Wearable sensors provide detailed data on movement, but do not directly reduce tremors.

In this way, personalized and data-centric assistive technologies are limited. Therefore, there is a need to develop integrated solutions that can act to reduce tremor but also have embedded sensors as well as real-time processing. Systems with these combined characteristics have the potential to provide functional assistance and quantitative monitoring simultaneously, granting more effective clinical evaluation, personalization, and validation of assistive devices for TE.

3 Proposed System Architecture

This work proposes to develop a low-cost system aimed at reducing tremors and enabling their monitoring. The system combines a passive mechanical component – the bracelet – with embedded electronic sensors, creating a solution that integrates physical assistance for tremor reduction and data collection on a single platform. Figure 1 shows the proposed system, composed of two main components: (i) a passive weighted bracelet, manufactured by additive manufacturing, and (ii) an embedded inertial sensor module.

3.1 Mechanical Design and Additive Manufacturing

The bracelet was designed in modules with a compartment for inserting weights (lead material), allowing for an adjustable mass of up to 600 g. The prototypes were made using PLA filaments, allowing for various iterations and evaluation of ergonomics, flexibility, and durability. The choice of PLA (polylactic acid) for the initial prototypes resulted in a lightweight yet rigid bracelet capable of supporting the weights used in the initial tests.

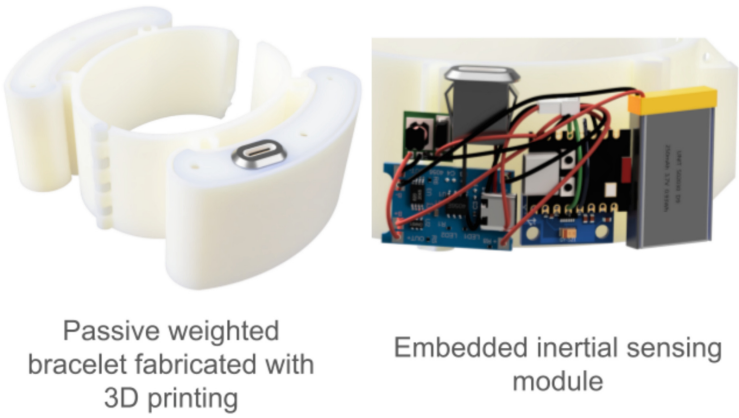


Fig. 1. Overview of the proposed system. On the left, we show the passive weighted wristband, manufactured using additive manufacturing, with the weight distribution containers. On the right, we have the embedded inertial sensor module, showing the ESP32 microcontroller, the triaxial accelerometer, and the power supply.

The magnetic clasps were designed to make it easier to close the bracelet with just one hand, as patients may have difficulty due to motor limitations caused by tremors. The design process involved several cycles of prototyping and refinement.

3.2 Embedded Electronics and Sensing

The design of the solution's electronic module took into account the need for low power consumption to extend usage time without recharging, reliable sensor data acquisition, and a compact size that would allow the module to be embedded in the

bracelet. Figure 2 shows the functional block diagram. The central component is the ESP-32 microcontroller, a 32-bit single-core RISC-V processor suitable for embedded processing.

Motion data is collected by an MPU-6050 Inertial Measurement Unit (IMU) via the accelerometer. The IMU communicates with the microcontroller using the I2C protocol through the SDA and SCL pins. The acceleration of the three axes is captured at a sampling rate of 100 Hz, exceeding the Nyquist rate requirements for the typical frequency range of tremors we want to capture, which is between 2 Hz and 10 Hz. In this way, the vibration signal information can be successfully collected.

The power for the microcontroller and IMU is provided by a 3.7V Lithium Polymer (Li-Po) battery. The charging of the battery is done via a USB-C port using a TP4056 battery charger, which ensures safe charging. The microcontroller and IMU are powered with 5V, so a DC-DC (step-up) converter is used to increase the battery voltage from 3.7V to 5V. This embedded hardware configuration provides for uninterrupted monitoring of tremor frequency.

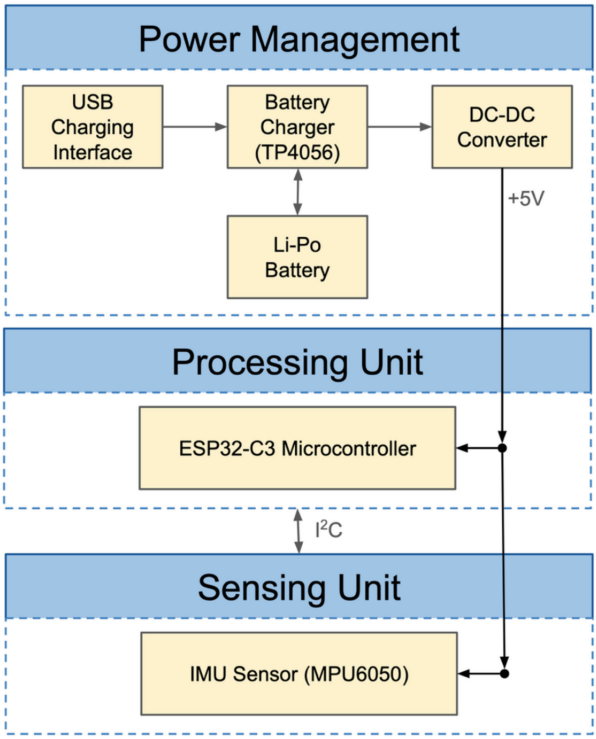


Fig. 2. The functional block diagram of the electronic module is divided into three parts: The power management unit, where battery charging occurs using the TP4056 and the battery voltage is increased from 3.7V to 5V; the processing unit using the ESP-32 microcontroller; and the sensing unit using the MPU-6050 accelerometer to monitor movement. The I^2C communication protocol between the microcontroller and the IMU, and the 5V power supply lines, are also shown in the diagram.

4 Methods

We used continuous improvement methodologies and iteration cycles in the development of this work, with a technological research approach and an emphasis on technical feasibility.

4.1 Bench Testing and Sensor Validation

To validate the sensing module, we performed a bench test using a damped harmonic oscillator model. The bracelet, printed using additive manufacturing (3D printing), was suspended by a spring mechanism that approximately replicate the oscillation of a human arm at a given frequency. Figure 3 shows the experimental arrangement used for sensor validation.

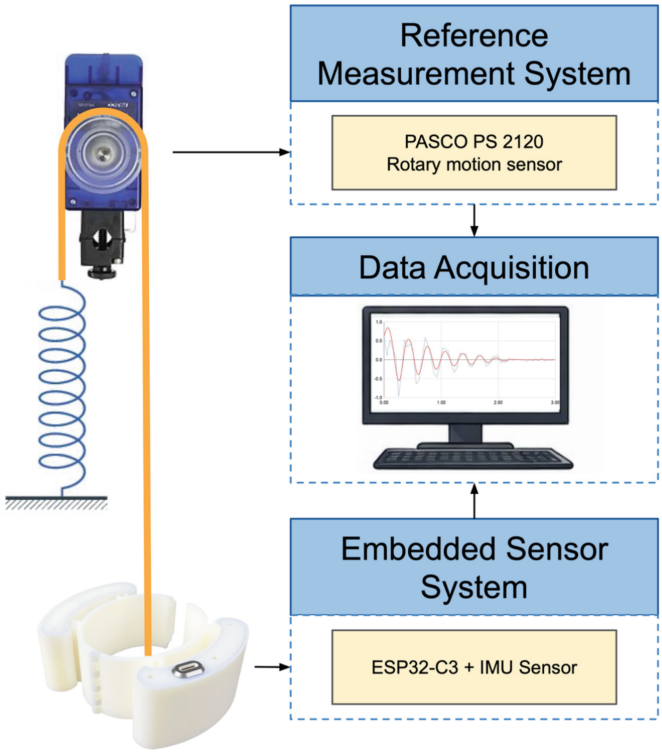


Fig. 3. Setup of the damped harmonic oscillator bench test connecting the mass-spring system to the bracelet, carried out to validate the bracelet's embedded sensing module. The functional block diagram shows that data from the bracelet's sensing module is collected simultaneously with data from the PASCO rotary motion sensor for later comparison and validation.

Data Acquisition Protocol: The experiment protocol recorded the system oscillation using the first configuration, with the weight container empty (0 g), and the second

configuration, with 137 g of mass (lead material) added to the weight container. Four independent trials were performed for each of these two conditions for ensuring statistical repeatability.

Signal Processing Optimization: The Fast Fourier Transform (FFT) is an algorithmically expensive frequency domain transformation. To calculate the frequency of oscillation using only the raw triaxial acceleration data obtained directly from the MPU-6050 sensor at a sampling rate of 100 Hz, we used only the time domain. The period (T) between consecutive peaks of the acceleration signal is measured, and the frequency is calculated ($f = 1/T$). This method decreases the processing load and power consumption of the ESP-32 microcontroller, keeping the accuracy of frequency calculation in the range of 2 to 10 Hz.

The data obtained by the bracelet's embedded sensing module were validated using a high-precision PASCO rotary motion sensor (PS-2120). This sensor was coupled to the test setup to serve as a reference. The PASCO sensor data were recorded using DataStudio software at a sampling rate of 100 Hz, simultaneously with the data acquisition performed by the bracelet's embedded system. The comparison between the period (T) measured between consecutive peaks in the graph with the PASCO sensor data and the data obtained by the bracelet's embedded system shows the accuracy of the low-cost embedded system developed in this work.

5 Results

Bench testing confirmed both the reliability of the embedded sensor and the effectiveness of the mechanical design. To evaluate the repeatability of the time-domain peak-to-peak calculation method, four independent trials were conducted for each mass configuration.

The oscillation frequency measured simultaneously by either the PASCO rotary motion sensor and the Embedded Sensor System, was essentially the same. Figure 4 compares the signals collected by the PASCO reference sensor and the signal collected by the bracelet's embedded sensing module over time.

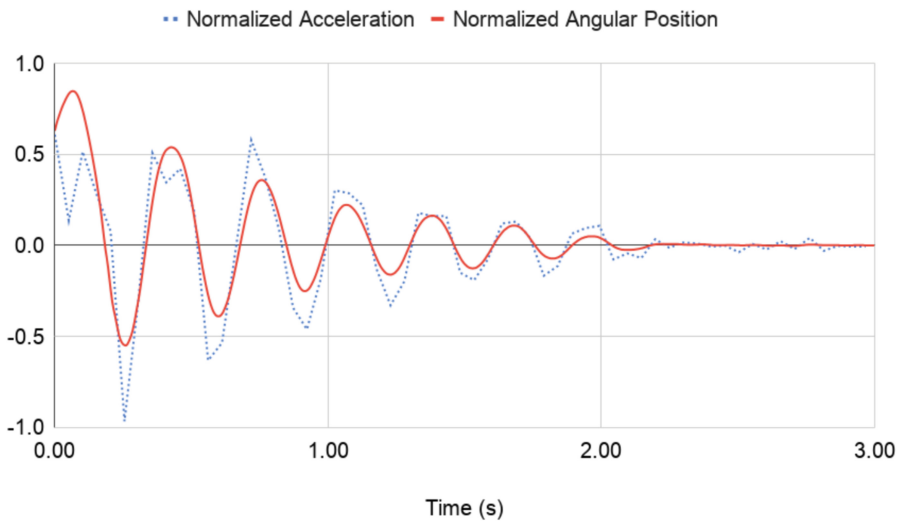


Fig. 4. The graph shows the comparison between the signal collected by the PASCO reference sensor and the signal collected by the bracelet's embedded sensing module over time. The red curve shows the normalized angular position collected by the high-precision PASCO laboratory sensor, and the dashed blue curve shows the normalized raw acceleration data collected by the embedded sensing module. The alignment between the signals is consistent, peak-to-peak, validating the use of the embedded sensing module and the integrated period extraction method for frequency calculation.

As shown in Table 1, the ESP32 successfully and consistently measured an average oscillation frequency of 2.99 Hz (SD = 0.044) when the bracelet's weight compartment was empty. After filling the internal container to its maximum mass capacity (137 g), the average measured oscillation frequency reliably decreased to 2.50 Hz (SD = 0.004). The tests showed a low standard deviation, indicating that the bracelet's embedded sensing module successfully collected data stably and that the frequency calculation algorithm worked well.

Table 1. Oscillation frequencies measured in the damped harmonic oscillator bench test trials performed with the bracelet loaded and unloaded.

Test Condition	Trial 1 (Hz)	Trial 2 (Hz)	Trial 3 (Hz)	Trial 4 (Hz)	Mean (Hz)	SD (Hz)
Empty Compartment (0 g)	3.03	3.01	2.97	2.93	2.99	0.0441
Fully Weighted (137 g)	2.49	2.50	2.49	2.50	2.50	0.0036

The frequency change validates the principle of passive inertial damping, according to mass-spring dynamics. Adding mass reduces the natural frequency of the mechanical system. The results show that the bracelet's embedded sensing module is able to collect

the dynamic vibrations produced in the bench test and that the use of containers with variable mass weight is a viable engineering solution to attenuate oscillations.

6 Discussion

Table 2 compares the contribution of the system proposed in this work with other existing technologies found in the literature. We compare the system type, tremor attenuation capacity, integrated monitoring in the solution, and computational cost.

The combination of passive mechanical tremor attenuation with embedded digital monitoring addresses a gap in existing assistive technologies for essential tremor. The comparison table shows that existing solutions force a choice between the attenuation capacity offered by passive devices and the objective monitoring offered by intelligent devices. The hybrid system proposed in this work combines both features in a single solution with a low-cost platform.

Table 2. Comparison of the characteristics of the system proposed in this work with other existing assistive technologies for tremor.

Approach/Reference	System Type	Tremor Attenuation	Objective Monitoring	Computational Cost
Proposed System	Hybrid/Wearable	Yes (Passive)	Yes	Low (Time-domain analysis)
Traditional Passive Devices (e.g., weighted gloves)	Mechanical/Wearable	Yes (Passive)	No	None
Active Exoskeletons	Robotic/Wearable	Yes (Active)	Yes	High (Complex control loops)
Smartwatches/IMU Trackers	Electronic/Wearable	No	Yes	Medium/High (FFT-based)

According to the principles of biomechanics, the experimental results obtained in the bench test validate the principle of inertial damping. According to the dynamic mass-spring model ($w = \sqrt{k/m}$), an increase in the system's mass results in a diminution in its natural frequency of oscillation, and the data collected by the bracelet's embedded sensing module and validated by the PASCO reference sensor confirm this physical phenomenon. The oscillation rate changes from 2.99 Hz to 2.50 Hz with the addition of 137 g of mass to the bracelet's weight container. Standard deviations of 0.044 for 2.99 Hz and 0.004 for 2.50 Hz reinforce the reliability and repeatability of the measurements performed by the bracelet's embedded sensing module.

The validation of the data collected by the bracelet's embedded sensing module with the high-precision PASCO laboratory sensor (Fig. 4) shows that the calculation of the oscillatory frequency does not require complex frequency domain transformations, such as FFT, which is good news from a computational point of view. The microcontroller

processing load was significantly reduced by using the implemented time-domain period extraction algorithm. This optimization helps increase the battery life of the bracelet and other wearable health devices that need to calculate frequency based on acceleration information.

The technical results are promising, but a number of factors need to be considered when applying the proposed system in real-life situations.

There are many differences in tremor characteristics (frequency, amplitude, and direction) between different patients, and these differences are challenging for developing an efficient passive inertial damping device. The device was validated in the typical tremor frequency range, between 2 and 10 Hz, but individual differences can affect the device's success in real life use. Attenuation depends on the relationship between the mass added to the device in its weight container and the natural frequency of oscillation of the limb. Therefore, using a fixed mass setting may not work for all users. Customized calibration strategies have to be developed to adjust the mass added to the device according to each patient's tremor profile.

Each patient may have an ideal mass for their case, and the modular design of the bracelet allows for this customization, but using excessive mass may cause discomfort, fatigue, and reduce usability for the patient. Conversely, using insufficient mass may not provide the necessary attenuation to reduce tremors. The embedded sensing module can be used to perform necessary adjustments based on collected data, allowing iterative changes to the added mass based on tremor measurements obtained by the system.

The effect of noise within real-world environments can affect the amplitude and frequency estimation generated by the bracelet's integrated sensor module. The peak-to-peak time-domain method showed good accuracy in the controlled, low-noise bench-top environment, but real-world signals are affected by voluntary movements, sensor misalignment, and environmental disturbances, possibly decreasing performance with irregular or non-periodic movements. The use of pre-processing techniques, such as bandpass filtering in the oscillatory frequency range (2–10 Hz) or the combination of multiple sensing axes, can increase robustness in real-life situations.

There are differences between time-domain and frequency-domain approaches, such as the Fast Fourier Transform (FFT). Frequency-domain methods are widely used because they are good at detecting principal oscillatory components, especially in signals with high noise or multiple frequencies. However, the trade-off is that these methods require greater processing power and are not ideal for embedded systems like the one we are proposing in this work. The proposed time-domain method, on the other hand, is much less complex and demands less processing power, making it more suitable for use in embedded systems and wearable devices. However, this method is more vulnerable to noise and signal irregularities and may be less precise when dealing with overlapping frequencies and non-stationary signals. Therefore, the choice of the appropriate method should depend on the specific requirements of the application, balancing computational effectiveness and robustness.

As a whole, the proposed system presents technical feasibility and an integration advantage, but an evaluation under real-world conditions with patients is necessary to have a clearer view of its clinical application in the real world, as well as its usability and long-term efficacy.

7 Future Work

Future work includes clinical validation with patients diagnosed with essential tremor in real-world settings to assess the system's robustness against noise from voluntary movements. Long-term studies are also needed to evaluate the device's ergonomic comfort, quantify continuous battery life, and determine patient adherence to its use. Furthermore, future research will focus on developing a standardized clinical protocol to figure out the optimal mass for each patient, utilizing the embedded sensing module to personalize the attenuation strategy.

8 Conclusion

This study developed a low-cost, 3D-printed wearable system that addresses a gap in essential tremor management by integrating passive mechanical attenuation with real-time embedded monitoring. The implementation of an onboard time-domain period extraction algorithm proved highly effective, allowing accurate tremor frequency calculation with minimal computational load. Furthermore, the evaluation using a controlled mass-spring setup and ground-truth sensors provided a rigorous and repeatable framework to validate the embedded electronics. The experimental evidence confirmed the principle of passive inertial damping, demonstrating a consistent reduction in oscillation frequency when mass was added to the device. Preliminary results show the technical feasibility of the solution and the possibility of combining passive mechanical devices with digital health technologies. The system proposed in this work constitutes a starting point for future clinical studies aimed at improving the autonomy and quality of life of individuals with essential tremor.

Author Contributions. Gustavo Adolpho Bonesso: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Writing – original draft, review & editing.

Carlos Marcelo Gurjão de Godoy: Conceptualization, Supervision, Methodology, Writing – review & editing.

Eliana Chaves Ferretti: Conceptualization, Methodology, Supervision, Validation, Investigation, Resources, Data curation.

Regina Célia Coelho: Conceptualization, Supervision, Methodology, Project administration, Writing – review & editing.

Funding. This work was developed using resources obtained from the FINANCIADORA DE ESTUDOS E PROJETOS (FINEP) - Ref. Finep nº 2174/22.

Disclosure Statement. The authors declare that they have no conflict of interest.

References

1. Louis, E.D., McCreary, M.: How common is essential tremor? Update on the worldwide prevalence of essential tremor. Tremor Other Hyperkinet. Mov. **11**, 28 (2021). https://doi.org/10.5334/tohm.632

2. Elble, R.J.: What is essential tremor? Topical collection on movement disorders. Curr. Neurol. Neurosci. Rep. **13**, 353 (2013). https://doi.org/10.1007/s11910-013-0353-4
3. Bain, P.G., Findley, L.J., Atchison, P., et al.: Assessing tremor severity. J. Neurol. Neurosurg. Psychiatry **56**, 868–873 (1993). https://doi.org/10.1136/jnnp.56.8.868
4. Manchanda, M.: Gyro glove: stabilizing the lives of those with hand tremor **20**, 4781–4794 (2021). https://doi.org/10.17051/ilkonline.2021.04.514
5. Zesiewicz, T.A., Shaw, J.D., Allison, K.G., et al.: Update on treatment of essential tremor. Curr. Treat. Options. Neurol. **15**, 410–423 (2013). https://doi.org/10.1007/s11940-013-0239-4
6. Adabi, K., Ondo, W.G.: Shaking up essential tremor: peripheral devices and mechanical strategies to reduce tremor. Tremor Other Hyperkinet. Mov. (N. Y). **14**, 55 (2024)
7. Vescio, B., De Maria, M., Crasà, M., et al.: Development of a new wearable device for the haracterization of hand tremor. Bioengineering **10**, 1025 (2023). https://doi.org/10.3390/bioengineering10091025
8. Lora-Millan, J.S., Delgado-Oleas, G., Benito-León, J., Rocon, E.: A review on wearable technologies for tremor suppression. Front. Neurol. **12** (2021)
9. Caroppo, A., Manni, A., Rescio, G., et al.: Movement disorders and smart wrist devices: a comprehensive study. Sensors **25**, 266 (2025). https://doi.org/10.3390/s25010266
10. Del Din, S., Elshehabi, M., Galna, B., et al.: Gait analysis with wearables predicts conversion to parkinson disease. Ann. Neurol. **86**, 357–367 (2019). https://doi.org/10.1002/ana.25548
11. Peng, Y., Ma, C., Li, M., et al.: Intelligent devices for assessing essential tremor: a comprehensive review. J. Neurol. **271**, 4733–4750 (2024)

Analyzing Internal Sensitivity in Neural Network Representations via Jacobian Norms and Perturbation Analysis

Aninda Kumar Sharma(✉)

School of Computer and Mathematical Sciences, University of Adelaide, Adelaide, SA 5005, Australia
anindakumar.sharma@student.adelaide.edu.au

Abstract. Deep neural networks are widely used in computational and machine learning applications. However, deep neural networks with similar predictive power may have vastly different internal representation responses to perturbations. This is important in the computational analysis of neural networks because predictive power and input-output smoothness are not sufficient to describe the internal organization and output response in deep neural networks. This paper proposes a geometric concept called opacity to describe this intrinsic sensitivity in deep neural networks. Opacity is measured using the Jacobian norms in individual layers and the internal perturbation radius. This is a quantitative method to describe the influence of internal representations in deep neural networks. The paper also shows that opacity is not reducible to global input-output sensitivity. Architectures with comparable input sensitivity can differ substantially in internal perturbation radius. This suggests that sensitivity in deep neural networks is an intrinsic property. Because opacity is computed from standard forward passes without adversarial perturbations or auxiliary probes, it enables direct cross-architecture comparison under matched accuracy conditions. This paper also evaluates multilayer perceptrons, convolutional networks, and vision transformers using the MNIST and CIFAR-10 datasets. Our results show that architectures with comparable accuracy can differ by up to an order of magnitude in internal perturbation radius, highlighting the role of architectural inductive bias in shaping representational geometry. The findings suggest that opacity provides a quantitative and computationally grounded insight in analysing internal bottlenecks and stability properties in deep neural networks.

Keywords: Neural network · Internal sensitivity · Representation learning · Jacobian norms · Perturbation analysis

1 Introduction

Deep neural and transformer-based models are now widely used across computational machine learning applications, including natural language processing,

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 79–95, 2026.
https://doi.org/10.1007/978-3-032-30491-9_6

visual recognition, and cyber-security tasks [1–3]. This broad use of neural models motivates the need to study not only their predictive performance but also the internal structure of their learned representations. Computational analysis of neural networks typically emphasize predictive performance, input–output smoothness, or robustness under external perturbations. However, these views are limited as these do not show how changes relate to representations that are in between the final representation and the corresponding input. When models with similar predictive accuracy exhibit different internal stability regimes, a quantitative measure of hidden-state sensitivity can help identify architectural differences that are not visible from output performance alone. Internal layerwise sensitivity analysis is therefore a natural extension of computational approaches that relate representational structure to measurable differences in perturbation behaviour across neural architectures.

The Predictive accuracy is often the primary basis for comparing neural networks, but it does not directly characterize the internal representational structure of a model. Networks with similar predictive accuracy may exhibit substantially different internal stability regimes, suggesting that performance alone is insufficient for understanding how learned representations behave. Although accuracy and input-output smoothness have been widely studied, the geometry of hidden representations remains comparatively under-characterized. In particular, it is still unclear whether the internal sensitivity of learned representations is primarily governed by Lipschitz behavior, training dynamics, architectural structure, or their interaction.

This paper provides a structural concept we refer to as opacity, which refers to how difficult it is to induce changes in output from perturbations in the hidden representations or latent variables. More generally speaking, a representation is said to be opaque if perturbations of large size must be applied in hidden space to induce a given change in output size. The concept of opacity differs from traditional smoothness in that opacity is defined with respect to hidden (internal, intermediate) layers of the network rather than input (external) layers of the network. Thus, it captures more of a geometric property of the representation that is learned, rather than the learned function as a whole. In essence, from the perspective of measuring uncertainty, opacity reflects how stable a learned representation is at an internal level and serves as a complement to the input-output level measures of robustness that currently exist.

Currently, most analyses of the stability of neural networks emphasize global sensitivity to input perturbations (robustness) or Lipschitz bounds. The previous methods that were applied only consider the network from a single mapping point of view and therefore do not have the ability to distinguish between neural network architectures which exhibit similar input behaviour but are divergent internally. Hence, input based metrics cannot solely characterise the representational geometry of a neural network, and therefore additional structural metrics are required to quantify the organisation of information in a neural network across its depth.

To formalize the concept of opacity as dimensionality of neuron representation (in Sect. 3), we combine Jacobian norms and minimum internal perturbation radii to develop a layerwise sensitivity-radius functional for quantifying the opacity of each of a network's layers. We find that opacity is not contingent upon the predictive accuracy of output, nor is it solely based on the smoothness between an input and output of a network.

In Sect. 4, we develop a method for estimating the opacity of a network using layerwise perturbation search and randomized directional Jacobian projections. Furthermore, our estimators for opacity are architecture agnostic, and therefore can be applied uniformly across multilayer perceptrons, convolutional networks and transformers, without modification.

In Sect. 5, we provide empirical results on MNIST and CIFAR-10, where we find that at least some neural networks that exhibit similar predictive accuracy exhibit different internal perturbation radii, with transformers having significantly larger stability radii than multilayer perceptron baselines, and CNNs exhibiting intermediate stability radii.

All of the above empirical results demonstrate that the analysis of opacity is an under-characterised geometrical structure of a neural network's representation that is orthogonal to a neural network's predictive accuracy and input Lipschitz behaviour. Thus, our analysis of opacity as a dimension will add value to the analysis of neural networks.

2 Background Study

The stability and structure of neural representations have been studied from several theoretical and empirical perspectives, though typically through input-output analyses rather than internal geometric characterizations. A large body of work on adversarial robustness demonstrates that small input perturbations can result in large output changes in deep networks [4,5]. Robust optimization methods formalize this phenomenon through minmax training targets [6], at the same time as certification strategies offer provable input-space guarantees below Lipschitz or randomized smoothing assumptions [7–9]. These works ordinarily recognition on worldwide input sensitivity, treating networks as enter-output mappings without at once studying the internal geometry of intermediate representations.

Related theoretical efforts connect robustness and generalization to norm-based or spectral complexity measures. Margin-based totally and spectral norm bounds try to manipulate generalization through international Lipschitz constants or layerwise operator norms [10–12]. Empirical research has examined the correlations between sensitivity metrics and generalization performance [13,14]. While those techniques quantify elements of useful smoothness, they do no longer isolate internal representational balance as a separate structural amount. In unique, global norm manage does now not always suggest uniform balance across intermediate layers.

The geometry of discovered representations has additionally been investigated via representational similarity and subspace analyses. Canonical correlationprimarily based strategies [15,16] evaluate internal representations throughout networks and training ranges. Visualization of loss landscapes and flatness houses has supplied perception into parameter-area geometry [17]. More recently, phenomena consisting of neural fall apart characterize emergent geometric shape within the terminal layers of deep classifiers [18]. These research monitor structural regularities in illustration area, but they commonly describe statistical alignment or clustering consequences rather than perturbation balance with appreciate to output alternate.

From a dynamical systems perspective, analyses of sign propagation and dynamical isometry take a look at how structure impacts gradient float and conditioning [19,20]. The neural tangent kernel framework characterizes training dynamics in overparameterized regimes [21]. Information-theoretic viewpoints interpret deep learning through compression and mutual-information trade-offs [22,23]. While these views address stability and structure in different forms, none directly formalize a layerwise minimum perturbation radius that links hidden-space perturbations to controlled output variation.

While existing robustness and generalization frameworks primarily characterize input-space sensitivity, output stability, or norm-based smoothness, the present work focuses on perturbation stability inside intermediate representation spaces. In contrast to prior robustness, generalization, and representational-similarity frameworks, opacity is designed to quantify perturbation stability directly within hidden representation spaces. Its novelty lies in treating the minimum hidden-space perturbation required for controlled output change as a layerwise geometric property, rather than as an input-output robustness measure, a norm-based complexity bound, or a statistical similarity score.

Collectively, earlier work establishes that neural networks may be analyzed by entering robustness, norm-based complexity, representational similarity, dynamical conditioning, or information compression. However, those frameworks do not provide a unified geometric useful that (i) operates explicitly in hidden space, (ii) quantifies the minimal internal perturbation required for output change, and (iii) permits pass-structure comparison impartial of predictive accuracy. The opacity practical delivered in this work addresses this hole by formalizing inner perturbation stability as a structural property of discovered representations as opposed to completely of enteroutput mappings.

Despite sizable progress in knowledge robustness, generalization, and representation geometry, present frameworks usually quantify global sensitivity, alignment, or data waft. They do not formalize the minimal internal perturbation required to set off the controlled output variant as structural assets of a community. In particular, prior work does no longer isolate a layerwise hidden-state radius that hyperlinks inner representational stability to useful behavior in a principled manner. Our formulation of opacity addresses this gap by defining a geometric functional over intermediate representations, thereby separating architectural structure from training-dependent sensitivity. This perspective

complements current robustness and generalization theories at the same time as introducing a brand new axis along which neural models may be analyzed.

3 Opacity as an Internal Geometric Quantity

This section formalizes opacity as a geometric property of internal representations. We define layerwise opacity radii, relate them to Jacobian norms under mild regularity conditions, and show that architectural structure alone can induce arbitrarily small global opacity—even when input–output sensitivity remains fixed. This section presents the main theoretical formulation and proof sketches; detailed proofs and technical derivations are provided in Appendix A.

3.1 Layerwise Opacity Radius

Let $f : \mathbb{R}^d \to \mathbb{R}^k$ be a neural network composed of L intermediate representations:

$$h_0 = x, \qquad h_{\ell+1} = g_\ell(h_\ell), \qquad f(x) = h_L. \tag{1}$$

For any layer $\ell \in \{0, \ldots, L-1\}$, define the *tail mapping*

$$F_\ell(h) := g_{L-1} \circ \cdots \circ g_\ell(h), \qquad f(x) = F_\ell(h_\ell(x)). \tag{2}$$

Fix an internal layer ℓ and an output tolerance $\eta > 0$. We define the layerwise opacity radius at x as

$$r_\ell(x;\eta) := \inf\Big\{ r > 0 \;:\; \sup_{\|\varepsilon\|_2 \le r} \|F_\ell(h_\ell(x)+\varepsilon) - F_\ell(h_\ell(x))\|_2 \;\ge\; \eta \Big\}. \tag{3}$$

This is the smallest internal perturbation magnitude that can induce an output change of size at least η.

Define the *pointwise opacity*

$$R(x;\eta) = \min_{\ell=0,\ldots,L-1} r_\ell(x;\eta), \tag{4}$$

and the *global opacity*

$$R(f;\eta) = \mathbb{E}_{x\sim\mathcal{D}}\big[R(x;\eta)\big]. \tag{5}$$

Opacity thus measures the smallest internal bottleneck through which a significant output change can be induced.

3.2 First-Order Characterization

Let

$$J_\ell(x) = \left\| \frac{\partial F_\ell}{\partial h}\big(h_\ell(x)\big) \right\|_{2\to 2}. \tag{6}$$

denote the operator norm of the Jacobian from representation h_ℓ to the output.

Lemma 1 (First-Order Opacity Approximation). *Fix $\ell \in \{0, \dots, L-1\}$ and let F_ℓ denote the tail mapping so that $f(x) = F_\ell(h_\ell(x))$. Assume F_ℓ is differentiable at $h_\ell(x)$ and its Jacobian is locally Lipschitz in a neighborhood of $h_\ell(x)$. Then for sufficiently small η,*

$$r_\ell(x;\eta) = \frac{\eta}{J_\ell(x)} + O(\eta^2), \tag{7}$$

where $J_\ell(x) = \left\| \frac{\partial F_\ell}{\partial h}(h_\ell(x)) \right\|_{2\to 2}$.

Interpretation. Opacity at a layer is inversely proportional, to first order, to the sensitivity of the output with respect to that representation. Layers with large Jacobian norm admit small opacity radii.

Remark (Smoothness Requirement). If the Jacobian $\partial f / \partial h_\ell$ is locally Lipschitz, the $O(\eta^2)$ remainder is controlled by the second-order variation of the network in that layer. No global linearity assumption is required.

3.3 Architectural Decoupling

We now show that opacity can be driven to zero independently of input–output sensitivity.

Theorem 1 (Architectural Opacity Decoupling). *There exists a piecewise-linear (ReLU) network architecture and a data distribution supported inside a single linear region such that:*

1. *the input–output Lipschitz constant is fixed, and*
2. *the global opacity $R(f;\eta)$ can be made arbitrarily small.*

Proof sketch. Inside a fixed ReLU activation region, the network reduces to a linear map:

$$f(x) = W_L W_{L-1} \cdots W_1 x. \tag{8}$$

Choose a decomposition

$$W_1 = \frac{1}{\alpha} W, \qquad W_L = \alpha I, \tag{9}$$

with all intermediate layers identity maps. Then $f(x) = Wx$, so the input–output mapping remains unchanged. However, the internal Jacobian from representation h_1 to the output satisfies

$$J_1(x) = \left\| \frac{\partial f}{\partial h_1}(x) \right\|_{2\to 2} = \|W_L\| = \alpha \|I\|. \tag{10}$$

As $\alpha \to \infty$, the internal sensitivity $J_1(x)$ becomes unbounded, and by Lemma 1,

$$r_1(x;\eta) = \frac{\eta}{J_1(x)} + O(\eta^2) \to 0. \tag{11}$$

Thus $R(f;\eta) \to 0$ while the global Lipschitz constant of f remains fixed. □

Interpretation. Opacity is not determined by the overall input–output mapping. It depends critically on internal factorization. Architecture alone can create arbitrarily thin internal bottlenecks.

3.4 Attention and Representation Diffusion

Transformer layers redistribute information across tokens via attention weights. Let a single attention block produce

$$z = A(h)Vh, \tag{12}$$

where $A(h)$ is the attention matrix and V a value projection. The Jacobian with respect to a token representation h_i decomposes as

$$\frac{\partial z}{\partial h_i} = V^\top A_{\cdot i} + \frac{\partial A}{\partial h_i} Vh. \tag{13}$$

Thus opacity depends not only on value projections but also on how attention weights vary with the representation. Architectures that distribute attention mass broadly across tokens may increase layerwise radii by diffusing local perturbations, whereas sharply concentrated attention may reduce them. Opacity therefore reflects structural information flow patterns intrinsic to the architecture.

3.5 Summary of Theoretical Implications

The preceding subsections have shown that opacity admits a local geometric characterization through layerwise perturbation radii and is inversely related, to first order, to layerwise Jacobian norms. They also show that opacity can be made arbitrarily small under architectural reparameterization without changing input-output sensitivity. Finally, the analysis of attention mechanisms shows that structured diffusion effects can modulate opacity through both value projections and attention gradients.

These results position opacity as a structural internal property of neural representations rather than a mere consequence of global Lipschitz behavior. Detailed proofs and additional technical derivations corresponding to these results are provided in Appendix A.

4 Estimation of Opacity

Section 3 introduced opacity as a layerwise sensitivity–radius functional. We now describe practical estimators for this quantity that are architecture-agnostic and scalable to modern neural networks.

Our objective is to estimate both the input sensitivity proxy and the minimal internal perturbation radius.

Both procedures rely only on automatic differentiation and forward evaluation, and apply uniformly to multilayer perceptrons, convolutional networks, and transformer architectures.

4.1 Directional Jacobian Sensitivity Estimation

Exact computation of Jacobian operator norms is computationally expensive. Instead, we approximate layerwise sensitivities using randomized directional projections.

Given a model f and input x, we approximate

$$\|J_{h_\ell \to f}(x)\|_{2\to 2}$$

by sampling random unit vectors u in output space and computing

$$\|\nabla_{h_\ell} \langle f(x), u \rangle\|_2 .$$

For sufficiently many sampled directions, the maximum observed norm provides a stable proxy for the operator norm. This procedure requires only backpropagation and scales linearly with the number of sampled directions.

The same estimator is used both for input sensitivity (when $\ell = 0$) and for layerwise sensitivity terms entering opacity.

4.2 Layerwise Radius Estimation

Opacity depends on the minimal perturbation in hidden space required to induce a fixed output change. For each layer ℓ, we estimate

$$r_\ell(x) = \inf \{\|\delta\|_2 : \|f_\ell(h_\ell(x) + \delta) - f(x)\|_2 \geq \eta\}, \tag{14}$$

where f_ℓ denotes the remainder of the network from layer ℓ onward.

We approximate this value through directional search combined with binary refinement. For each hidden layer, random unit directions v are sampled in hidden space. The perturbation magnitude is then expanded along each direction until the output change exceeds the threshold η, after which binary search is used to refine the boundary.

The minimal radius across sampled directions provides an upper bound on the true minimal perturbation radius.

This procedure requires only forward passes and scales linearly in the number of directions and refinement steps.

4.3 Opacity Estimator

For each input x, opacity is computed as the minimum layerwise radius:

$$R(x) = \min_\ell r_\ell(x).$$

In practice, we evaluate this quantity on a random subset of test data and report the mean input sensitivity L_{avg}, the maximum input sensitivity $L_{\max}$, and the mean opacity $\widehat{R}$.

The estimator is architecture-agnostic and does not rely on model-specific assumptions.

4.4 Matched-Sensitivity Separation Protocol

To empirically validate the structural decoupling established in Sect. 3.3, we compare networks under matched sensitivity conditions.

Specifically, we identify pairs of models with comparable predictive accuracy, within 1%, and comparable input sensitivity L_{avg}, within a fixed tolerance.

Within such matched pairs, differences in opacity cannot be attributed to predictive performance or global smoothness. Observed discrepancies, therefore, provide empirical evidence that opacity captures structural information not encoded in traditional metrics.

4.5 Computational Complexity

Let N denote the number of evaluation samples, L the number of layers, S the number of hidden directions, and T the number of binary refinement steps. Let K denote the number of sampled output directions for Jacobian estimation.

The overall computational cost scales as

$$\mathcal{O}\big(NL(K + S\ \mathrm{T})\big).$$

In practice, moderate values of S and T yield stable estimates, making the procedure feasible for medium-scale architectures.

5 Empirical Evidence for Architecture-Dependent Opacity

Sections 3–4 defined *opacity* as an internal property of neural representations, formalized through bottleneck geometry and operationalized via the empirical estimator $\hat{R}$. In this section, we evaluate whether opacity, so defined, exhibits systematic structure across architectures and whether such structure can be attributed primarily to architectural design rather than to training noise, regularization, or predictive performance. All experiments follow exactly the measurement pipeline introduced in Sect. 4 and are interpreted strictly through the theoretical lens developed in Sect. 3.

Table 1. Architecture-Level Comparison of Test Accuracy, Average Input Sensitivity (L_{avg}), and Estimated Bottleneck Radius ($\hat{R}$).

Architecture	Dataset	Test Acc.	L_{avg}	$\hat{R}$
MLP	MNIST	0.98	7.93	0.46
CNN	MNIST	0.99	3.21	2.43
CNN	CIFAR-10	0.69	9.05	1.70
TinyViT	CIFAR-10	0.66	2.09	4.10

5.1 Architecture-Level Separation in Opacity

Table 1 above reports test accuracy, average input sensitivity L_{avg}, and estimated bottleneck radius $\hat{R}$ for the MLP, CNN, and Transformer (TinyViT) models on MNIST and CIFAR-10.

Figure 1 below visualizes the same results by embedding models in the $(L_{\text{avg}}, \hat{R})$ plane. The results Indicate That Internal Representational Geometry Varies Across Architectures and Is Not Determined Solely by Predictive Performance or Input-Level Sensitivity.

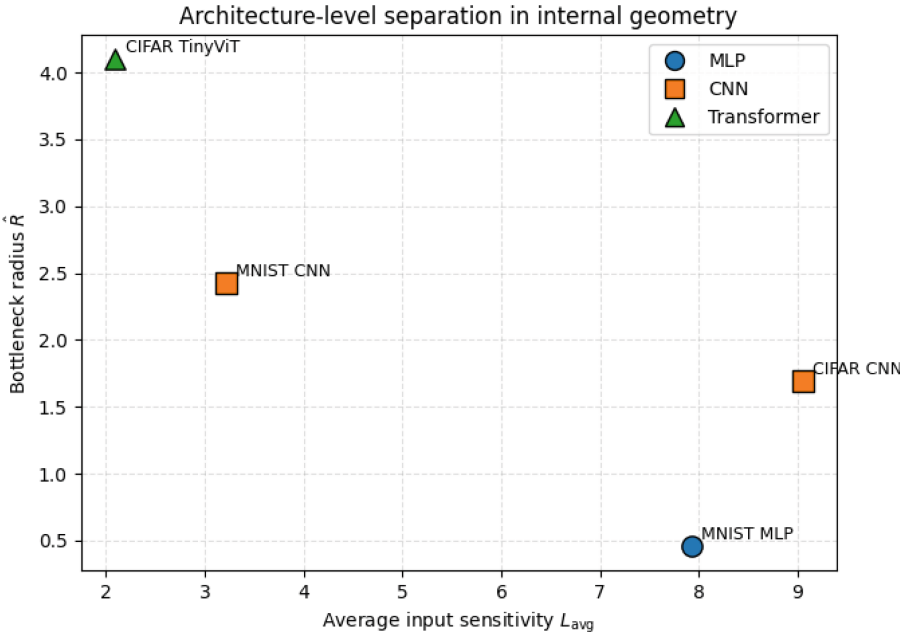


Fig. 1. Architecture-level Separation In Opacity.

Specifically, two observations are central. First, **opacity is not determined by predictive accuracy**. For example, the MNIST MLP and MNIST CNN both achieve near-saturated accuracy, yet their opacity estimates differ sharply (MLP: $\hat{R} \approx 0.46$; CNN: $\hat{R} \approx 2.43$). This directly reflects the bottleneck formulation of Sect. 3: even when end-task performance is matched, the minimum-radius layer governing opacity may occur at different depths and scales depending on architectural constraints.

Second, **opacity is not reducible to input-level sensitivity**. As shown in Fig. 1, models with low L_{avg} do not necessarily exhibit low opacity. In particular, the TinyViT model on CIFAR-10 occupies a regime characterized by simultaneously low input sensitivity ($L_{\text{avg}} \approx 2.09$) and large bottleneck radius

($\hat{R} \approx 4.10$), whereas the CIFAR-10 CNN exhibits substantially higher input sensitivity but a smaller $\hat{R}$. This separation demonstrates that opacity captures an internal property not controlled by input smoothness alone.

Taken together, Table 1 and Fig. 1 provide direct empirical support for the title claim: **opacity varies systematically with architecture** and cannot be inferred from accuracy or input sensitivity.

5.2 Layerwise Sensitivity Profiles and the Structure of Opacity

Section 3 characterizes opacity through a *min-over-layers* bottleneck mechanism. If opacity is genuinely architecture-dependent, this dependence should manifest in the layerwise internal sensitivity profiles J_ℓ.

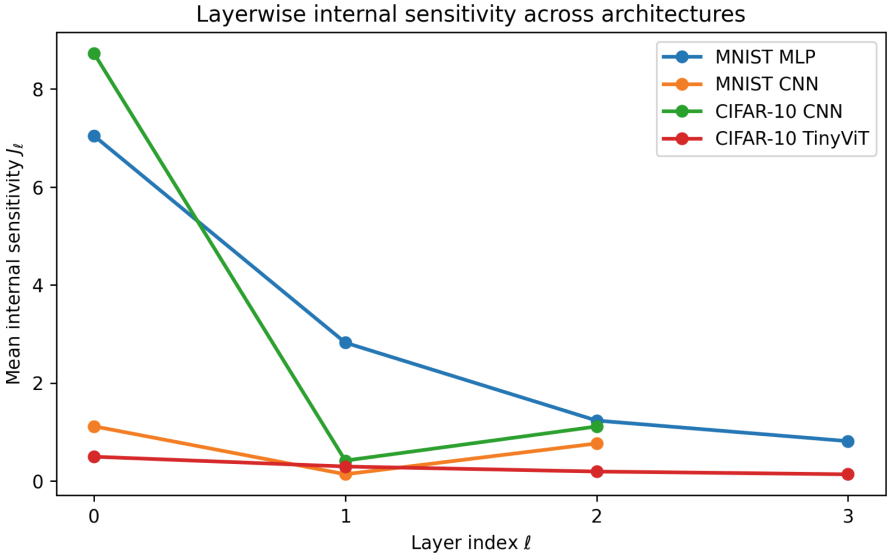


Fig. 2. Layerwise Sensitivity Profiles and the Structure of Opacity.

The MNIST MLP exhibits a high initial sensitivity followed by gradual decay, consistent with diffuse early representations and a deep bottleneck that yields a small $\hat{R}$. Above, Fig. 2 plots J_ℓ in depth for the evaluated models, calculated as described in Sect. 4. CNNs display non-monotonic profiles, characterized by sharp early compression and partial re-expansion, reflecting localized architectural bottlenecks and corresponding to larger opacity on MNIST. In contrast, the TinyViT model shows uniformly low and smoothly decreasing sensitivity across layers, consistent with distributed representation mixing and a comparatively large bottleneck radius.

These patterns provide *mechanistic grounding* for the scalar opacity estimates reported in Table 1. Differences in $\hat{R}$ arise not from noise, but from coherent depth-wise sensitivity structure induced by architecture.

5.3 Limited Tunability of Opacity Within an Architecture (Table 2)

A natural counter-hypothesis is that opacity can be freely adjusted through training-time interventions such as dropout or weight decay. To test this, Table 2 reports matched-performance configurations within a fixed architecture (MNIST MLP), selected to have similar test accuracy and closely matched L_{avg}, while varying regularization strength. Configurations are selected to have similar test accuracy ($\pm 1\%$) and comparable input sensitivity (L_{avg} within $\pm 7\%$). The estimated bottleneck radius ($\hat{R}$) varies only modestly, indicating limited tunability of opacity via regularization.

Table 2. Matched-Performance Configurations Within MNIST MLP.

Dropout	Weight Decay	Test Acc.	$\hat{R}$
0.0	10^{-3}	0.9749	0.44
0.2	10^{-4}	0.9781	0.48
0.0	10^{-3}	0.9749	0.44
0.2	10^{-3}	0.9694	0.48
0.0	10^{-4}	0.9737	0.46
0.0	10^{-3}	0.9749	0.44

Despite these variations, the resulting opacity estimates remain confined to a narrow range. Once predictive performance and input sensitivity are controlled for, changes in $\hat{R}$ are modest relative to the cross-architecture differences observed in Table 1. This indicates that opacity, as formalized in Sect. 3, is **not a fragile training artifact**, but is instead strongly constrained by architectural structure.

6 Conclusion

We redefined opacity as a formally defined geometric functional over internal neural representations and demonstrated that it constitutes a structural property distinct from classical input-based smoothness measures. Our theoretical evaluation and empirical results provide evidence that opacity captures an internal property of representation geometry that cannot be characterized solely in terms of predictive accuracy or global sensitivity. Specifically, two architectures with equivalent predictive performance may differ substantially with respect to their opacity properties; therefore, the internal geometric organization of a network is likely determined primarily by its architectural structure. The findings outlined in the paper indicate that, from a computational science perspective, a study of internal representation behavior can also be conducted using other measurable quantities in addition to external performance. Therefore, when comparing

architectures, predictive accuracy is not the only factor of concern; how internal perturbations are handled and transmitted between layers is also important.

This perspective reframes opacity as a feature of representational bottlenecks and perturbation stability rather than simply as a high-level concept that is synonymous with interpretability or robustness. By identifying a small perturbation radius for hidden states that is shown to be independent of standard Lipschitz-type metrics, we establish an architectural dimension of neural networks that has generally remained unexplored in the literature on generalization and robustness. Our findings suggest that the internal geometry of neural networks may influence the way in which perturbations move through the layers of the network and potentially, in turn, shape the nature of the uncertainty associated with prediction accuracy, regardless of calibration and probabilistic outputs.

One avenue for future research is to develop a deeper analytical understanding of opacity through the lens of compositional nonlinear operators. By investigating this avenue, it may be possible to obtain deeper insights related to the nature of opacity in extremely deep networks and/or networks that have been heavily over-parameterized. Investigating the extension of our model to probabilistic or Bayesian architectures may also provide insight into potential formal relationships between the underlying geometric constraints of neural networks and epistemically based uncertainties.

Our findings suggest that the internal geometry of neural networks may influence how perturbations propagate through network layers and may, in turn, be related to aspects of predictive uncertainty that are not fully captured by calibration or probabilistic output measures alone. Opacity does not replace existing robustness or uncertainty notions; instead, it complements them by exposing internal geometric regimes that are invisible to conventional smoothness metrics. More broadly, this view supports the computational study of neural networks as layered systems whose internal behaviour can be analysed quantitatively. We suggest that viewing opaqueness as a geometric attribute of architecture provides a new perspective on neural representations and offers an avenue for studying structural invariants in modern learning systems.

A Proofs and Technical Details for Section 3

A.1 Preliminaries and Notation

Let

$$f : \mathbb{R}^d \to \mathbb{R}^k$$

be a feedforward neural network with intermediate representations

$$h_0 = x, \qquad h_{\ell+1} = g_\ell(h_\ell), \qquad f(x) = h_L.$$

For any layer ℓ, denote the tail mapping

$$F_\ell(h) := g_{L-1} \circ \cdots \circ g_\ell(h),$$

so that

$$f(x) = F_\ell(h_\ell(x)).$$

Let

$$J_\ell(x) = \left\| \frac{\partial f}{\partial h_\ell}(x) \right\|_{2\to 2} = \left\| \frac{\partial F_\ell}{\partial h_\ell}(h_\ell(x)) \right\|_{2\to 2}.$$

All norms are Euclidean unless stated otherwise.

A.2 First-Order Approximation of the Opacity Radius (Proof of Lemma 3.1)

Recall the definition of the layerwise opacity radius:

$$r_\ell(x;\eta) = \inf \left\{ r > 0 : \sup_{\|\varepsilon\|_2 \le r} \|F_\ell(h_\ell(x)+\varepsilon) - F_\ell(h_\ell(x))\|_2 \ge \eta \right\}. \tag{15}$$

Assumptions. Assume:

1. F_ℓ is differentiable at $h_\ell(x)$,
2. its Jacobian is locally Lipschitz in a neighborhood of $h_\ell(x)$.

Step 1: Taylor Expansion. For sufficiently small perturbations ε,

$$F_\ell(h_\ell(x)+\varepsilon) = F_\ell(h_\ell(x)) + J_\ell(x)\varepsilon + R(\varepsilon),$$

where

$$\|R(\varepsilon)\|_2 \le C\|\varepsilon\|_2^2$$

for some local constant $C > 0$.

Step 2: Factorization Trick. Let the overall linear mapping be

$$f(x) = Wx.$$

Construct a depth-3 network (within a fixed linear region) by choosing

$$W_1 = \frac{1}{\alpha} W, \qquad W_2 = I, \qquad W_3 = \alpha I,$$

with $\alpha > 0$. Then

$$W_3 W_2 W_1 = (\alpha I)\, I \left(\frac{1}{\alpha} W\right) = W,$$

so the input–output function is unchanged on this linear region.

Step 3: Internal Jacobian at Layer 1. The Jacobian from representation h_1 to the output is

$$J_1 = W_3 W_2 = \alpha I,$$

and hence

$$\|J_1(x)\| = \alpha \|I\| \xrightarrow[\alpha \to \infty]{} \infty.$$

Therefore, by the first-order radius relation,

$$r_1(x;\eta) \approx \frac{\eta}{\|J_1(x)\|} = \frac{\eta}{\alpha \|I\|} \xrightarrow[\alpha \to \infty]{} 0.$$

Thus opacity vanishes while the input–output mapping remains fixed. □

A.3 Opacity Under Piecewise Linear Activations

Even outside a single linear region, the first-order analysis applies locally.

For almost every input (except on measure-zero activation boundaries), ReLU networks are differentiable. Thus Lemma 3.1 applies almost everywhere under the data distribution.

Therefore global opacity

$$R(f;\eta) = \mathbb{E}_{x \sim \mathcal{D}}[R(x;\eta)]$$

inherits the decoupling phenomenon except possibly on activation boundaries.

A.4 Attention Layer Derivative Details

Consider a single attention head:

$$z = A(h)Vh,$$

where

$$A(h) = \text{softmax}\left(\frac{Q(h)K(h)^\top}{\sqrt{d}}\right).$$

For a token representation h_i,

$$\frac{\partial z}{\partial h_i} = V^\top A_{\cdot i} + \left(\frac{\partial A}{\partial h_i}\right) Vh.$$

The first term corresponds to value projection sensitivity. The second term captures sensitivity induced through attention weight changes.

Thus the Jacobian norm reflects both:

1. mixing strength (via A),
2. sensitivity of attention scores.

This justifies the claim in Sect. 3.4 that attention can increase or decrease opacity depending on its structural diffusion properties.

References

1. Nadif, M.A., Hasan, M.D.S., Sharma, A.K.: BanglaTag: next-gen Bangladeshi number plate recognition hierarchical processing network. In: 2024 27th International Conference on Computer and Information Technology (ICCIT). IEEE (2024)
2. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
3. Sharma, A.K.: DistilBERT based malicious URL detection system. In: 2024 4th International Conference on Digital Society and Intelligent Systems (DSInS), pp. 307–312. IEEE (2024)
4. Szegedy, C., et al.: Intriguing Properties of Neural Networks. In: International Conference on Learning Representations (ICLR) (2014)
5. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations (ICLR) (2015)
6. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (ICLR) (2018)
7. Hein, M., Andriushchenko, M.: Formal guarantees on the robustness of a classifier against adversarial manipulation. In: Advances in Neural Information Processing Systems. NeurIPS (2017)
8. Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., Usunier, N.: Parseval networks: improving robustness to adversarial examples. In: International Conference on Machine Learning (ICML) (2017)
9. Cohen, J.M., Rosenfeld, E., Kolter, Z.: Certified adversarial robustness via randomized smoothing. In: International Conference on Machine Learning (ICML) (2019)
10. Bartlett, P.L., Foster, D.J., Telgarsky, M.: Spectrally-normalized margin bounds for neural networks. In: Advances in Neural Information Processing Systems. NeurIPS (2017)
11. Neyshabur, B., Bhojanapalli, S., McAllester, D., Srebro, N.: Exploring generalization in deep learning. In: Advances in Neural Information Processing Systems (NeurIPS) (2017)
12. Arora, S., Ge, R., Neyshabur, B., Zhang, Y.: Stronger generalization bounds for deep nets via a compression approach. In: International Conference on Machine Learning (ICML) (2018)
13. Novak, R., Bahri, Y., Abolafia, D.A., Pennington, J., Sohl-Dickstein, J.: Sensitivity and generalization in neural networks: an empirical study. In: International Conference on Learning Representations (ICLR) (2018)
14. Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., Bengio, S.: Fantastic generalization measures and where to find them. In: International Conference on Learning Representations (ICLR) (2020)
15. Raghu, M., Gilmer, J., Yosinski, J., Sohl-Dickstein, J.: SVCCA: singular vector canonical correlation analysis for deep learning dynamics and interpretability. In: Advances in Neural Information Processing Systems. NeurIPS (2017)
16. Morcos, A.S., Raghu, M., Bengio, S.: Insights on representational similarity in neural networks with canonical correlation. In: Advances in Neural Information Processing Systems (NeurIPS) (2018)
17. Li, H., Xu, Z., Taylor, G., Studer, C., Goldstein, T.: Visualizing the loss landscape of neural nets. In: Advances in Neural Information Processing Systems (NeurIPS) (2018)

18. Papyan, V., Han, X., Donoho, D.L.: Prevalence of neural collapse during the terminal phase of deep learning training. Proc. Natl. Acad. Sci. **117**(40), 24652–24663 (2020)
19. Saxe, A.M., McClelland, J.L., Ganguli, S.: Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In: International Conference on Learning Representations (ICLR) (2014)
20. Pennington, J., Schoenholz, S.S., Ganguli, S.: Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. In: Advances in Neural Information Processing Systems. NeurIPS (2017)
21. Jacot, A., Gabriel, F., Hongler, C.: Neural tangent kernel: convergence and generalization in neural networks. In: Advances in Neural Information Processing Systems (NeurIPS) (2018)
22. Tishby, N., Zaslavsky, N.: Deep learning and the information bottleneck principle. In: IEEE Information Theory Workshop. ITW (2015)
23. Achille, A., Soatto, S.: Information dropout: learning optimal representations through noisy computation. IEEE Trans. Pattern Anal. Mach. Intell. (2018)

Numerical Algorithm for the Biermann Effect in ICF Self-generated Magnetic Fields and Its Applications

Yang Rong, Song Peng, and Wang Qiang(✉)

Institute of Applied Physics and Computational Mathematics, Beijing, China
{yang_rong,song_peng,wang_qiang}@iapcm.ac.cn

Abstract. A numerical algorithm is conducted based on a two-dimensional cylindrical coordinate system for the Biermann-Battery effect in the self-generated magnetic field of ICF. The numerical calculation of the Biermann-Battery term is highly prone to generating large non-physical strong magnetic fields due to pressure jumps and density discontinuities. A finite volume discrete scheme based on arbitrary quadrilateral meshes is designed, and a nodal quantity interpolation method suitable for practical application problems is presented. Furthermore, a new discrete scheme for suppressing oscillations is developed. The accuracy and oscillation suppression effect of the scheme are tested on general distorted meshes. Simulations of typical model problems are carried out, and qualitatively reasonable magnetic field distributions are obtained.

Keywords: self-generated magnetic field · Biermann Effect · arbitrary quadrilateral grid

1 Introduction

The self-generated magnetic field in laser-driven inertial confinement fusion (ICF) will greatly affect the electron heat conduction, and then affect the plasma parameter distribution and implosion symmetry in the black hohraum. Meanwhile, the applied magnetic field can optimize the laser-plasma interaction and implosion physics, which may reduce the ignition energy threshold and increase the fusion gain.

The Lasnex and Hydra programs have already included complete magnetic field program packages since the 1980 s and 1990 s. When considering the magnetic field modeling in the simulation of some typical problems, the numerical results were closer to the experimental results [1–5]. Therefore, it is necessary and very important to develop numerical simulations including magnetic field effects in radiation hydrodynamics programs.

This report mainly introduces the mathematical modeling of ICF laser fusion considering the magnetic field effect, focusing on the Biermann effect, and carries out the relevant numerical algorithm research based on the two-dimensional

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 96–109, 2026.
https://doi.org/10.1007/978-3-032-30491-9_7

cylindrical geometry program, as well as the numerical discrete schemes and related technologies adopted. It mainly includes the design of two finite volume schemes based on the arbitrary quadrilateral grids, the interpolation of nodal quantities, and their application effects in suppressing magnetic field oscillations in numerical simulations. Additionally, the accuracy of the algorithms is examined by the typical test cases, and the numerical simulation results of the algorithms in typical application models are provided.

The structure of this paper is arranged as follows: The second section of this paper gives a basic description of the governing equations. The third section elaborates on the physical mechanism of the Biermann effect. The fourth section details the design of two discrete schemes. The fifth section demonstrates the robustness of the schemes through numerical examples, Finally, the conclusions are presented in the last part.

2 Numerical Algorithm of the Coupled System

Considering the case of self-generated magnetic field, the evolution equation for the magnetic field is:

$$\begin{aligned}\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{u} \times \mathbf{B}) - \nabla \times \left(\frac{c}{4\pi n_e e}(\nabla \times \mathbf{B}) \times \mathbf{B}\right) + \nabla \times \left(\frac{1}{n_e e}\nabla P_e\right) \\ -\frac{c^2}{4\pi}\nabla \times [\overleftrightarrow{\eta} \cdot (\nabla \times \mathbf{B})] + \frac{1}{4\pi e}\nabla \times [\overleftrightarrow{\beta} \cdot (\nabla \times \mathbf{B})].\end{aligned} \tag{1}$$

The first term on the right side of the equation is the magnetic field Lorentz force, also known as the magnetic freezing term; the second term is the Hall term, which represents the magnetic field generated by the macroscopic motion of the electron, the third term is the electron heat pressing term, also known as the Biermann-Battery term, the fourth term is the resistance diffusion term, and the fifth term is the Nernst term. The effect of self-generated magnetic field has a great influence on the electron and ion equations in radiation hydrodynamics equations. The electron and ion equations are as follows:

$$\begin{aligned}\frac{\partial \varepsilon_e}{\partial t} + \nabla \cdot (\varepsilon_e u) = -P_e \nabla \cdot \mathbf{u} - \nabla \cdot \left(-\overleftrightarrow{\kappa}_e \cdot \nabla(k_B T_e) - \frac{(k_B T_e)}{e}\overleftrightarrow{\beta} \cdot \mathbf{J}\right) - W_{er} - W_{ei} \\ +(\overleftrightarrow{\eta} \cdot \mathrm{J} - \frac{1}{e}\overleftrightarrow{\beta} \cdot \nabla T_e) \cdot \mathrm{J} - P_e \nabla \cdot \left(\frac{\mathrm{J}}{e n_e}\right) - \nabla \cdot \left(\frac{\varepsilon_e \mathrm{J}}{e n_e}\right).\end{aligned} \tag{2}$$

$$\frac{\partial \varepsilon_i}{\partial t} = -P_i \nabla \cdot \mathbf{u}_i - \nabla \cdot \left(-\overleftrightarrow{\kappa}_i \cdot \nabla(k_B T_i)\right) - W_{ie} + W_{iF}. \tag{3}$$

In Eqs. (1)–(3), the magnetic field transport coefficients, electron and ion heat flux coefficients are all anisotropic tensor forms, and their specific functional expressions are given by Braginskii's [6] model.

While self-generated magnetic field coupling radiation fluid dynamics equations, the whole system equation is complex. Aimed at this system, a set of applicable numerical solution algorithm is given. First, a split scheme is adopted to reduce the complexity; the whole system is divided into two parts: the fluid

and energy equations. The self-generated magnetic field mainly affects the energy equation. For the energy system, based on the speed change of the physical mechanism, the magnetic field equation and its influence on the electron equation are solved first, and then the radiation, electron and ion coupling systems are solved.

In this study, the Biermann-Battery term of magnetic field is studied. It is the cross product of the pressure and electron number density to generate the magnetic field source, and the electron number density and pressure have strong discontinuities at the material interface and near the shock wave, which will easily produce a numerical strong magnetic field and oscillation, particularly for arbitrary quadrilateral grids [7–11].

3 Biermann-Battery Effect

The Biermann-Battery effect arises from the significant mass difference between electrons and ions, attributable to microscopic charge separation in plasmas: pressure accelerates electrons far more strongly than ions, and the relative acceleration between the two components induces charge separation, which requires the electric field to satisfy an equilibrium condition.

$$-\nabla P_e - en_e \mathrm{E} = 0,$$

where n_e and P_e are the electron number density and pressure, respectively. Substituting the ideal gas equation of state into the momentum equation yields the expression for the electric field strength. Further substituting this into Maxwell's equations gives

$$\frac{\partial B}{\partial t} = \frac{ck_B}{en_e} \nabla T_e \times \nabla n_e.$$

In the induction equation, it serves as a magnetic field source term, generating a non-zero magnetic field from an initially unmagnetized state when the directions of temperature gradient and density gradient in the plasma are misaligned. For instance, in ICF laser ablation, plasma expansion occurs primarily along the target surface normal direction. Consequently, the electron density gradient mainly aligns perpendicular to the target surface. However, since electron thermal conduction at the target surface contributes more significantly to temperature than plasma adiabatic expansion, the electron temperature gradient predominantly lies parallel to the target surface. As shown in Fig. 1, irradiation non-uniformity at the focal spot will exacerbate this misalignment. The thermoelectric field generated by this misalignment between temperature and density gradients will drive currents, inducing a quasi-static self-generated magnetic field. This quasi-static magnetic field exhibits a toroidal structure, as illustrated in Fig. 2. To estimate the strength of such magnetic fields, it is assumed that $\frac{\partial B}{\partial t} \sim \frac{1}{\tau_L}$, $\frac{\nabla n_e}{n_e} \sim \frac{1}{L_n}$, $\frac{\nabla T_e}{T_e} \sim \frac{1}{L_T}$, then $B[\mathrm{MGauss}] = 10(\frac{\tau_L}{1\mathrm{ns}})(\frac{T_e}{1\mathrm{keV}})(\frac{30\mu m}{L_n})$. Self-generated toroidal magnetic fields at the mega-gauss (hundred-tesla) level can be produced in hohlraum plasmas.

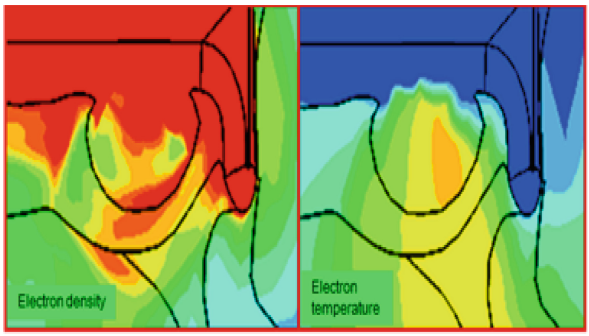


Fig. 1. Electron density and temperature distributions near the laser irradiation spot within the implosion cavity.

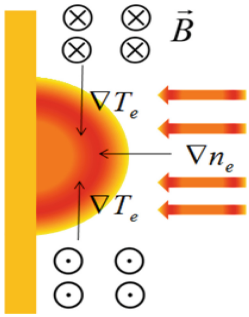


Fig. 2. Schematic diagram of toroidal magnetic field generation by thermoelectric mechanism.

In the cylindrical coordinate system, when focusing on a toroidal magnetic field, represented as $\mathbf{B} = (0, B_\theta, 0)$, and assuming $\dfrac{\partial B_\theta}{\partial \theta} \sim 0$. The equation $\nabla \cdot \mathbf{B} = 0$ is inherently fulfilled. At this point, the scalar form of the magnetic field equation in two-dimensional cylindrical geometry can be written as

$$\begin{aligned} &\frac{dB_\theta}{dt} = -\frac{c\partial}{e\partial r}\left(\frac{1}{n_e}\frac{\partial P_e}{\partial z}\right) + \frac{c\partial}{e\partial z}\left(\frac{1}{n_e}\frac{\partial P_e}{\partial r}\right) + \\ &\frac{c^2}{4\pi}\left(\frac{\partial}{\partial r}\left(\frac{\eta_\perp}{r}\frac{\partial (rB_\theta)}{\partial r} - \frac{\eta^\wedge}{r}\frac{\partial (rB_\theta)}{\partial z}\right) + \frac{\partial}{\partial z}\left(\frac{\eta_\perp}{r}\frac{\partial (rB_\theta)}{\partial z} + \frac{\eta^\wedge}{r}\frac{\partial (rB_\theta)}{\partial r}\right)\right). \end{aligned} \tag{4}$$

4 Numerical Algorithm of Biermann-Battery Term

To adapt for coupling with the radiation hydrodynamics equations, Eq. (4) are discretized using a finite volume scheme on arbitrary quadrilateral grids generated by two-dimensional fluid motions.

Assume the computational domain is a bounded two-dimensional regionΩ, partitioned into several mutually non-overlapping arbitrary structured quadrilateral grids. The full-integer indices (i, j) denote grid node coordinates, while the half-integer indices $(i + 1/2, j + 1/2)$ represent cell centroids. An arbitrary quadrilateral cell formed by nodes $(i, j), (i + 1, j), (i + 1, j + 1)$, and $(i, j + 1)$ is designated as grid cell K. The four edges of cell K are denoted as $K_l, l = 1, \cdots, 4$ (as shown in Fig. 3).

The magnetic field equation is discretized by a finite volume scheme. First, integrate Eq. (4) over the solid of revolution formed by the partitioned arbitrary quadrilateral cell in region. The first term on the right-hand side represents the magnetic resistive diffusion term, which is discretized by the classical nine-point diffusion scheme. This paper primarily focuses on the second term, the Biermann-Battery term, so that we neglect the magnetic resistive diffusion term. Thus, we

have

$$\int_{K_{i,j}} \frac{\partial \mathbf{B}}{\partial t} dV = \int_{K_{i,j}} \nabla \times (\frac{1}{n_e e} \nabla P_e) dV \tag{5}$$

The Eq. (5) is integrated over each cell K and the Gauss divergence formula is used to obtain:

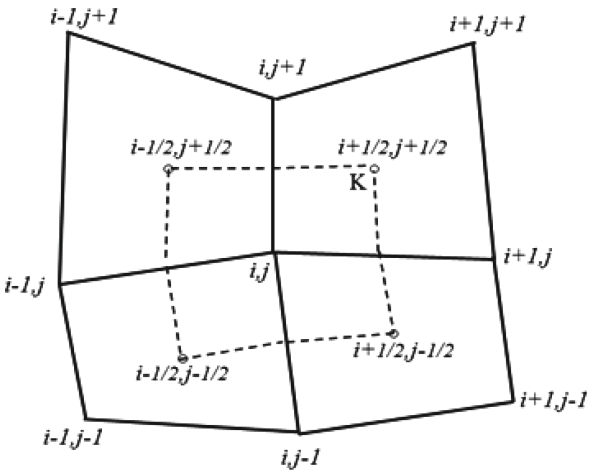


Fig. 3. Mesh Partition.

$$\begin{aligned}
&\frac{c}{e} \int_K r(-\frac{\partial}{\partial r}\left(\frac{1}{n_e}\frac{\partial P_e}{\partial z}\right) + \frac{\partial}{\partial z}\left(\frac{1}{n_e}\frac{\partial P_e}{\partial r}\right))dzdr \\
&= r_{i+1/2,j+1/2}\frac{c}{e}\oint_K \left(\frac{1}{n_e}\frac{\partial P_e}{\partial z}\right)dz + \left(\frac{1}{n_e}\frac{\partial P_e}{\partial r}\right)dr \\
&= r_{i+1/2,j+1/2}\frac{c}{e}\sum_{l=1}^{4}\left(\frac{1}{n_{e\,K^{l+1/2}}}\right)(P_{eK^{l,i+1,j}} - P_{eK^{l,ij}})
\end{aligned} \tag{6}$$

In the above formula, $r_{i+1/2,j+1/2}$represents the r-coordinate at the grid cell center. $K^{l+1/2}$denotes the edge-centered quantity of the $l-th$ edge in grid cell K. Therefore, $K^{l,ij}$and $K^{l,i+1,j}$respectively represent the starting and ending points of the $k-th$ edge along the counterclockwise direction of grid cell. This scheme requires node-centered pressure quantities and edge-centered electron number densities. Since the magnitude of the electron number density is extremely large, directly applying this formula would introduce significant numerical errors. In practical computations, we apply an identity transformation to the thermal pressure term:

$$\nabla \times (\frac{c}{n_e e}\nabla P_e) = \frac{c}{e}\nabla \times (\frac{1}{n_e}P_e \nabla \ln n_e) \tag{7}$$

Similarly, using loop integration, we have

$$\begin{aligned}
&\frac{c}{e}\int_K r(-\frac{\partial}{\partial r}\left(\frac{1}{n_e}\frac{\partial P_e}{\partial z}\right)+\frac{\partial}{\partial z}\left(\frac{1}{n_e}\frac{\partial P_e}{\partial r}\right))dzdr \\
&= r_{i+1/2,j+1/2}\frac{c}{e}\oint_K\left(\frac{1}{n_e}\frac{\partial P_e}{\partial z}\right)dz+\left(\frac{1}{n_e}\frac{\partial P_e}{\partial r}\right)dr \\
&= r_{i+1/2,j+1/2}\frac{c}{e}\sum_{l=1}^{4}\left(\frac{P_e}{n_e}\right)_{K^{l+1/2}}(\ln n_{eK^{l,i+1,j}}-\ln n_{eK^{l,ij}})
\end{aligned} \tag{8}$$

In practical simulation applications, the electron number density and pressure may exhibit strong discontinuities near interfaces due to factors such as multi-material interactions, shock waves, and equation-of-state jump discontinuities. Direct application of this formula would lead to nonphysical numerical solutions. Figure 4 simulates a laser ablation Au planar target model, which theoretically should generate negative magnetic fields as illustrated in Fig. 2. However, numerical simulations using Formula (8) produce nonphysical positive magnetic fields due to equation-of-state discontinuities.

Considering the material pressure using the ideal gas equation of state, we have:$P_e = k_B n_e T_e$, then$\frac{P_e}{n_e} = k_B T_e$. Material temperature remains continuous across discontinuities, and the Biermann-Battery term can be expressed as

$$\nabla\times(\frac{c}{n_e e}\nabla P_e)=\frac{c}{e}\nabla\times(k_B T_e\nabla\ln n_e) \tag{9}$$

The final discrete form using loop integration can be written as Eq. (10), referred to as original scheme or scheme1, it can avoid the above nonphysical numerical case.

$$\begin{aligned}
&\frac{c}{e}\int_K r(-\frac{\partial}{\partial r}\left(\frac{1}{n_e}\frac{\partial P_e}{\partial z}\right)+\frac{\partial}{\partial z}\left(\frac{1}{n_e}\frac{\partial P_e}{\partial r}\right))dzdr \\
&= r_{i+1/2,j+1/2}\frac{c}{e}\oint_K\left(\frac{1}{n_e}\frac{\partial P_e}{\partial z}\right)dz+\left(\frac{1}{n_e}\frac{\partial P_e}{\partial r}\right)dr \\
&= r_{i+1/2,j+1/2}\frac{c}{e}\sum_{l=1}^{4}(k_B T_e)_{K^{l+1/2}}(\ln n_{eK^{l,i+1,j}}-\ln n_{eK^{l,ij}})
\end{aligned} \tag{10}$$

4.1 Interpolation of Node Quantity

In the calculation of scheme (10), the discretization involves edge-centered temperature quantities, node-centered electron number densities. For any node Q_i surrounded by four cell centers $K_i(i = 1, 2, 3, 4)$, their corresponding unknown quantities are denoted as $u_0, u_i(i = 1, 2, 3, 4)$. The general interpolation of electron number density employs mass-weighted averaging of surrounding cells

$$u_0 = \sum_{i=1}^{4} w_i u_i,$$

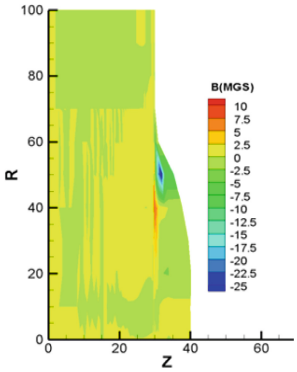


Fig. 4. Non-physical magnetic field obtained by Formula (8).

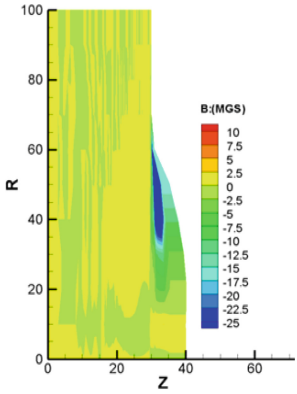


Fig. 5. Reasonable magnetic field distribution by Formula (10).

where $w_i = \dfrac{m_i}{\sum_{i=4} m_i}$. When simulating multi-material problems involving light-heavy materials, the mass-weighted averaging may introduce significant defects near interfaces: the effective density weighting of light-material cells is reduced. Near light-heavy material interfaces, this implies $n_e^{node} \approx n_e^{heavy}$. Consequently, in light-material cells$n_e^{node} \approx n_e^{heavy} \gg n_e^{light}$, and nonphysically steep gradients emerge in the low-density regions. Prolonged accumulation of these errors causes continuous growth of magnetic field strength near interfaces (Fig. 6). This ultimately leads to numerical divergence (Fig. 5).

If a simple weighting is adopted, $w_i = \dfrac{1}{4}, \quad i = 1, 2, 3, 4$. Then $n_e^{light} \sim n_e^{node} \sim n_e^{heavy}$. Compared to the mass-weighted interpolation algorithm, the gradients near light materials are substantially reduced. Figure 6 shows that an abnormally high magnetic field peak appears near the interface when the mass-weighted average is adopted (the blue part indicates the strong magnetic field at the light-heavy interface), which eventually causes the simulation to diverge. In contrast, Fig. 7 uses a simple weighted average; its magnetic field distribution conforms to physical expectations and provides practical numerical simulation results.

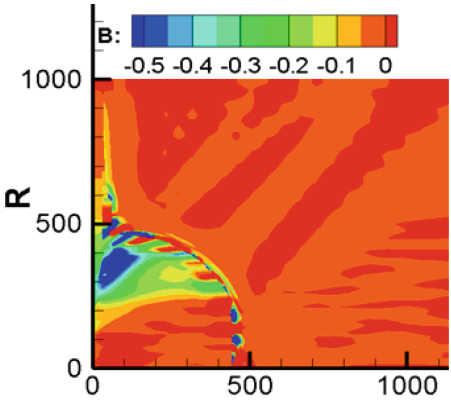


Fig. 6. Mass-weighted averaging causing non-physical magnetic fields near interfaces.

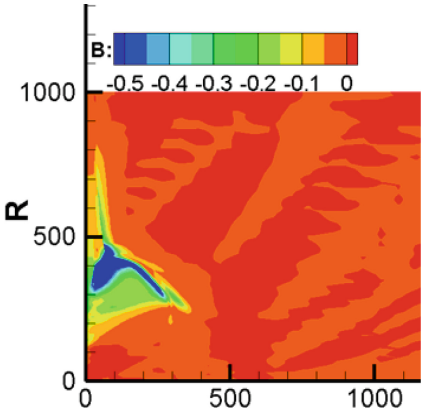


Fig. 7. Physically reasonable magnetic field distribution obtained by simple interpolation.

4.2 Least-Squares Scheme

In the coupled simulation of magnetic fields and radiation hydrodynamic equations, although the loop integral discrete scheme can yield roughly reasonable simulation results, numerical oscillations will occur in long-term Lagrangian simulations. To suppress the oscillatory behavior of the scheme, we design a novel discrete approach. We reformulate the Biermann term as follows:

$$\frac{c}{e}\nabla \times (k_B T_e \nabla \ln n_e) = \frac{ck_B}{e}\nabla T_e \times \nabla \ln n_e. \tag{11}$$

Starting from Eq. (11), we directly reconstruct gradients of electron number density and temperature on the grid. Conventional gradient reconstruction mostly adopts cell-centered and edge-centered quantities within the local grid cell. It features a simple stencil structure and low computational cost, yet fails to effectively utilize the characteristics of diagonal flow fields. In cases of slight grid distortion, large gradient variations of flow fields or discontinuity regions, gradient fitting is prone to numerical oscillation and accuracy degradation.

In this new scheme, taking the target grid cell as the center, a total of 9 cells in the surrounding 3×3 nine-grid scheme are selected to construct the reconstruction stencil. Neighboring cells are involved in gradient fitting, which greatly increases the constraint redundancy. The overdetermined system is solved by the least squares method to improve the stability of gradient reconstruction. This large-stencil structure delivers better computational performance for distorted and skewed grids.

Assuming electron temperature and electron number density follow linear distributions over adjacent nine-grid cells,

$$f(x) = a_1 x + a_2 z + a_3,$$

we approximate the distribution function via least-squares minimization by solving the following optimization problem,

$$minimize\{a_i, i = 1, 2, 3 \left| \sum_{j=1,\cdots,9} \|f_j - (a_1 x_j + a_2 z_j + a_3)\|^2 \right.\}. \tag{12}$$

where $(x_j, z_j),\ j = 1, \cdots, 9$ denotes coordinates of adjacent nine-grid cell centers, f_j and corresponds to either T_{ej}or $\ln n_{ej}$. This yields gradient distributions for electron number density and electron temperature. It should be noted that the electron number density here constructs a linear distribution of its logarithmic function. It is called Scheme 2 or the new scheme.

5 Numerical Test

The two discrete schemes proposed in the previous section were tested using an exact solution model to evaluate their performance across different grid types. The computational accuracy of the schemes was demonstrated for both orthogonal grids and general distorted grids. The algorithm was further applied to coupled simulations of realistic magnetic fields and radiation hydrodynamics equations to examine its overall performance.

Model 1. Accuracy Solution Model
Supposed that the control equation is

$$\frac{\partial B}{\partial t} = \frac{ck_B}{en_e} \nabla T_e \times \nabla n_e.$$

The initial distribution is $Te = 2 + cos(\pi y)/\pi$, $lnNe = 2 - cos(\pi x)/\pi$. Then the analysis solution is $B = -sin(\pi x)sin(\pi y)$.

The two discrete schemes proposed in the previous section were first tested using an exact solution model to evaluate their performance across different grid types. The computational accuracy of the schemes was demonstrated for both orthogonal grids and general distorted grids. The algorithm was further applied to coupled simulations of realistic magnetic fields and radiation hydrodynamics equations to examine its overall performance.

Based on the different grid types in the region $\Omega = [0, 1] \times [0, 1]$ the two schemes are tested, Fig. 8 shows the contour distribution of magnetic field intensity for the scheme1 on orthogonal grids and general distorted grids. It can be seen that the algorithm preserves the physical solution's contour distribution well on orthogonal grids. However, on general distorted grids, the numerical results exhibit significant oscillations, failing to maintain the physical solution's contour distribution. Accuracy tests across various grid resolutions reveal that the original algorithm achieves 1.5-order convergence accuracy on orthogonal grids but fails to achieve a meaningful convergence order on general distorted grids (see

Table 1. Accuracy Order of Scheme 1

Mesh	L2 Error (orthogonal grids)	L2 Order	L2 Error (general distorted)	L2 Order
20 × 20	6.537E-3	–	1.743E-2	–
40 × 40	2.167E-3	1.59	1.851E-2	–
80 × 80	7.351E-3	1.56	1.730E-2	–

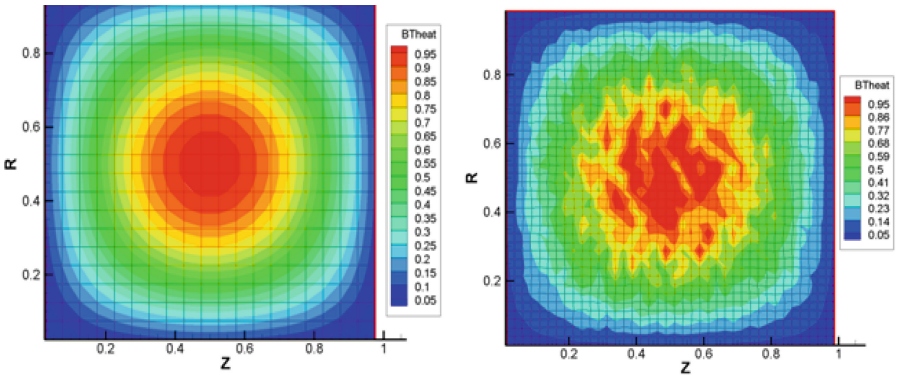


Fig. 8. Magnetic field intensity distribution on orthogonal grid (left) and randomized grid (right).

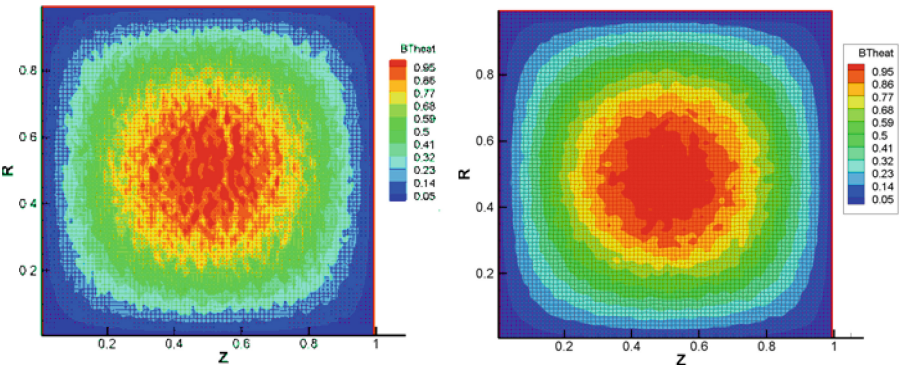


Fig. 9. Magnetic field intensity distribution on randomized grid by scheme 1 (left) and scheme 2 (right).

Table 2. Accuracy Order of Scheme 2

Mesh	L2 Error (orthogonal grids)	L2 Order	L2 Error (general distorted)	L2 Order
20 × 20	4.256E-2	–	4.375E-2	–
40 × 40	1.983E-2	1.1	2.099E-2	1.05
80 × 80	9.502E-3	1.06	1.104E-2	0.93

Table 1). This indicates that in Lagrangian computations, grid distortion easily triggers numerical oscillations in the Biermann discrete scheme.

Figure 9(left) and (right) present numerical solutions of the original and new algorithms on a 80 × 80 randomized grid respectively. The original algorithm exhibits significant oscillations, while the new algorithm effectively suppresses numerical oscillations and maintains contour distribution of magnetic field intensity. Precision tests demonstrate that the new algorithm achieves first-order convergence accuracy on both orthogonal and general randomized grids (see Table 2). Although slightly less accurate compared to orthogonal grids, this algorithm demonstrates superior precision and adaptability on Lagrangian grids.

Model 2. Laser Ablation Au Target Model
Computational domain is $[0, 30\mu m] \times [0, 100\mu m]$, grid size is 200 × 10, laser energy is $231.73J$, focal spot radius is 50 μm. The radiation transport coupled with self-generated magnetic field is used to simulate the model.

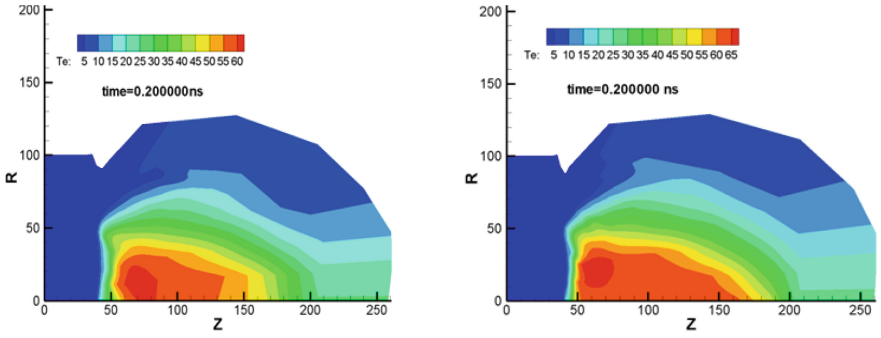


Fig. 10. Electron temperature contour by scheme 1 (left) and scheme 2 (right).

Figures 10 and 11 present the electron temperature and magnetic field intensity distributions obtained using the original algorithm and the new algorithm, respectively. The figures reveal that compared to the original algorithm, the overall distributions of electron temperature and magnetic field intensity obtained with the new algorithm show minor differences. However, the peak magnetic field values are slightly reduced, while the peak electron temperature values exhibit a modest increase.

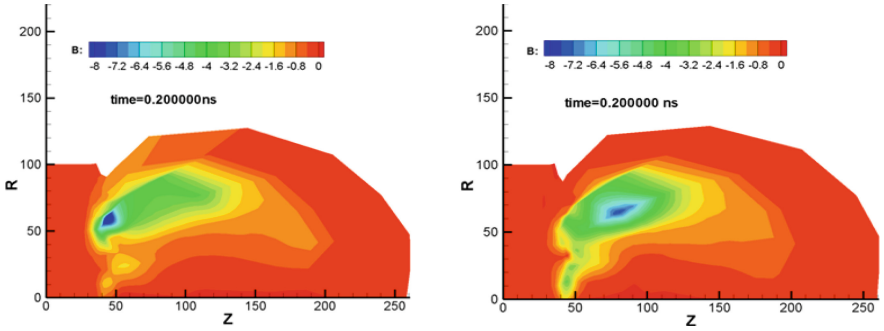


Fig. 11. Magnetic field intensity contour by scheme 1 (left) and scheme 2 (right).

Model 3. Cylindrical Hohlraum Model

The size of the cylindrical hohlraum is $1040\mu m \times 100\mu m$, with the cavity wall made of Au (initial density $19.24g/cm^3$) and a thickness of $20\mu m$. The hohlraum is filled with low-density Au gas (initial density $0.0008g/cm^3$) at an initial temperature of $300K$. The laser injection port has a diameter of $800\mu m$ and an incident angle of 45, with an energy of $2kJ$. The laser adopts a trapezoidal pulse, with a rising edge of $300ps$, a waveform duration of $1ns$, and a wavelength of $0.35\mu m$. The radiation transport coupled with self-generated magnetic field is used to simulate the model.

Figures 12, 13, 14 and 15 compare the electron temperature and magnetic field intensity distributions obtained with the original algorithm and the new algorithm respectively. The magnetic field intensity distribution demonstrates significantly reduced oscillations and lower peak values under the new algorithm. The electron temperature distribution obtained with the new algorithm shows minor differences from the original algorithm. These results indicate that the new algorithm effectively suppresses nonphysical oscillations in the magnetic

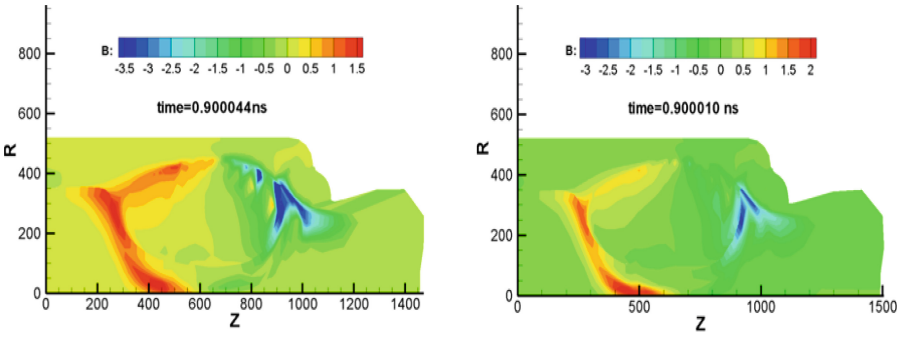


Fig. 12. Magnetic field intensity contour by scheme 1 (left) and scheme 2 (right).

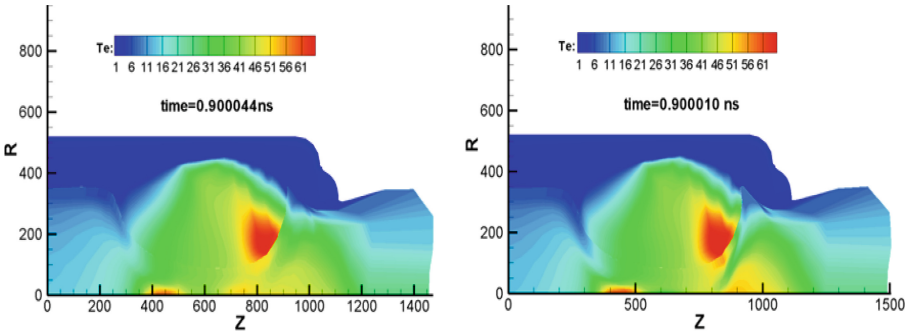


Fig. 13. Electron temperature contour by scheme 1 (left) and scheme 2 (right).

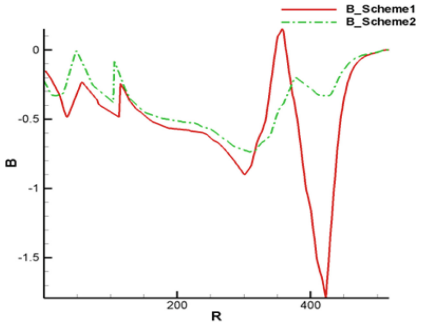


Fig. 14. Magnetic field intensity compare along z = 400.

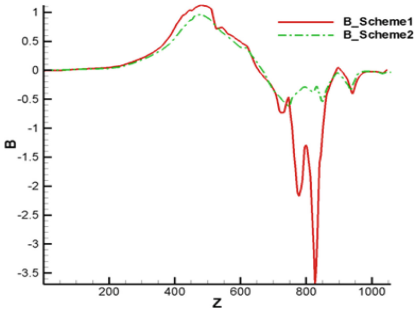


Fig. 15. Magnetic field intensity compare along r = 200.

field. Simultaneously, the new algorithm yields reduced magnetic field intensity and slightly elevated electron temperatures.

6 Conclusion

This paper presents a numerical simulation study on the Biermann-Battery term of self-generated magnetic fields. Through analyzing Biermann-Battery term in practical application simulations, two types of finite-volume discrete scheme and the corresponding nodal quantity interpolation method are proposed. The computational accuracy and oscillation suppression performance of both schemes are compared, with algorithm effectiveness verified in practical simulations.

Acknowledgments. This work has been partially supported by the National Science Foundation of China (No. 12101067.)

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Farmer, W.A., et al.: Simulation of self-generated magnetic fields in an inertial fusion Hohlraum environment. Phys. Plasmas **24**, 052703 (2017)
2. Gao, L., Nilson, P.M., Igumenshchev, I.V., et al.: Precision mapping of laser-driven magnetic fields and their evolution in high-energy-density. Phys. Rev. Lett. **114**, 215003 (2015)
3. Raphael, Plasma Physics, NNT:2016SACLX004 (2016)
4. Li, C.K.: Phys. Rev. Lett. **97**, 135003 (2006)
5. Wolff, M.: Mathematical and numerical analysis of the resistive magnetohydrodynamics system with self-generated magnetic field terms, PhD Thesis, Université de Strasbourg (2011)
6. Jia, J.-Y., Held, E.D.: Closure and transport theory for high-collisionality electron-ion plasmas. Phys. Plasmas **20**, 042114 (2013)
7. Fatenejad, M., et al.: Modeling HEDLA magnetic field generation experiments on laser facilities. High Energy Density Phys. **9**, 172–177 (2013)
8. Xu, H., O'Shea, B., et al.: The Biermann battery in cosmological MHD simulations of population III star formation. Astrophys. J. **688**, 2 (2008)
9. Graziani, C., et al.: The Biermann catastrophe in numerical magnetohydrodynamics. Astrophys. J. **802**, 1 (2015)
10. Armstrong, A.P., et al., Numerical demonstration of Biermann-generated magnetic fields in a laser-driven cylindrical implosion platform. Phys. Plasmas **32**, 072705 (2025)
11. Fryxell, B.: FLASH: an adaptive mesh hydrodynamics code for modeling astrophysical thermonuclear flashes. Astrophys. J. Suppl. Ser. **131**, 273–334 (2000)

Frequency Domain Prediction of NACA 0012 Self-noise Using Data-Driven Machine Learning

Rishav Mishra[1], Alvis Fong[2], and Bade Shrestha[1(✉)]

[1] Department of Mechanical and Aerospace Engineering, Western Michigan University, Kalamazoo, USA
bade.shrestha@wmich.edu

[2] Department of Computer Science, Western Michigan University, Kalamazoo, USA

Abstract. The paper presents the results of investigation on the feasibility of predicting the frequency domain of self-noise spectrum of a NACA 0012 airfoil employing a hybrid machine learning algorithm framework. The framework consisted of three learning architectures: CatBoost, Deep Neural Networks (DNN) and Support Vector Regression (SVR). These learning architectures were further optimized by Arithmetic Optimization Algorithm (AOA) using hyperparameter tuning. Instead of using widely available NASA Airfoil Self-Noise dataset, the models were trained and assessed employing the experimental data acquired from a low-speed anechoic wind tunnel at Western Michigan University. The experimental noise data sets of airfoil trailing edge were generated for various angle of attacks at a freestream velocity of 15 m/s. Each model was examined using baseline and a Fourier-enhanced formulation incorporating logarithmic and harmonic frequency features. The performance of each model was evaluated using standard regression metrics; root mean square error (RMSE), relative absolute error (RAE) and coefficient of determination (R^2). The results demonstrated that the Fourier-enhanced feature representations regularly predicated better results for all models tested for both broadband and tonal noise characteristics. Among the three architectures examined, CatBoost tuned with AOA achieved the best performance overall. These results support the effectiveness of using a hybrid machine learning framework with AOA based hyperparameter optimization to predict the frequency spectrums of trailing edge self-noise.

Keywords: Applied Computing · Physical Sciences and Engineering · Aerospace · Aeroacoustics · Machine learning

1 Introduction

Wind energy has become one of the widely adopted renewable and sustainable energy sources in the past few decades due to its competitiveness and reliability. As the global warming problem is exaggerating, expansions of the renewable and sustainable energy installations are critical in reducing and resolving these environmental challenges. As wind energy installations are rapidly increasing, growing numbers of communities find themselves in their proximity causing serious concern of noise pollution generated by

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 110–129, 2026.
https://doi.org/10.1007/978-3-032-30491-9_8

these wind farms. These are generally low frequency noises that cause significant annoyance and long-term health problems. While there have been a large number of research and investigations conducted, still there is a challenge in developing viable new technology to overcome this issue. The research particularly focused on mitigating the adverse effects of turbine-induced noise pollution, or airfoil self-noise pollutions emphasizing the development of advanced technologies to minimize its environmental and societal impact. Airfoil self-noise is caused by the interaction between an airfoil blade and the turbulence produced in its own boundary layer and nearby wakes [1]. The basic mechanisms of sound generation by wind turbine blades can be categorized into many types such as vortex shedding noise, turbulence-structure interaction noise, and trailing edge noise, etc. as illustrated in Fig. 1. The dominant source of the trailing edge noise is generated by turbulent boundary layer vortices scatter at the sharp edge of the airfoil [1, 2].

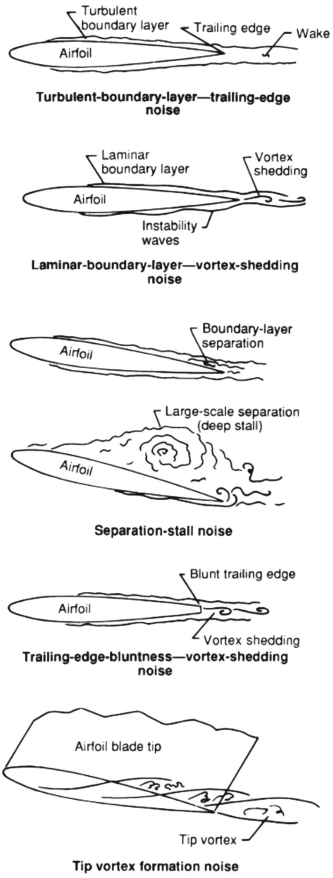


Fig. 1. Aerodynamic Noise Generation Mechanisms

The major challenge in noise reduction is a lack of models that can accurately determine the airfoil self-noise and its source terms. Since this noise is inherently connected with complex fluid-structure interactions with the solid surfaces, it depends on many factors like airfoil geometry, flow conditions, and boundary layer dynamics. The major challenge lies in modeling aerodynamic noise that accurately captures the complex, and stochastic dynamics of turbulent flows. However, simplified turbulence models offer easier implementation but limited accuracy, hindering the development of quieter airfoils.

Peterson et al. [3] conducted earlier experimental studies focusing on low Reynolds numbers and moderate angles of attack to reveal the role of boundary layer instabilities in tonal noise generation. Their experiments were conducted in an anechoic open jet wind tunnel at NASA Langley Research Center using NACA 0012 airfoils with chord lengths (C) up to 0.3 m. The airfoils were subjected to a 2D laminar inflow with varying freestream velocities (U) up to 71 m/s and angles of attack (α) up to 25°, corresponding to chord-based Reynolds numbers (Re_c) reaching 1.5×10^6. Their experiments uncovered a characteristic "*ladder-type*" acoustic signature, that corelated the dominant tonal frequency component with freestream velocity. This scaling behavior was quantified through an empirical relation for the characteristic shedding frequency f_s with kinematic viscosity of air ν, and it expressed as:

$$f_s = \frac{0.02 \times U_\infty^{\frac{3}{2}}}{\sqrt{c \times \nu}} \tag{1}$$

Along with the primary tone, several discrete and secondary tonal components were observed, each following a consistent frequency–velocity scaling proportional to $U_\infty^{0.8}$, creating rise to the ladder pattern. Subsequent experiments by Arbey and Bataille [4] and Arcondoulis et al. [5] reconfirmed the presence of this ladder pattern and provided further insight into its spectral composition. They demonstrated that the acoustic radiation consists of two principal components:(i) a broadband spectral hump associated with unstable boundary layer frequencies, and (ii) a superimposed series of discrete tonal peaks, as illustrated in Fig. 2. These observations suggested that trailing edge noise arises from a complex interaction between broadband aerodynamic processes and discrete instability driven by tonal mechanisms.

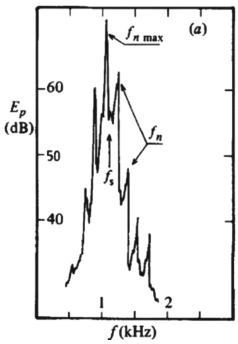


Fig. 2. Typical Noise Radiated Spectra NACA0012.8 @ U = 20.2m/s [4]

Though these experimental studies provided critical physical insight into the mechanisms underlying trailing edge noise, translating this understanding into predictive numerical models remains challenging. Advancements in computing now enable high fidelity turbulence simulations capable of resolving many of the flow features responsible for aerodynamic noise generation. However, numerical approaches still rely on experimental data for validation to ensure their accuracy. In prior studies, promising results were obtained using a similar data-driven methodology applied to NASA's NACA 0012 airfoil dataset (n0012-il) [2, 6]. The dataset comprises 1,503 samples collected under varying wind tunnel velocities and angles of attack while maintaining a fixed span and observer position. Although the dataset spans a broad range of operating conditions, the overall availability of labeled experimental data remains limited from the perspective of data-driven model development. In [2], CatBoost was combined with various metaheuristic optimization algorithms, while [6] employed a deep neural network framework to model airfoil self-noise. In the latter study, the relatively limited availability of experimental data motivated the exploration of multiple training and testing split ratios together with fivefold cross validation, in order to evaluate model robustness and reduce bias. Although better predictive performance was achieved, both studies highlighted the importance of expanding the availability of labeled experimental data to further improve generalization.

To tackle data availability constraints, alternative approaches have been explored using synthetically generated datasets. For example, in [7], a generalized airfoil database was constructed using a reference set of 31 commonly used wind turbine airfoils. A wide range of airfoil geometries were generated using airfoil functional integration theory, B spline curves, and the Class Shape Transformation method. Noise labels for this generalized airfoil set were then computed using the Brooks-Pope-Marcolini (BPM) semi empirical model, and a residual neural network (ResNet-18) was trained for noise prediction. While this approach enabled large scale data generation, the resulting labels remained dependent on semi-empirical modeling assumptions rather than direct experimental measurements.

All these studies accentuated the significance of labeled aeroacoustics data obtained under varying flow regimes for the development of reliable data-driven prediction models, especially when the objective is to capture the full noise spectra rather than isolated acoustic metrics. The current study used experimental data obtained from the low-speed wind tunnel, where the time-based raw pressure fluctuations using microphone was transformed into the frequency domain using Fast Fourier Transform (FFT). The FFT enabled decomposition of a time domain signal into its constituent frequency components, thereby revealing the spectral content of the input acoustic signal and allowing study of tonal and broadband features [8, 9]. To examine the robustness of the learning frameworks, spectral estimates were computed using varying FFT window lengths, enabling assessment of model sensitivity to spectral resolution.

The resulting full band noise spectra were used as learning targets, aiding a spectrum resolved assessment of each model's ability to reproduce the frequency domain characteristics of airfoil self-noise within the frequency range of interest. The proposed framework predicts the full distribution SPL(f), addressing a limitation of many data-driven aeroacoustics approaches that primarily report a single scalar noise metric. Since

the model learns the complete SPL(f) curve under controlled flow conditions, it captures frequency dependent behavior (including broadband trends and tonal components) that are essential for noise diagnosis and mitigation. The framework applies CatBoost tuned using AOA, while DNN and SVR tuned using the same strategy serve as benchmark models within a unified pipeline. Furthermore, rather than treating noise prediction as tabular regression, this work uses experimentally measured spectra and introduces a Fourier-enhanced frequency representation that encodes harmonic structure and multi scale spectral trends, together with angle of attack terms. Using a unified AOA based hyperparameter optimization across all models ensured consistent training and fair comparison, enabling practical frequency aware noise modeling for airfoil designs with lower noise and sustainable wind energy design.

2 Methodology

2.1 Experimental Setup

All experiments were performed in a controlled laboratory facility using a low-speed wind tunnel housed within a large (3 × 3 × 3 m) anechoic chamber. The internal walls of the chamber and all the reflective surfaces within the chamber are lined with sound absorbing foam wedges of varying shapes and sizes designed to reduce background noise contamination and limit the acoustic reflections within the chamber. The test section of the wind tunnel has a circular outlet of radius 0.115 m. Airflow enters the anechoic chamber through a circular inlet and exits via a 0.35 m diameter window on the opposite side. The airfoil model was installed at the center of the jet and mounted between two side plates that were flush with the wind tunnel outlet in order to suppress tip vortex formation and promote quasi two-dimensional flow, as shown in Fig. 3. Acoustic measurements were obtained using a $^1/_2$ " free field polarized microphone system consisting of a PCB Piezotronics Model 377A07 microphone coupled with a Model 426E01 preamplifier. The microphone was calibrated prior to testing with a Larson Davis CAL200 precision acoustic calibrator. During measurements, the microphone was located 2 chord lengths from the trailing edge of the airfoil, depending on the configuration at mid span and oriented perpendicular to the freestream direction. To reduce spurious acoustic contamination, the microphone mounting structure was treated with sound absorbing material to limit reflections from the support hardware.

The airfoil investigated in this study is a symmetric two-dimensional NACA 0012 profile. Experiments were conducted using nominal chord length, C = 0.1143 m, with a spanwise length of approximately 0.12 m. Measurements were carried out at five geometric angles of attack, $\alpha_g = 0°, 3°, 5°, 7°$ and 10°, under a fixed freestream velocity of $U_\infty = 15$ m/s. These conditions correspond to chord-based Reynolds numbers of $Re_c \approx 96{,}000$, assuming the air density $\rho = 1.169$ kg/m^3 at 28 °C within the testing facility. The finite width of the open jet test section introduces flow confinement effects that alter the effective aerodynamic angle of attack relative to the prescribed geometric value. As a result, two-dimensional correction factors were applied to account for jet contraction and blockage effects. In such configurations with limited jet height, the true angle of attack α differs from the prescribed geometric angle α_g, necessitating the

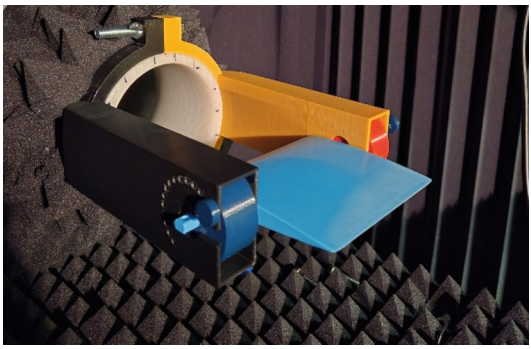


Fig. 3. Experimental Setup with NACA0012 Baseline Model

use of empirical corrections. Following the established approach outlined in [10], the relationship between α and α_g is given by:

$$\alpha = \frac{\alpha_g}{\zeta} \tag{2}$$

where:

$$\zeta = (1 + 2\sigma)^2 + \sqrt{12\sigma} \tag{3}$$

and

$$\sigma = \frac{\pi^2}{48}\left(\frac{C}{W}\right)^2 \tag{4}$$

Here C is the airfoil chord length and W = 0.115 m is the width of the circular jet within which the airfoil is placed. Acoustic data acquisition was performed using a National Instruments NI 6212 data acquisition system interfaced through the Python NI DAQ module. A custom graphical user interface developed using Node.js was employed to control the acquisition process and interface directly with the microphone system. The frontend allowed the user to specify key parameters such as sampling rate and acquisition duration. The measured voltage fluctuations over the acquisition period were converted into pressure fluctuations using the calibrated microphone sensitivity, and the resulting pressure signals were then transformed into the frequency domain using the fast Fourier transform.

2.2 CatBoost Classification Algorithm

CatBoost is a gradient boosted decision tree (GBDT) algorithm developed to improve learning performance on structured datasets that contain heterogeneous and categorical features. The algorithm was introduced by Prokhorenkova et al. with the primary objective of reducing bias caused by conventional categorical encoding strategies and improving generalization performance in supervised learning tasks [11]. A broader conceptual and empirical discussion of CatBoost within the family of gradient boosted decision trees is provided in the interdisciplinary survey by Hancock and Khoshgoftaar

[12]. A prominent distinguishing component of CatBoost is its native handling of categorical variables through ordered target statistics. Unlike traditional approaches that compute every category statistic using the full training dataset, which is prone to introducing target leakage, CatBoost computes target-based encodings using only samples that precede the current instance under a random permutation of the training data. This mechanism ensures that the encoded feature values do not contain information from the target of the current sample, thereby mitigating conditional shift and reducing overfitting [11, 12]. For a categorical feature x_i, the encoded numerical value for the k-th training instance is given by:

$$\hat{x}_i^{(k)} = \frac{\sum_{j<k} 1(x_i^{(j)}=x_i^{(k)})y_j + ap}{\sum_{j<k} 1(x_i^{(j)}=x_i^{(k)}) + a} \tag{5}$$

where $1(\cdot)$ denotes the indicator function, y_j is the target value associated with the j-th sample, p represents a prior estimate commonly chosen as the global mean of the target variable, and a is a regularization parameter introduced to stabilize the encoding for infrequently occurring categories [11, 12]. To complement ordered target statistics, CatBoost employs ordered boosting, where gradient estimates for each decision tree are computed using the same data ordering applied during categorical encoding. This strategy prevents biased gradient estimation caused by incorporating information from the current sample and improves generalization performance [11]. CatBoost further adopts symmetric decision trees, in which the same splitting rule is applied at each tree level. This structure produces balanced trees with predictable depth and inference cost, reducing overfitting and enabling efficient training and evaluation [11, 12].

CatBoost has been widely adopted in established application domains such as automation, healthcare, transportation, and education, where structured data with mixed feature types are common. Beyond these conventional use cases, its reliable performance has enabled effective deployment even in niche domains. For example, in geotechnical engineering, CatBoost has been successfully applied to slope stability prediction, where it effectively captures nonlinear relationships among geological and environmental parameters while maintaining stable predictive performance [13]. Similarly, CatBoost based hybrid models have been employed in educational data mining to assess student academic performance using datasets composed of mixed numerical and categorical attributes, achieving superior classification accuracy compared to conventional machine learning approaches [14]. Together, this evidence highlights CatBoost's suitability for structured data scenarios characterized by heterogeneity and complex feature interactions.

2.3 Deep Neural Network (DNN)

Deep Neural Networks (DNN) constitute a powerful class of data-driven models capable of learning complex nonlinear relationships present in the raw data through hierarchical feature extraction. DNNs automatically learn hierarchical representations directly from data by stacking multiple layers of nonlinear transformations, enabling effective modeling of structured data as shown in the example in Fig. 4 [15, 16].

In contrast to the other models that specifically rely on manual features, DNNs is a leading in a wide range of applications and disciplines regardless of the input data.

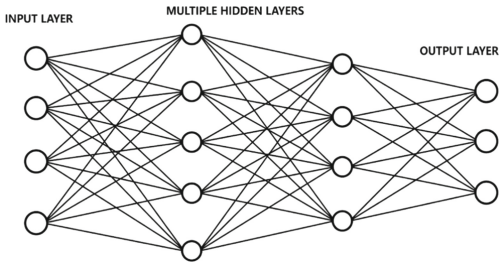


Fig. 4. Illustrative example of a DNN with input, hidden and output layers

Structurally, a standard feedforward DNNs architecture consists of an input layer, a sequence of hidden layers, and a final output layer. Within each hidden layer, the network performs an affine transformation followed by a nonlinear activation which helps the model with predictions. For a network with a depth of L layers, the forward propagation through a specific layer l is defined as:

$$a^{(l)} = g^{(l)}(W^{(l)}a^{(l-1)} + b^{(l)}) \tag{6}$$

where $\mathrm{W}^{(l)}$ and $\mathrm{b}^{(l)}$ denote the weights and biases of layer l, $g^{(l)}(\cdot)$ is a nonlinear activation function, and $\mathrm{a}^{(0)}$ is the initial input feature vector. This formulation follows the standard feedforward architecture used in DNNs, as described in [15, 16]. The training of DNNs involves optimizing the parameters involved in the network by minimizing a specific loss function through gradient-based methods. In the context of classification, the final layer typically utilizes the softmax activation function to generate class posterior probabilities. This relationship is defined as:

$$p_k = \frac{exp(z_k)}{\sum_j exp(z_j)} \tag{7}$$

where z_k represents the pre-activation logic for the k-th class. This formulation was popularized in early deep learning systems, where DNNs replaced traditional generative models for estimating posterior probabilities of latent states [17]. For regression problems, which are central to data-driven physical modeling, the learning objective is typically defined using the mean squared error (MSE):

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \|y^{(i)} - \hat{y}^{(i)}\|_2^2 \tag{8}$$

where $y^{(i)}$ and $\hat{y}^{(i)}$ denote the true and predicted outputs, respectively. The application of MSE as the standard loss metric for deep neural network regression is widely accepted in the field [15, 18]. The model is optimized by updating the parameters which are computed incorporating the gradients of the loss function via the backpropagation algorithm. This approach allows for the efficient training of deep architectures utilizing stochastic gradient descent (SGD) or its adaptive variants [15, 16] by systematically applying the chain rule across the network layers. A important aspect of DNNs is their capacity to develop layered internal representations. Early layers generally identify simple, localized patterns, and these features become progressively more abstract and specialized for

the task as they move through higher layers. This progressive feature refinement enables deep networks to model complex nonlinear patterns, forming the basis for their effectiveness in a wide range of perception tasks [16], including speech processing applications [17]. Beyond these traditional perception tasks, recent studies have demonstrated the effectiveness of deep hierarchical representations in diverse application areas such as lane detection in autonomous systems [19], flow prediction in turbomachinery and fluid dynamics [20], and complex medical image interpretation for disease diagnosis [21].

2.4 Support Vector Regression (SVR)

Support Vector Regression is a kernel based supervised learning algorithm that is rooted in the principles of statistical learning theory and derived from the Support Vector Machine Framework (SVM) methodology. In contrast to the traditional regression methods that aim to minimize the empirical loss on each data point directly, SVR formulates regression as a margin-based optimization problem. The objective is to discover a function that accurately reconstructs target values while simultaneously enforcing model complexity through regularization. The structural risk minimization principle underlying SVR confers robustness and excellent generalization properties, especially when applied to high-dimensional data or datasets with significant noise [22, 23]. Given a training set of samples $\{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, SVR constructs a regression function by minimizing a regularized objective subject to an ε-sensitive error tolerance. Following the formulation presented by [24], the SVR optimization problem is expressed as:

$$\underset{w,b,\xi_i,\xi_i^*}{min} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \tag{9}$$

subject to

$$\begin{aligned} y_i - (w^T x_i + b) &\leq \varepsilon + \xi_i, \\ (w^T x_i + b) - y_i &\leq \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* &\geq 0, \end{aligned} \tag{10}$$

In the above formulation, the regularization parameter $C > 0$ governs the critical equilibrium between the complexity of the model flatness and the penalty imposed on deviations from the training data. This optimization algorithm is the foundational concept of ε-sensitive error and functions as the baseline SVR model. These fundamental principles serve as the basis for numerous advanced regression variants that offer greater sophistication and robustness [24]. To expand SVR capabilities beyond linear regression, input data is often projected into a higher dimensional feature space using a nonlinear transformation. Within this generalized kernel-based framework, the regression function is expressed as:

$$f(x) = w^T \phi(x) + b \tag{11}$$

where $\phi(\cdot)$ represents the feature mapping generated by a kernel function, and b denotes the bias term. This explicit representation in kernel space is integral to the original

formulation of Support Vector Machines and serves as the structural foundation for kernelized SVR models [25]. By introducing Lagrange multipliers and applying the kernel trick, the SVR solution can be expressed in dual form as:

$$f(x) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) K(x_i, x) + b \tag{12}$$

where $K(x_i, x) = \phi(x_i)^T \phi(x)$ is a kernel function. In this form, only a subset of training samples contributes to the regression function, yielding sparse solutions defined by support vectors and enabling computationally efficient modeling, particularly in large scale settings. In real world implementations, improvements in model adaptability are crucial. Therefore, various SVR modifications have been proposed in published literature. Refined tolerance mechanisms built upon the classical formulation is seen in [24] to better manage noise and heterogeneous data, while retaining the fundamental SVR objective. Elsewhere, linear SVR formulations have deemed to be cost-effective with competitive predictive performance alternative in high-dimensional feature spaces [23]. Applications of SVR are diverse, and it has been utilized in domains characterized by nonlinear relationships and structured input spaces. In engineering, it functions as a surrogate model to accelerate CFD workflows, including aerodynamic design problems such as wind turbine blade twist optimization [26]. In manufacturing and process modeling, it has demonstrated strong predictive accuracy for deformation estimation and bending behavior under various operating conditions [27]. Beyond engineering applications, SVR has also been widely adopted in data-driven decision and forecasting problems. In the medical field, it is utilized to model disease progression and continuous clinical outcomes, capitalizing on its resilience to noise and efficacy with limited sample sizes [28, 29].

2.5 Arithmetic Optimization Algorithm

The Arithmetic Optimization Algorithm (AOA) is a population-based metaheuristic optimization technique inspired by the fundamental arithmetic operators: addition, subtraction, multiplication, and division [30]. In contrast to the numerous prevalent optimization methodologies that rely on metaphors from biology or physics, AOA utilizes these mathematical operations directly as search strategies. This allows the algorithm to proficiently guide candidate solutions through the search space, culminating in a structure that is conceptually straightforward yet possesses the capacity for robust and flexible performance. AOA begins by initializing a population of candidate solutions uniformly within search bounds. The j-th dimension of the i-th candidate is initialized as:

$$X_{i,j} = LB_j + r_{i,j}(UB_j - LB_j) \tag{13}$$

where LB_j and UB_j denote the lower and upper bounds, and $r_{i,j} \in (0,1)$ is a random number [30]. Mathematical Optimization Accelerator (MOA) controls the balance between exploration and exploitation defined as:

$$MOA(t) = Min + t \times \frac{Max - Min}{T} \tag{14}$$

where t is the current iteration, T is the maximum number of iterations, *Min* and *Max* are control parameters. While, Mathematical Optimization Probability (MOP) governs the step size of population updates:

$$MOP(t) = 1 - (\frac{t}{T})^{\frac{1}{\alpha}} \tag{15}$$

with α controlling the rate of decay [30]. During the exploration phase, multiplication and division operators are used to generate large search steps:

$$X_{i,j}^{t+1} = Best_j \times \frac{r}{MOP} \text{ or } X_{i,j}^{t+1} = \frac{Best_j}{r \times MOP} \tag{16}$$

where $Best_j$ is the best solution so far. In the exploitation phase, addition and subtraction operators refine solutions locally:

$$X_{i,j}^{t+1} = Best_j \pm r \times MOP \tag{17}$$

The transition between these phases and the selection of arithmetic operators are determined probabilistically using the MOA value, allowing AOA to adapt its search behavior over iterations [30]. While the standard AOA delivers good results, it may suffer from slow convergence and a tendency to become trapped in local optima in complex search spaces [31]. To mitigate such issues, Fang et al. reported that utilizing AOA to optimize SVM parameters leads to superior convergence characteristics [32], while incorporation of multi strategy fusion techniques such as opposition based learning and chaotic initialization strengthened robustness and convergence stability [33]. Recently, hybrid learning frameworks that combine AOA with advanced models like CatBoost have shown higher predictive accuracy for aeroacoustic applications validating the effectiveness of AOA as a potent optimization strategy for data-driven noise prediction [2]. This specific paradigm directly motivates the approach adopted in the present work.

2.6 Proposed Hybrid AOA Optimization for CatBoost, DNN and SVR

While Arithmetic Optimization Algorithm (AOA) has previously been combined with CatBoost for airfoil self-noise prediction, most notably in [2], the present work extends this paradigm in both methodological scope and data representation. Prior implementations utilized AOA within a singular learning architecture, often constrained by limited feature sets. In contrast, the framework proposed here introduces a generalized hybrid AOA strategy capable of simultaneously optimizing CatBoost, DNN and SVR. This mutli model approach enables a systematic and rigorous comparison across fundamentally distinct algorithmic classes. Furthermore, accurately predicting airfoil self-noise requires a deep understanding of how different flow regimes affect the acoustic signature of the airfoil, specifically distinguishing between broadband turbulence and narrowband tonal instabilities [5, 34]. Consequently, the proposed approach is specifically tailored for frequency domain datasets, where the objective is not merely a scalar SPL prediction but rather the reconstruction of the entire Sound Pressure Level (SPL) spectrum. This requirement necessitates specialized feature engineering and evaluation metrics designed to preserve both broadband characteristics and critical narrowband tonal peaks.

Therefore, a feature has been embedded into the proposed model, which is critical for meaningful aeroacoustics modeling, as given below.

Proposed Algorithm: Hybrid AOA Optimization for CatBoost, DNN and SVR

1 Input:

Train file $\mathbf{D_{tr}} = \{(f_i, \alpha_i, y_i)\}_{i=1}^{N_{tr}}$, test file $\mathbf{D_{te}} = \{(f_i, \alpha_i, y_i)\}_{i=1}^{N_{te}}$ max AOA iterations T, population size N_p.

2 Preprocess:

Convert (f, α, y) to numeric, remove non-finite rows, keep $y \geq 0$, clip $f \geq 10^{-9}$.

3 Feature Engineering (two variants):

3A Baseline:

$$X^{(B)} = [log\ (1+f), \alpha, \alpha f, (\alpha f)^2]$$

3B Fourier-enhanced: Let $x = \log_{10}(f)$. Construct

$X^{(F)} = [log\ (1+f), x, x^2, \ldots\ldots, x^P, sin(2\pi kx), cos(2\pi kx), \alpha, \alpha^2, \alpha x, \alpha x^2, (f)]$

For $k = 1, \ldots., K$, with optional raw f.

4 Standardize:

Fit StandardScaler on X_{tr}, transform X_{tr}, X_{te}.

5 AOA Setup (model specific search vectors):

Initialize a population of candidates x_i within bounds:

CatBoost: $\{\text{depth}, \eta, \lambda, T_{bag}\}$, **DNN:** $\{\text{depth}, \log_{10}(\eta), \log_{10}\lambda\}$,

SVR: $\{\log_{10}(C), \log_{10}(\gamma), \epsilon\}$

6 While $t < T$ **do:**

a. Compute MOA and MOP to balance exploration and exploitation.

b. Update candidate solutions using arithmetic operators

c. Evaluate fitness using validation RMSE with *fidelity-controlled subsampling*.

d. Update the global best solution.

e. Set $t = t + 1$

7 End While

8 Final Training (using best AOA hyperparameters):

Retrain each model on the full standardized training set using the optimal AOA-selected hyperparameters (with early stopping behavior), then predict on X_{te}.

9 Output:

Optimized models for both $X^{(B)}$ **and** $X^{(F)}$, frequency domain SPL predictions and RMSE based evaluation on test set.

The developed framework effectively bridges the gap between physics informed frequency domain feature with model specific learning architectures, while employing AOA as a unified, model agnostic optimizer for hyperparameter selection. To ensure proper assessment of this hybrid methodology, a comprehensive strategy is devised:

i. controlled dataset partitioning based on repeated wind tunnel experiments,
ii. systematic comparison of model performance before and after the introduction of Fourier-enhanced frequency features, and
iii. quantitative evaluation using established statistical error indices computed over the full frequency spectrum.

At each AOA iteration, candidate hyperparameter sets are refined using the arithmetic operators governed by exploration-exploitation control parameters. Model performance is assessed based on validation error metrics computed on predicted frequency domain SPL values. To reduce computational load, a fidelity controlled subsampling approach is implemented during optimization. Initial iterations evaluate candidates on reduced data subsets, while subsequent iterations gradually enhance evaluation fidelity. This

ensures efficient exploration of hyperparameter search space while preserving stability and accuracy near convergence (also refer to Appendix Fig. 9).

2.7 Dataset Description and Preprocessing

The datasets used in this study were obtained from the experiments conducted in a low-speed anechoic wind tunnel facility, where acoustic pressure signals were measured for symmetric NACA 0012 airfoils under controlled operating conditions. To evaluate the model's generalization capability across geometric configurations, airfoil model with chord length of C = 0.1143 m was subjected to a constant free-stream velocity of U_∞ = 15 m/s, sweeping through a range of geometric angles of attack $\alpha_g = 0°, 3°, 5°, 7°$ and 10°. For each test condition, a free-field pre-polarized microphone system was positioned at the midspan location of the airfoil and at a distance of 2C above the trailing edge to isolate the radiated noise while minimizing near-field hydrodynamic interference [35]. The microphone was shielded using a standard aerodynamic windscreen to minimize the influence of flow-induced pseudo sound, which contains of hydrodynamic pressure fluctuations caused by local turbulence [36]. To ensure sufficient temporal and statistical convergence, data acquisition was performed at a sampling rate of 20kHz with a recording duration of 30 s per run. The raw microphone pressure time histories were transformed into the frequency domain using FFT to obtain the power spectral density (PSD) of the acoustic pressure fluctuations. Following the standard aeroacoustics signal processing practice, the Sound pressure level (SPL) was computed from PSD using:

$$\text{SPL} = 10log_{10}(\frac{P_{xx}}{p_{\text{ref}}^2}) \tag{18}$$

where P_{xx} denotes the mean squared pressure density and $p_{\text{ref}} = 20\mu Pa$ is the reference acoustic pressure in air [37]. The resulting SPL spectra constitute the primary modeling target. The input vector utilized within the learning frameworks comprises the frequency domain SPL spectra, geometric angles of attack, free-stream velocity and airfoil chord length. To mitigate data leakage and ensure an unbiased evaluation, the dataset was partitioned into equal training and testing subsets based on two distinct, sequential experimental runs performed under identical nominal flow conditions as shown in Fig. 5. This partitioning strategy maintains closely matched experimental conditions while allowing model evaluation under the quasi-transient nature of the flow, thereby assessing predictive performance on unseen realizations that reflect inherent flow variability.

2.8 Effect of Frequency Representation on Model Performance

To study the impact of the proposed frequency domain feature engineering across all the models, each model is assessed using two distinct configurations that vary only in their representation of the frequency input. In the baseline configuration, frequency is provided to the models as a compact, low order representation based on $log(1+f)$ and simple interaction terms with angle of attack, i.e., $[log(1+f), \alpha, \alpha f, (\alpha f)^2]$. In doing so, frequency is a simple monotonic scaling and low order couplings, which forces the learning algorithms to derive the spectral structure, periodicity, and harmonic content implicitly from

the limited frequency parametrization. Conversely, the Fourier-enhanced configuration substitutes the low order representation with a structured basis built on $x = log_{10}(f)$, polynomial expansions x, harmonic sine/cosine components $sin(2\pi kx) and cos(2\pi kx)$, and α-interaction terms (eg., $\alpha, \alpha^2, \alpha x, \alpha x^2$), with the optional inclusion of raw f. By explicitly encoding spectral periodicity and multi-scale trends in the feature space, the burden on the models to discover oscillatory and broadband characteristics purely from data is reduced, yielding an efficient and physically meaningful representation of airfoil self-noise behavior. Similar hybrid strategies utilizing Fourier based spectral representations have been shown to significantly enhance feature extraction and classification accuracy in other acoustic monitoring applications, such as industrial pipeline leak detection [38].

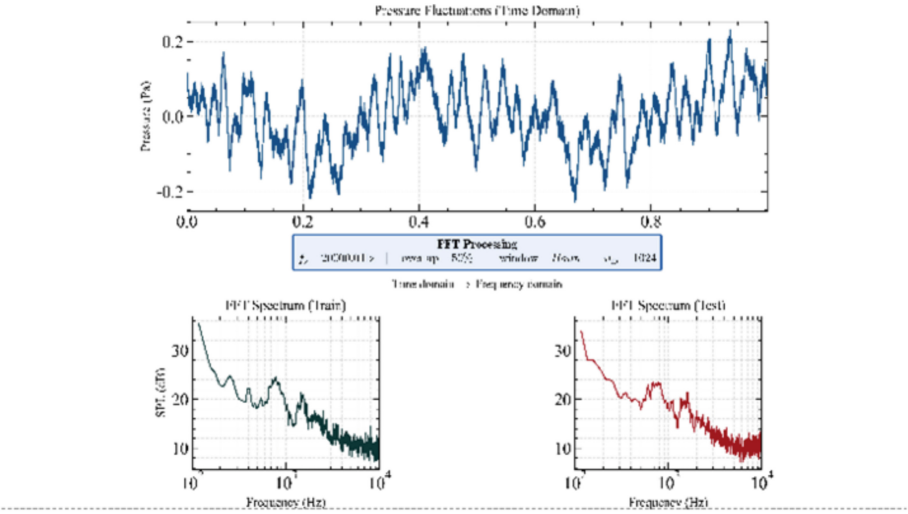


Fig. 5. Example Frequency domain transformation and resulting SPL spectra split

3 Results

The predictive performance of the hybrid CatBoost–AOA, DNN–AOA, and SVR–AOA frameworks was evaluated using both baseline and Fourier-enhanced hybrid models. The quantitative metrics were summarized in Tables 1 and 2, together with the parity plots and the SPL spectra comparisons shown in Figs. 6 and 7. It provided insight into how chosen frequency representation influences each model's ability to reproduce the broadband and tonal noise characteristics. In the baseline configuration, frequency was represented using low order transformations and interaction terms, which emphasized smooth spectral variation rather than explicit harmonic structure. CatBoost-AOA captured the broadband and dominant tonal peaks with reasonable accuracy; although the tonal peaks were slightly smoothed, their locations and relative magnitudes were largely preserved, showing that the model can exploit structured feature interactions even without harmonic encoding. DNN-AOA reproduced the broadband structure and follows general trend of the tonal peaks, but failed to resolve them as sharp narrowband peaks,

particularly at higher frequencies. In contrast, SVR-AOA only captured the broadband behavior, while the tonal components were strongly attenuated or absent in the predicted spectra. Overall, these results suggest that low order frequency features support learning of broadband trends, whereas accurate reconstruction of narrowband tonal peaks depends strongly on the model architecture, with CatBoost showing the highest inherent capability and DNN and SVR exhibiting progressively reduced narrowband spectral fidelity.

Table 1. Quantitative Performance of baseline hybrid models without explicit Fourier-enhanced features.

	Train				Test			
Model	**RMSE**	**MAE**	**RAE**	R^2	**RMSE**	**MAE**	**RAE**	R^2
CatBoost-AOA	0.5350	0.3572	0.0662	0.99519	0.5523	0.3744	0.0696	0.99486
DNN-AOA	0.7571	0.5044	0.0935	0.99037	0.7572	0.5030	0.0935	0.99034
SVR-AOA	1.1575	0.7028	0.1303	0.97749	1.1544	0.6982	0.1298	0.97755

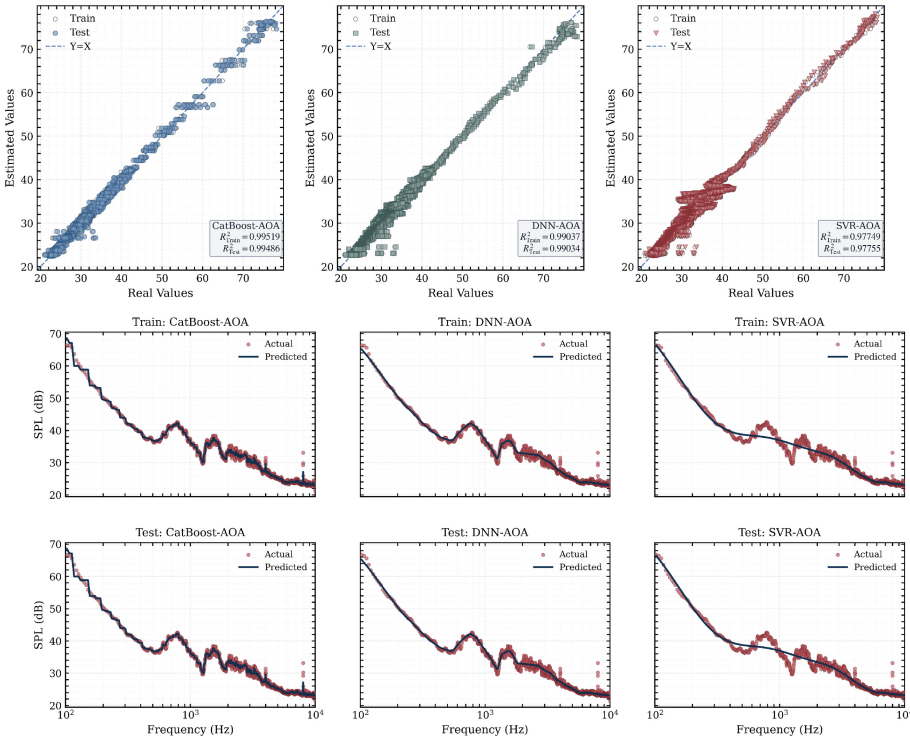


Fig. 6. Parity plots of baseline hybrid models without explicit Fourier-enhanced features (Top) and comparison of measured and predicted SPL spectra obtained from the same models (Bottom)

Table 2. Quantitative Performance of baseline hybrid models with Fourier-enhanced features.

	Train				Test			
Model	**RMSE**	**MAE**	**RAE**	R^2	**RMSE**	**MAE**	**RAE**	R^2
CatBoost-AOA (Fourier)	0.3079	0.2399	0.0445	0.99841	0.3453	0.2688	0.0500	0.99799
DNN-AOA (Fourier)	0.9158	0.4684	0.0869	0.98591	0.9189	0.4787	0.0890	0.98577
SVR-AOA (Fourier)	0.5320	0.3252	0.0603	0.99525	0.5514	0.3456	0.0642	0.99488

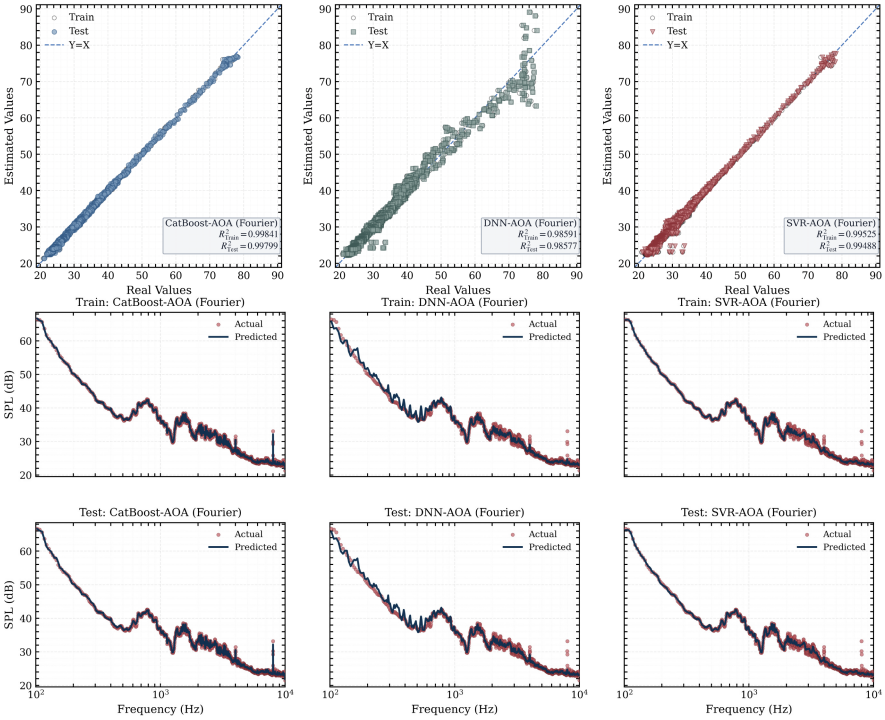


Fig. 7. Parity plots of baseline hybrid models with Fourier-enhanced features (Top) and comparison of measured and predicted SPL spectra obtained from the same models (Bottom)

With the introduction of Fourier-enhanced frequency features, the models exhibited different responses depending on the architecture. Logarithmic frequency scaling combined with harmonic sine and cosine terms provided a structured representation of frequency variation that improved the representation of tonal content relative to the baseline formulation. In the cases of the CatBoost-AOA and SVR-AOA, this yielded better agreement with measured tonal peaks, while preserving broadband trends, reflected by lower RMSE, MAE, and RAE and higher R^2 values. Although the DNN-AOA model with Fourier features better encoded the locations of tonal peaks, it introduced localized low frequency oscillations that manifest as nonphysical spectral artifacts. These

oscillations increased overall prediction error, leading to higher RMSE and lower R^2 compared to the baseline DNN formulation despite improved tonal peak visibility. This behavior suggested increased sensitivity of the DNNs architecture to harmonic feature interactions, which degraded global error metrics even when it was able to capture the local spectral features.

Despite the SVR-AOA model produced smoother predictions without such oscillations, its ability to resolve sharp tonal peaks diminished at higher frequencies, indicating limited capacity to model abrupt spectral transitions within the chosen kernel space. Overall, the Fourier-enhanced CatBoost-AOA model provided the most accurate and stable performance, consistently capturing both broadband trends and discrete tonal peaks across the full frequency range, which were supported by strong parity agreement and the lowest error metrics. These results indicated that Fourier based frequency encoding improved tonal representation, and its effectiveness depended strongly on model architecture.

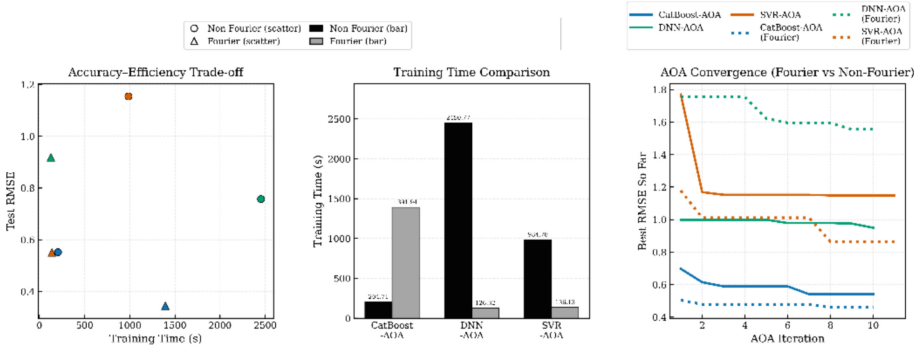


Fig. 8. Accuracy-Efficiency trade off (Left), Training Time Comparison (Middle) and Convergence behavior of the AOA for different learning models (Right)

Furthermore, Fig. 8 collectively evaluated the efficiency, accuracy-efficiency trade off, and convergence behavior of the proposed models. Fourier-enhanced configuration substantially reduced training time for DNN-AOA and SVR-AOA, while increasing computational cost for CatBoost-AOA due to higher complexity of tree-based optimization. In terms of accuracy, CatBoost-AOA (Fourier) achieved the lowest average RMSE, whereas SVR-AOA (Fourier) provided balanced trade-off between accuracy and efficiency. The convergence analysis showed that Fourier features improved the AOA search for CatBoost-AOA and SVR-AOA, while the DNN-AOA exhibited less consistent convergence, indicating higher sensitivity to feature encoding and optimization settings.

4 Conclusion and Future Work

With the integration of Fourier based input, the CatBoost-AOA model proved to be the most robust hybrid model delivering precise and physically consistent predictions, successfully capturing both broadband noise and discrete tonal peaks associated with trailing edge instabilities. Although SVR-AOA offered a competitive compromise between

accuracy and computational efficiency, the CatBoost-AOA demonstrated superior ability to resolve the narrowband spectral details vital to aeroacoustics research. In contrast DNN-AOA exhibited less consistent convergence across runs and produced localized low frequency oscillations that appeared as nonphysical tonal peaks, suggesting that the network was more sensitive to harmonic feature interactions and might require additional regularization or architecture tuning to ensure stable spectral behavior.

To improve resolution and accuracy of the models, more experimental data will be incorporated with wide ranges of stream velocities, chord lengths and clusters of microphones in future research. It will provide higher FFT resolution, yielding finer frequency spacing, that will improve repeatability of weak tonal peak identification and background noise separation. Expanding the Fourier feature with higher order harmonic terms can provide a better frequency representation and capture the multiscale oscillation behaviors. All these expansions are expected to increase the robustness and the generalization of the models to reliably and meaningfully predict the spectrum of airfoil trailing edge noise.

Appendix

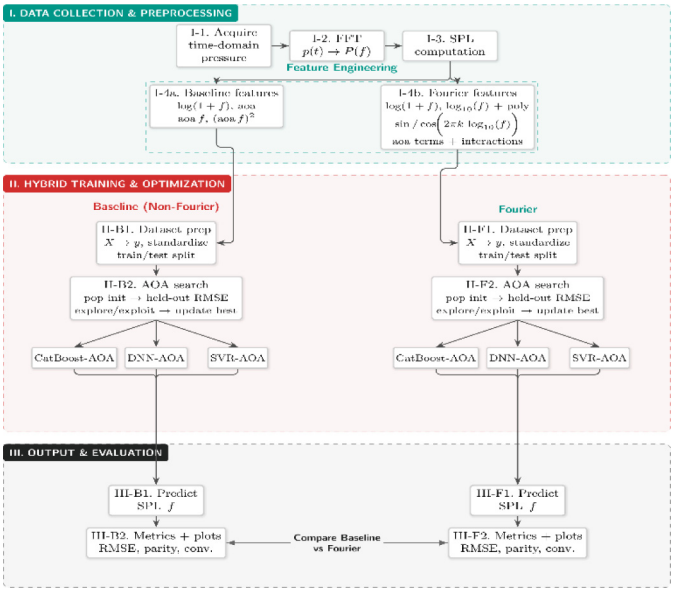


Fig. 9. Overview of the Hybrid AOA-Optimized Machine Learning Framework

References

1. Brooks, T.F., Stuart Pope, D., Marcolini, M.A.: Airfoil self-noise and prediction. Technical Report No. L-16528. NASA, Washington, DC, USA (1989)

2. Rastgoo, A., Khajavi, H.: A novel study on forecasting the airfoil self-noise, using a hybrid model based on the combination of CatBoost and arithmetic optimization algorithm. Expert Syst. Appl. **229**(2023), 120576 (2023)
3. Paterson, R.W., Vogt, P.G., Fink, M.R., Lee Munch, C.: Vortex noise of isolated airfoils. J. Aircraft **10**(5), 296–302 (1973)
4. Arbey, H., Bataille, J.: Noise generated by airfoil profiles placed in a uniform laminar flow. J. Fluid Mech. **134**(1983), 33–47 (1983)
5. Arcondoulis, E., Doolan, C., Zander, A.C.: Airfoil noise measurements at various angles of attack and low Reynolds number. In: Proceedings of ACOUSTICS. Australian Acoustical Society, pp. 23–25 (2009)
6. Redonnet, S., Bose, T., Seth, A., Li, L.K.B.: Airfoil self-noise prediction using deep neural networks. Eng. Anal. Boundary Elem. **159**(2024), 180–191 (2024)
7. Wang, Q., Zhang, H., Wang, X., Ni, Y.: Wind turbine airfoil noise prediction method based on generalized airfoil database and residual neural network. Appl. Sci. **15**(9), 5123 (2025)
8. Cooley, J.W., Tukey, J.W.: An algorithm for the machine calculation of complex fourier series. Math. Comput. **19**(90), 297–301 (1965)
9. Oppenheim, A.V.: Discrete-Time Signal Processing. Pearson Education India (1999)
10. Arce León, C., Ragni, D., Pröbsting, S., Scarano, F., Madsen, J.: Flow topology and acoustic emissions of trailing edge serrations at incidence. Exp. Fluids **57**(5), 91 (2016)
11. Prokhorenkova, L., Gusev, G., Vorobev, A., Veronika Dorogush, A., Gulin, A.: CatBoost: unbiased boosting with categorical features. In: Advances in Neural Information Processing Systems (NeurIPS 2018), vol. 31 (2018)
12. Hancock, J.T., Khoshgoftaar, T.M.: CatBoost for big data: an interdisciplinary review. J. Big Data **7**(1), 94 (2020)
13. Cai, Y., Yuan, Y., Zhou, A.: Predictive slope stability early warning model based on CatBoost. Sci. Rep. **14**(1), 25727 (2024)
14. Hao, D., Yang, X., Qi, T.: Hybrid machine learning models based on CatBoost classifier for assessing students' academic performance. Int. J. Adv. Comput. Sci. Appl. **15**, 7 (2024)
15. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge, MA, USA (2016)
16. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(2015), 436–444 (2015)
17. Hinton, G., et al.: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process. Mag. **29**(6), 82–97 (2012)
18. Das, L., Sivaram, A., Venkatasubramanian, V.: Hidden representations in deep neural networks: part 2 regression problems. Comput. Chem. Eng. **139**(2020), 106895 (2020)
19. Kim, J., Kim, J., Jang, G.-J., Lee, M.: Fast learning method for convolutional neural networks using extreme learning machine and its application to lane detection. Neural Netw. **87**(2017), 109–121 (2017)
20. Gao, G., Wang, G.: A deep neural network with attention mechanism for flow prediction of compressor blade. Sci. Rep. **15**(1), 16336 (2025)
21. Anderson, P.G., et al.: Deep learning improves physician accuracy in the comprehensive detection of abnormalities on chest X-rays. Sci. Rep. **14**(1), 25151 (2024)
22. Sifaou, H., Kammoun, A., Alouini, M.-S.: A precise performance analysis of support vector regression. In: Proceedings of the International Conference on Machine Learning (ICML 2021), pp. 9671–9680. PMLR (2021)
23. Ho, C.-H., Lin, C.-J.: Large-scale linear support vector regression. J. Mach. Learn. Res. **13**(1), 3323–3348 (2012)
24. Wei, J., He, X.: Support vector regression model with variant tolerance. Measur. Control **56**(9–10), 1705–1719 (2023)
25. Hsu, C.-W., Lin, C.-J.: A comparison of methods for multiclass support vector machines. IEEE Trans. Neural Netw. **13**(2), 415–425 (2002)

26. Kaya, M.: A CFD based application of support vector regression to determine the optimum smooth twist for wind turbine blades. Sustainability **11**(16), 4502 (2019)
27. Safari, M., Hossein Rabiee, A., Joudaki, J.: Developing a support vector regression (SVR) model for prediction of main and lateral bending angles in laser tube bending process. Materials **16**(8), 3251 (2023)
28. Du, W., Cheung, H., Johnson, C.A., Goldberg, I., Thambisetty, M., Becker, K.: A longitudinal support vector regression for prediction of ALS score. In: Proceedings of the 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2015), pp. 1586–1590. IEEE (2015)
29. Akay, Ö., Tunçeli, M.: Use of the support vector regression in medical data analysis. Exp. Appl. Med. Sci. **2**(4), 242–256 (2021)
30. Abualigah, L., Diabat, A., Mirjalili, S., Elaziz, M.A., Gandomi, A.H.: The arithmetic optimization algorithm. Comput. Methods Appl. Mech. Eng. **376**(2021), 113609 (2021)
31. Dhal, K.G., Sasmal, B., Das, A., Ray, S., Rai, R.: A comprehensive survey on arithmetic optimization algorithm. Arch. Comput. Methods Eng. **30**(5), 3379–3404 (2023)
32. Fang, H., Fu, X., Zeng, Z., Zhong, K., Liu, S.: An improved arithmetic optimization algorithm and its application to determine the parameters of support vector machine. Mathematics **10**(16), 2875 (2022)
33. Feng, C., Cai, M., Luo, J., Li, B.: Arithmetic optimization algorithm improved by multi-strategy fusion. In: Proceedings of the 4th International Conference on Artificial Intelligence, Big Data and Algorithms (AIBDA 2024), pp. 896–901 (2024)
34. Nash, E.C., Lowson, M.V., McAlpine, A.: Boundary-layer instability noise on aerofoils. J. Fluid Mech. **382**(1999), 27–61 (1999)
35. Zajamsek, B., Doolan, C.J., Moreau, D.J., Fischer, J., Prime, Z.: Experimental investigation of trailing edge noise from stationary and rotating airfoils. J. Acoust. Soc. Am. **141**(5), 3291–3301 (2017)
36. Barnard, A.R.: Flow induced noise reduction techniques for microphones. Sound Vibr. **48**, 10 (2014)
37. Al Tlua, B., Rocha, J.: Experimental investigation of NACA-0012 airfoil instability noise with sawtooth trailing edges. Wind Eng. **46**(3), 972–982 (2022)
38. Siddique, M.F., Ahmad, Z., Ullah, N., Kim, J.: A hybrid deep learning approach: integrating short-time Fourier transform and continuous wavelet transform for improved pipeline leak detection. Sensors **23**(19), 8079 (2023)

Beyond Endpoint Quality: A Phase-Aware Analysis of Search Dynamics in Triangle Reassembly

Damir Hasanspahić(✉), Adis Alihodžić, and Damir Hasić

University of Sarajevo, Sarajevo, Bosnia and Herzegovina
damir.hasanspahic@gf.unsa.ba, {adis.alihodzic,d.hasic}@pmf.unsa.ba

Abstract. This paper studies the reassembly of triangles into the unit square from scattered Delaunay triangulations. Each triangle is independently translated and rotated, all adjacency information is removed, and the objective is to reconstruct a feasible non-overlapping arrangement that maximizes coverage. We compare Variable Neighborhood Search, Iterated Local Search, Artificial Bee Colony, IterativeSort, and IterativeRand under the same permutation representation, geometric evaluation operator, feasibility checks, and a budget of 21 evaluation calls on instances with 94–994 triangles. The analysis tracks phase-end best-so-far values, improvement incidence and gains, within-phase score levels, and the point at which the run-best value is reached before stagnation. Variable Neighborhood Search gives the strongest overall phase profile and remains the most reliable default choice, especially on larger instances. Iterated Local Search concentrates improvements earliest; Artificial Bee Colony shows the most persistent late-phase improvement behavior; and IterativeSort serves as a good heuristic starting point, especially in the middle of the search. Comparable terminal quality can therefore conceal materially different phase-wise search dynamics.

Keywords: triangle reassembly · Delaunay triangulation · geometric reconstruction · computational geometry · combinatorial optimization · metaheuristics · search dynamics · phase-aware analysis

1 Introduction

Packing and geometric reassembly problems study how finitely many geometric objects can be placed inside a bounded region under containment and non-overlap constraints while optimizing a global objective. When the pieces are irregular and their original neighborhood structure is unavailable, the search space combines a continuous geometric component with a discrete combinatorial one, yielding the nonconvex and fragmented behavior typical of hard packing problems [3,6,7,13,16,22,23]. In evaluation-intensive settings, the trajectory of progress also becomes important: under a limited budget of expensive geometric

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 130–143, 2026.
https://doi.org/10.1007/978-3-032-30491-9_9

evaluations, endpoint quality alone can conceal substantial differences in how methods use the available search budget.

We study this issue for a triangle reassembly problem induced by Delaunay triangulations of the unit square. A point set in the square is triangulated, each triangle is then independently scattered by an arbitrary translation and an arbitrary rotation about some point in the plane, and all adjacency relations are erased. The task is to reconstruct as much of the original tiling as possible without overlap. Although the pieces come from an exact tiling, the lost combinatorial structure must be recovered implicitly from geometry alone. The problem therefore belongs to a combinatorial–geometric family of NP-hard problems [3,6,13,16] and exhibits a reconstruction setting in which the order of geometric attempts is particularly consequential.

Beyond its role as a controlled benchmark, the Delaunay-based reconstruction setting abstracts a class of situations in which irregular pieces must be placed or reassembled when original placement or adjacency information is unavailable. Related constrained-placement problems arise in irregular two-dimensional stock cutting and nesting, where irregular pieces are arranged within stock material without overlap and, in many applications, under rotation restrictions [10]. The geometric handling of such layouts is commonly supported by no-fit polygon constructions, including robust Minkowski-sum-based procedures for complex cases with holes, slits, and lock and key positions [8]. Related geometry-driven reassembly problems occur in 3D fragment reassembly for archaeology and forensic investigation, including the computer-aided restoration of damaged artifacts and evidence, where template guidance is integrated with fracture-region matching to recompose fragmented objects [29]. The simplified Delaunay setting used here is therefore not a direct model of these applications; rather, it provides a controlled way to isolate permutation and feasibility effects that also arise in broader constrained-placement and reassembly problems.

The present paper builds on the reconstruction setting introduced in [5]. Earlier experiments there considered instances induced by at most 60 points, i.e., up to 114 triangles. Here we retain the same reconstruction perspective but study instances induced by 50, 100, 200, 300, and 500 points, i.e., 94–994 triangles, under a shared geometric evaluation operator. Rather than altering the geometric core, we ask how different permutation-search paradigms behave when the evaluation mechanism, feasibility logic, and evaluation-call budget are all held fixed.

Our empirical focus is the phase structure of progress over the evaluation budget. The budget is partitioned into three equal call segments, and the comparison is organized around phase-end best-so-far profiles, within-phase score levels, productive-run incidence, improvement counts, jump amplitudes, the evaluation index k_{best} at which the eventual run-best value is reached, and the remaining stall length thereafter. Final coverage is reported as a supporting summary, whereas the primary focus is on when progress is produced and how it is distributed across phases.

Contributions

- We extend the scattered-triangulation reconstruction setting of [5] from the earlier limit of 60 points (114 triangles) to instances induced by up to 500 points (994 triangles) under a common unit-square model.
- We present a controlled comparative evaluation of three canonical metaheuristics and two lightweight heuristic methods within a unified experimental framework, using the same permutation representation, geometric evaluation operator, length-compatibility prior, and evaluation budget.
- We structure the empirical analysis around phase dynamics and show that methods with similar terminal coverage can differ substantially in when and how they generate, preserve, and exhaust progress throughout the run.

2 Related Work

Irregular packing and nesting form a broad class of problems in combinatorial optimization and computational geometry [7,22,23]. Polygon packing has been studied from theoretical, heuristic, and industrial perspectives, including maximum-packing heuristics [26] and nesting procedures based on no-fit polygons and Minkowski-sum constructions [7,8,10]. Across these settings, geometric feasibility and combinatorial choice remain tightly coupled, which is precisely the structural backdrop of the present reassembly problem.

Classical packing variants such as packing identical simple polygons, bin packing, strip packing, rectangle packing, and specialized triangle packing are NP-hard [3,6,13,16]. More recently, broad classes of two-dimensional packing problems with rotations have been shown to be $\exists\mathbb{R}$-complete [2], indicating a level of geometric hardness that goes beyond standard discrete-complexity formulations. These results place the present reconstruction task within a difficult combinatorial–geometric family.

Within the triangle-packing literature, graph-theoretic formulations study the existence of vertex-disjoint triangles as well as packing and covering variants [11,14,18,21,27]. In geometric and application-oriented settings, exact branch-and-bound procedures [1] and heuristic methods [12] have also been proposed. All of these formulations typically assume arbitrary input pieces or settings in which structural relations remain directly available.

Our setting differs from these formulations in one key respect: the pieces are derived from a coherent Delaunay triangulation that exactly tiles the container, but the adjacency structure is erased before the search begins, so compatibility must be reconstructed from geometry alone. This reconstruction perspective was introduced for scattered triangulations in [5], where geometry-driven constructive strategies were studied on instances up to 60 points, i.e., 114 triangles. The present paper follows that line of work at a larger scale and shifts the empirical emphasis from endpoint quality alone to the fixed-budget phase structure of search progress under a shared geometric evaluation operator.

3 Methods

3.1 Problem Definition

We restate the model of [5], specialized to the unit square. Let $R = [0,1] \times [0,1]$ be the unit square, with area $A(R) = 1$. Let $X = \{x_1, \ldots, x_n\} \subset R$ contain the four corners of R, with the remaining $n-4$ points strictly in the interior and no three points collinear. Let $T = \{T_1, \ldots, T_N\}$ be the Delaunay triangulation of X; then T partitions R into non-overlapping triangles, and since the convex hull has four vertices, $N = 2n - 6$ [4,9,15]. Each triangle T_i is then scattered by an independent planar rigid motion, i.e., by an arbitrary translation and an arbitrary rotation about some point, producing an unordered set $T' = \{T'_1, \ldots, T'_N\}$ with all positional and adjacency information removed [5].

A feasible reconstruction consists of a subset $S \subseteq T'$ together with a placement mapping $f : S \to \mathbb{R}^2$ that assigns to each triangle a rigid motion, such that

$$f(T'_i) \cap f(T'_j) = \emptyset \quad \forall\, T'_i \neq T'_j \in S, \tag{1}$$

$$f(T'_i) \subseteq R \quad \forall\, T'_i \in S. \tag{2}$$

The objective is to maximize covered area. Since $A(R) = 1$, the percentage coverage is

$$P(S) = 100 \cdot \sum_{T'_i \in S} A(T'_i), \tag{3}$$

and, because the triangles of T tile R exactly, the optimum value is 100%.

3.2 Shared Geometric Evaluation Operator

All compared methods search in the same solution space, namely permutations of the triangle set, and every candidate permutation is assessed by the same deterministic *Geometric Evaluation Operator*. For a given permutation π, the operator processes triangles in the prescribed order and incrementally attempts to build a feasible layout inside the unit square; its output is the resulting coverage score $P(\pi)$. Because the evaluation operator is fixed for all methods, the comparison isolates search behavior rather than geometric processing or feasibility handling.

The first triangle in the permutation serves as an anchor and is placed at an arbitrary feasible position inside the unit square, after which subsequent triangles are tentatively aligned to exposed edges of the current partial layout. The operator follows an incremental edge-attachment principle in the spirit of the S2/S3 constructive schemes in [5]. A candidate placement is accepted only if it remains inside the container and does not overlap previously placed triangles. To control geometric branching on larger instances, candidate edge pairs are considered only when $\ell_{\max} \leq 1.5\,\ell_{\min}$, i.e., when the longer edge is at most 50% longer than the shorter one. This filter is applied uniformly to all methods.

To keep evaluation stable up to $N \approx 10^3$, the operator uses uniform-grid spatial hashing for localized overlap queries, prioritized inspection of the 3×3

neighboring cells around the candidate location, coarse filters such as axis-aligned bounding-box checks before exact intersection testing, and pair-skip caching for edge pairs that have already failed within the current partial layout. These mechanisms are shared by all methods and leave the objective function and admissibility conditions unchanged.

Each evaluation call returns one scalar score $P(\pi)$, and all phase-aware descriptors are defined on the resulting ordered sequence of evaluation calls.

3.3 Compared Algorithms

All methods considered in this study operate in the same permutation space and are evaluated by the shared Geometric Evaluation Operator introduced in Sect. 3.2. The comparison is therefore designed to isolate search policy under a fixed evaluation mechanism. To preserve fairness, all selected metaheuristics are used in minimal canonical implementations, and their control parameters retain their standard roles.

Variable Neighborhood Search (VNS). VNS follows the classical principle of systematic neighborhood change combined with local improvement [17,28]. The search state is a permutation, and the method traverses neighborhoods of increasing scope through standard permutation perturbations. Whenever an improving solution is found, the search returns to the first neighborhood, in accordance with the standard VNS scheme. In the present study, VNS serves as the variable-neighborhood search component under the common evaluation setting.

Iterated Local Search (ILS). ILS follows the standard perturb–improve–accept pattern [24,25]. Starting from a current permutation, the method applies a perturbation, performs local improvement on the perturbed solution, and then decides whether the resulting local optimum replaces the incumbent one. In this setting, ILS provides a compact trajectory-based intensification baseline for the shared permutation landscape.

Artificial Bee Colony (ABC). ABC follows the standard employed/onlooker/scout organization [19,20]. In the present setting, candidate permutations act as food sources, and their quality is determined by the score returned by the shared geometric evaluation operator. The employed and onlooker phases explore modified permutations around existing candidates, whereas the scout phase restores diversification when progress stalls. ABC therefore represents the population-based search component in the comparison.

IterativeSort. IterativeSort is a deterministic heuristic reference derived from the descending-area ordering principle used in the earlier S2-style construction strategy [5]. The triangles are sorted in non-increasing order of area, and the

resulting permutation is submitted directly for evaluation. Its role is to provide a simple structured ordering rule without adaptive search, population dynamics, or iterative refinement.

IterativeRand. IterativeRand is a low-information heuristic reference in which candidate solutions are generated as random permutations of the triangle set. By removing the explicit size-based ordering signal present in IterativeSort, it provides a reference level for performance obtainable without structured guidance. It is therefore useful for separating the contribution of deliberate search from the contribution of the fixed evaluation mechanism itself.

The selected metaheuristics were chosen so as to span complementary search paradigms within the same permutation-based framework. Their joint consideration provides a sufficiently broad, though not exhaustive, view of the behavior of different metaheuristic search paradigms.

4 Experimental Design

Instance Sizes. We consider five point-set sizes, $n \in \{50, 100, 200, 300, 500\}$, which induce triangulation sizes $N = 2n - 6 \in \{94, 194, 394, 594, 994\}$.

Protocol. All methods are evaluated under the same regime, with computational effort controlled exclusively by the number of calls to the Geometric Evaluation Operator. Each run therefore uses the same fixed budget $B = 21$ evaluation calls, partitioned into three equal segments $F1 = \{1, \ldots, 7\}$, $F2 = \{8, \ldots, 14\}$, and $F3 = \{15, \ldots, 21\}$. These segments define the three analysis phases used throughout the paper and place progress directly on consumed evaluation budget, making it visible whether gains are concentrated early, sustained through F2, or preserved into F3. The choice $B = 21$ is supported not only by the phase-analysis design but also by supplementary budget-extension checks on the smallest instance, where a threefold larger horizon did not yield systematic or proportionate coverage gains across methods. In view of the problem complexity and the execution time required by larger instances, the number of independent runs was adjusted stepwise according to instance size: 20 runs for $N \in \{94, 194\}$, 10 runs for $N \in \{394, 594\}$, and 5 runs for $N = 994$. Reported summaries aggregate method–instance–run observations either globally or by size class, depending on the corresponding table or figure.

Phase-Aware Descriptors. Let s_k denote the score returned by the k-th evaluation call, and let $b_k = \max_{1 \le t \le k} s_t$ denote the best-so-far value after k evaluations. From these sequences, we derive the following descriptors. The terminal run quality is defined as $P = b_{21}$. For each phase F_j, we record the phase-end best-so-far value $F_{end,j}$, the number of strict best-so-far improvements $N_{imp,j}$, the productive-run incidence $Inc_j = \Pr(N_{imp,j} \ge 1)$, and the largest best-so-far jump $D_{max,j}$. To characterize the level of sampled solutions within a phase, we also record the phase mean and phase maximum, denoted by $S_{mean,j}$ and

$S_{max,j}$. To describe where progress is ultimately realized along the evaluation budget, we further record the evaluation-call index at which the eventual run-best value is last attained, denoted by k_{best}, together with the subsequent stall length, i.e., the number of remaining evaluation calls after k_{best} without further strict improvement. For phases $F2$ and $F3$, we also report the aggregate phase gain, defined as $Gain_j = F_{end,j} - F_{end,j-1}$. Within each productive phase, we summarize the local positions of the first and last strict improvement, denoted by imp@k and lastImp@k, where local positions are counted from 1 to 7 inside the corresponding phase segment.

These descriptors separate three complementary aspects of behavior: attained best-so-far quality, within-phase score levels, and where improvements occur within the evaluation budget. Jump-amplitude analysis is reported for F2 and F3, the phases in which nonzero best-so-far jumps are observed.

For inferential support, we use a light testing layer consisting of global Kruskal–Wallis tests and selected pairwise contrasts with Holm adjustment.

Implementation and Execution Environment. All algorithms were implemented in Python 3.11 and executed in single-threaded mode. All experiments were run on one x86-64 workstation equipped with an Intel® Core i5-3360M processor at 2.80 GHz and 8 GB of RAM. Because every method uses the same preprocessing pipeline, feasibility logic, and shared geometric evaluation operator under the same hardware and software configuration, the reported differences are attributable to search behavior within the common evaluation framework rather than to variation in the computational environment.

5 Discussion

All score-like coverages are reported in percentages, so gains and gaps are interpreted in percentage points. Phase-resolved descriptors refer to positions within the sequence of evaluation calls.

Accordingly, the phase-resolved results are interpreted not merely as rankings of terminal coverage, but also as diagnostics of search roles: strong opening levels indicate robust initial construction, middle-phase gains indicate useful exploitation after the opening layout, and late gains indicate remaining search reserve under a tight evaluation budget.

The highest observed final coverages are attained on $N = 994$ for all methods: VNS reaches 92.92%, ILS 92.50%, IterativeSort 92.44%, ABC 92.43%, and IterativeRand 86.92%. Figure 1 shows that mean final coverage also converges as N grows; on $N = 994$, the four strongest methods lie within 0.29% points in mean P (VNS 92.51%, ILS 92.39%, IterativeSort 92.28%, ABC 92.23%). Mean runtime grows with size in the expected way, from about 6.5–8.0 s at $N = 94$ to about 1520–1640 s at $N = 994$. The main separation therefore lies in where best-so-far progress occurs across the 21 evaluation calls.

Table 1 shows that VNS attains the highest mean final coverage (88.76%), followed by IterativeSort (88.38%), while ILS reaches the final best-so-far value

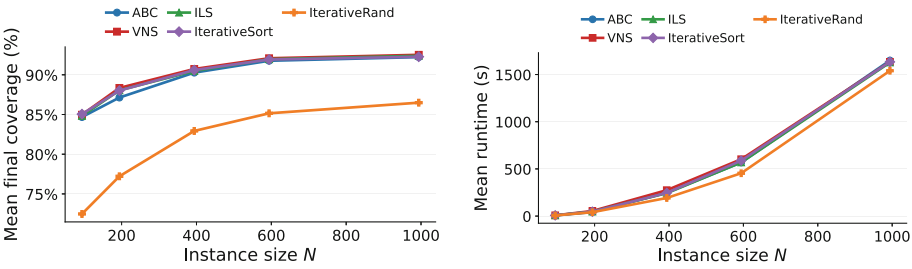


Fig. 1. Mean final coverage and mean runtime by instance size.

Table 1. Overall mean (std) final coverage and evaluation-call indices across all instances.

Method	P (%)	k_{best}	Stall
ABC	87.85 (2.99)	12.05	8.95
VNS	88.76 (2.92)	9.17	11.83
ILS	87.84 (2.99)	8.77	12.23
IterativeSort	88.38 (2.69)	10.38	10.62
IterativeRand	79.77 (5.24)	10.77	10.23

at the earliest mean evaluation-call index ($k_{best} = 8.77$). ABC reaches that point latest ($k_{best} = 12.05$) and therefore leaves the shortest trailing stall. These three descriptors already separate the main profiles: VNS is strongest in level, ILS is earliest in consolidation, and ABC preserves the largest post-opening reserve.

Opening Segment. Figure 2, Table 2, and Table 4 show that all methods remain productive in F_1, so the separation in the opening phase arises primarily through the level of the best-so-far trajectory rather than through the mere incidence of productive starts. Within that common setting, VNS builds the strongest opening level, with $F_{end,1} = 88.09\%$, and also leads the sampled-score descriptors in the first phase, with $S_{mean,1} = 85.01\%$ and $S_{max,1} = 88.09\%$. It then retains the highest phase-end value through F_2 and F_3. The same figure shows that VNS also leads $S_{mean,j}$ in all three phases, indicating that its advantage is not limited to isolated best-so-far updates. The same ordering is preserved by $S_{max,j}$, so within-phase maxima do not overturn the ranking established by $F_{end,j}$ and $S_{mean,j}$. IterativeSort (87.73%) and ILS (87.43%) remain close on $F_{end,1}$, which explains why the deterministic baseline already matches the leading metaheuristics after the first seven calls. ABC also reaches a competitive opening endpoint, with $F_{end,1} = 86.77\%$, but its markedly lower $S_{mean,1} = 77.46\%$ points to a wider gap between sampled solutions and the best-so-far path in the opening segment. IterativeRand records the highest early activity, with $N_{imp,1} = 2.81$, yet its opening score level remains clearly lower, at $F_{end,1} = 79.18\%$, showing that more frequent early improvements do not by themselves translate into

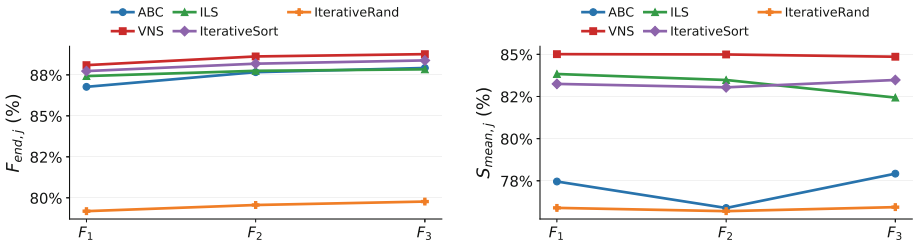


Fig. 2. Phase-end best-so-far levels and within-phase mean scores.

Table 2. Phase-end best-so-far levels and aggregate phase gains.

Method	$F_{end,1}$	$F_{end,2}$	$F_{end,3}$	$Gain_2$	$Gain_3$
ABC	86.77 (4.17)	87.66 (3.15)	87.92 (2.97)	0.89 (1.85)	0.26 (0.54)
VNS	88.09 (3.37)	88.62 (3.01)	88.76 (2.92)	0.53 (1.11)	0.14 (0.44)
ILS	87.43 (3.20)	87.75 (3.00)	87.84 (2.99)	0.32 (0.78)	0.09 (0.25)
IterativeSort	87.73 (3.09)	88.18 (2.86)	88.38 (2.69)	0.45 (0.98)	0.20 (0.73)
IterativeRand	79.18 (5.54)	79.56 (5.36)	79.77 (5.24)	0.38 (0.73)	0.21 (0.43)

stronger phase-end quality. Table 3 further refines the same picture: ILS closes productive F_1 progress earliest on average (lastImp@F_1 = 3.74), whereas IterativeSort remains active slightly deeper into the opening segment (4.07). The opening phase therefore separates methods both by attained level and by how quickly early progress is consolidated, while also revealing differences in score dispersion around the leading trajectory.

Table 3. Mean local positions of strict improvements on the phase-local scale 1–7.

Method	lastImp@F_1	imp@F_2	lastImp@F_2	imp@F_3	lastImp@F_3
ABC	3.91	3.79	3.97	5.37	5.57
VNS	3.90	3.03	4.00	3.65	4.00
ILS	3.74	3.56	4.03	4.46	4.92
IterativeSort	4.07	3.36	4.23	3.59	4.15
IterativeRand	4.28	3.64	4.06	3.77	4.05

Middle Segment. Table 5, Table 2, and Table 4 show that F_2 is the main point of behavioural separation among the methods. IterativeSort is the most reliable producer of additional F_2 updates, with the highest $N_{imp,2}$ (0.67) and the highest productive-run incidence Inc_2 (52.5%). ABC, however, converts productive F_2 runs into the largest gain and the largest jump amplitude, leading both $Gain_2$

Table 4. Within-phase score levels and phase maxima.

Method	$S_{mean,1}$	$S_{mean,2}$	$S_{mean,3}$	$S_{max,1}$	$S_{max,2}$	$S_{max,3}$
ABC	77.46	75.89	77.92	86.77	85.81	86.80
	(6.91)	(7.62)	(6.58)	(4.17)	(5.01)	(3.71)
VNS	85.01	84.99	84.86	88.09	88.13	88.01
	(5.77)	(5.69)	(5.66)	(3.37)	(3.37)	(3.37)
ILS	83.83	83.48	82.44	87.43	87.01	86.21
	(5.38)	(5.29)	(7.15)	(3.20)	(3.45)	(4.29)
IterativeSort	83.25	83.04	83.48	87.73	87.64	87.80
	(6.44)	(6.61)	(5.84)	(3.09)	(3.47)	(2.96)
IterativeRand	75.89	75.70	75.94	79.18	78.91	79.00
	(7.61)	(7.75)	(7.65)	(5.54)	(5.90)	(5.81)

(0.89%) and $D_{max,2}$ (0.88%). At the same time, VNS remains the strongest method in terms of absolute level in the middle phase, with the highest values of $F_{end,2} = 88.62\%$, $S_{mean,2} = 84.99\%$, and $S_{max,2} = 88.13\%$. Figure 3 shows where these updates occur within the phase. In F_2, ABC concentrates both first and last productive updates around local position 4, whereas IterativeSort spreads productive events more broadly across positions 3–5. This distinction explains why IterativeSort is the most reliable middle-phase updater, while ABC is the strongest middle-phase converter once an update occurs, with VNS nevertheless preserving the highest overall middle-phase quality level.

Table 5. Productivity, incidence (%), and jump descriptors by phase.

Method	$N_{imp,1}$	$N_{imp,2}$	Inc_2	$N_{imp,3}$	Inc_3	$D_{max,2}$	$D_{max,3}$
ABC	1.68	0.46	42.3	0.42	38.5	0.88	0.25
	(0.69)	(0.57)		(0.59)		(1.78)	(0.54)
VNS	2.46	0.61	44.0	0.24	20.2	0.46	0.13
	(0.99)	(0.81)		(0.51)		(0.94)	(0.40)
ILS	2.66	0.52	43.8	0.21	17.8	0.29	0.08
	(1.15)	(0.65)		(0.47)		(0.68)	(0.22)
IterativeSort	2.57	0.67	52.5	0.31	26.7	0.41	0.18
	(1.08)	(0.74)		(0.54)		(0.90)	(0.67)
IterativeRand	2.81	0.57	47.8	0.35	31.9	0.37	0.21
	(1.09)	(0.67)		(0.54)		(0.73)	(0.42)

Late Segment. The same descriptors show a clearer separation in F_3, where the late segment distinguishes residual improvement reserve from early saturation.

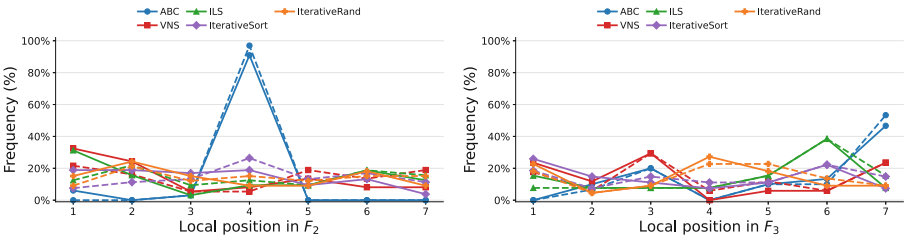


Fig. 3. Local-position frequencies of first (solid) and last (dashed) strict improvements in productive runs of F_2 and F_3.

Table 5, Table 2, and Table 4 show that ABC leads all late-phase productivity measures: $N_{imp,3} = 0.42$, $Inc_3 = 38.5\%$, $Gain_3 = 0.26\%$, and $D_{max,3} = 0.25\%$. Table 3 and Fig. 3 show that this late reserve is also deeper in the phase: ABC records imp@F_3 = 5.37 and lastImp@F_3 = 5.57, both later than the corresponding means of the other methods. VNS still keeps the highest late-phase level in Fig. 2 and remains strongest on the late sampled-score descriptors, with $S_{mean,3} = 84.86\%$ and $S_{max,3} = 88.01\%$, but its late productive incidence is much lower (20.2%). ILS reaches its final best-so-far value early in the budget and correspondingly retains the profile in which most progress is realized earlier, showing the smallest late productive incidence (17.8%). The late segment therefore distinguishes level preservation from residual improvement reserve: VNS preserves the highest level, whereas ABC retains the strongest capacity for additional best-so-far progress.

Instance Scale and Synthesis. Figure 4 shows that ABC's $Gain_2$ advantage is concentrated on the smaller and medium instances: 1.07% at $N = 94$ and 1.64% at $N = 194$, compared with only 0.03%–0.08% on $N = 594$ and $N = 994$. This explains why endpoint coverage converges on the largest instances even though the phase profiles remain distinct. On $N = 994$, mean final coverage is already tightly clustered among VNS, ILS, IterativeSort, and ABC, but Table 1, Table 5, and Table 3 still separate them by where progress is realized: VNS remains strongest in phase-end level, ILS reaches the final best-so-far value earliest, ABC retains the largest post-F_1 improvement reserve, and IterativeSort remains a good deterministic reference through reliable F_2 productivity.

Global Tests. Kruskal–Wallis tests detect between-method differences in phase-end quality ($F_{end,1}$–$F_{end,3}$: $H = 113.76$–130.29, all $p < 0.001$) and in the evaluation-call index at which the final best-so-far value is reached (k_{best}: $H = 12.67$, $p = 0.013$). For the selected phase-oriented pairwise contrasts, Holm-adjusted tests show that ABC and ILS differ both in late-phase gain ($Gain_3$: $p_{Holm} = 0.0060$) and in k_{best} ($p_{Holm} = 0.0060$). These findings remain consistent with the descriptive reading: the methods separate primarily by phase profile and by the evaluation-call indices at which progress is realized, rather than by terminal level alone.

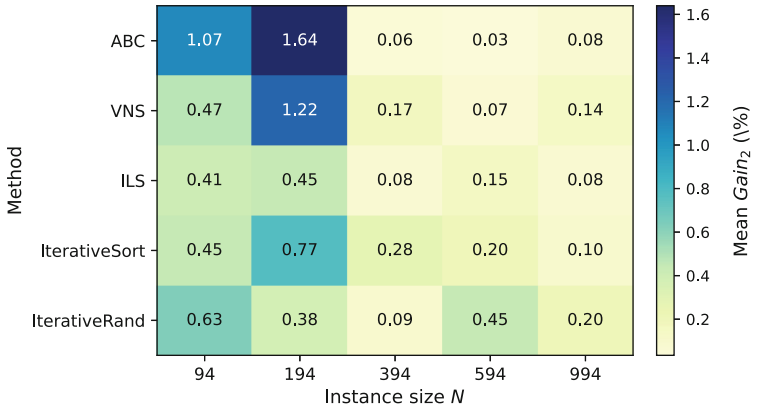


Fig. 4. Mean $Gain_2$ by instance size.

Robustness. A supplementary robustness check on the smallest instance ($N = 94$) under a threefold larger budget ($B = 63$) leads to the same practical conclusion. For the competitive methods, the mean gain realized after F1 remained modest: 0.04% points for ILS, 0.44 for VNS, 0.55 for ABC, and 0.67 for IterativeSort. Some individual runs still improved in F2 or F3, but the added calls did not produce a systematic or proportionate increase in coverage. This supports the use of $B = 21$ as a controlled phase-analysis budget rather than a budget chosen in a regime where substantially larger horizons would reliably transform the outcome.

6 Conclusion

We studied the reassembly of triangles into the unit square from scattered Delaunay triangulations through a controlled phase-aware comparison in which all methods shared the same permutation representation, geometric evaluation operator, admissibility logic, and evaluation-call budget. The benchmark extends the earlier reconstruction setting from instances with up to 114 triangles to instances with up to 994 triangles. Within this common setting, the results are most informative when viewed phase-wise, as the methods differ less in terminal quality than in how improvement unfolds over the evaluation budget. VNS delivers the strongest overall best-so-far profile and remains the most reliable default choice, especially on larger instances. IterativeSort stands out as a good heuristic starting point and the most reliable improver in F2. ILS exhibits the clearest early intensification, but also saturates sooner. ABC retains the strongest late-phase elasticity, with the highest residual productivity in F3 and the largest productive late moves among the competing methods.

Taken together, these findings show that similar terminal quality can conceal substantially different search dynamics.

A similar phase-structured analysis protocol could also provide a useful basis for examining broader families of baseline methods, adaptive hybrids, and related irregular reconstruction or polygon-packing settings where evaluation constitutes a substantial computational cost.

References

1. Abdelsadek, Y., Herrmann, F., Kacem, I., Otjacques, B.: Branch-and-bound algorithm for the maximum triangle packing problem. Comput. Industr. Eng. **81**, 147–157 (2015). https://doi.org/10.1016/j.cie.2014.12.006
2. Abrahamsen, M., Miltzow, T., Seiferth, N.: Framework for ∃ℝ-completeness of two-dimensional packing problems. TheoretiCS **3**, 11 (2024). https://doi.org/10.46298/theoretics.24.11
3. Adamaszek, A., Kociumaka, T., Pilipczuk, M., Pilipczuk, M.: Hardness of approximation for strip packing. ACM Trans. Comput. Theor. **9**(3), 14:1–14:7 (2017). https://doi.org/10.1145/3092026
4. Alihodžić, A.: Exploring Computational Geometry: Theory and Python Implementations. Springer, Cham (2026). https://doi.org/10.1007/978-3-032-06393-9
5. Alihodžic, A., Hasanspahic, D., Tuba, E., Hasic, D.: Triangle packing strategies within a rectangle. In: Proceedings of the IEEE International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA), Boracay, Philippines (2026, to appear). https://doi.org/10.1109/ACDSA67686.2026.11468118, https://ieeexplore.ieee.org/document/11468118
6. Allen, S.R., Iacono, J.: Packing identical simple polygons is NP-hard (2012). https://arxiv.org/abs/1209.5307
7. Bennell, J.A., Oliveira, J.F.: The geometry of nesting problems: a tutorial. Eur. J. Oper. Res. **184**(2), 397–415 (2008). https://doi.org/10.1016/j.ejor.2006.11.038
8. Bennell, J.A., Song, X.: A comprehensive and robust procedure for obtaining the nofit polygon using Minkowski sums. Comput. Oper. Res. **35**(1), 267–281 (2008). https://doi.org/10.1016/j.cor.2006.02.026
9. de Berg, M., Cheong, O., van Kreveld, M., Overmars, M.: Computational Geometry: Algorithms and Applications, 3rd edn. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-77974-2
10. Burke, E.K., Hellier, R.S.R., Kendall, G., Whitwell, G.: Complete and robust no-fit polygon generation for the irregular stock cutting problem. Eur. J. Oper. Res. **179**(1), 27–49 (2007). https://doi.org/10.1016/j.ejor.2006.03.011
11. Caprara, A., Rizzi, R.: Packing triangles in bounded degree graphs. Inf. Process. Lett. **84**(4), 175–180 (2002). https://doi.org/10.1016/S0020-0190(02)00274-0
12. Chen, C.B., He, D.H.: A heuristic method for solving triangle packing problem. J. Zhejiang Univ. Sci. A **6**(6), 565–570 (2005). https://doi.org/10.1631/jzus.2005.A0565
13. Chou, A.: NP-hard triangle packing problems. Research Science Institute manuscript, January 2016. https://math.mit.edu/research/highschool/rsi/documents/2015Chou.pdf
14. Corrádi, K., Hajnal, A.: On the maximal number of independent circuits in a graph. Acta Mathematica Academiae Scientiarum Hungaricae **14**(3–4), 423–439 (1963). https://doi.org/10.1007/BF01895727
15. Delaunay, B.: Sur la sphère vide. A la mémoire de Georges Voronoï. Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et naturelles (6), 793–800 (1934)

16. Fowler, R.J., Paterson, M.S., Tanimoto, S.L.: Optimal packing and covering in the plane are NP-complete. Inf. Process. Lett. **12**(3), 133–137 (1981). https://doi.org/10.1016/0020-0190(81)90111-3
17. Hansen, P., Mladenović, N.: Variable neighborhood search: principles and applications. Eur. J. Oper. Res. **130**(3), 449–467 (2001). https://doi.org/10.1016/S0377-2217(00)00100-4
18. Haxell, P.E.: Packing and covering triangles in graphs. Discret. Math. **195**, 251–254 (1999). https://doi.org/10.1016/S0012-365X(98)00183-6
19. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Tech. Rep. TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, October 2005. https://abc.erciyes.edu.tr/pub/tr06_2005.pdf
20. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Global Optim. **39**(3), 459–471 (2007). https://doi.org/10.1007/s10898-007-9149-x
21. Kirkpatrick, D.G., Hell, P.: On the completeness of a generalized matching problem. In: Proceedings of the 10th Annual ACM Symposium on Theory of Computing, pp. 240–245 (1978). https://doi.org/10.1145/800133.804353
22. Leao, A.A.S., Toledo, F.M.B., Oliveira, J.F., Carravilla, M.A., Alvarez-Valdés, R.: Irregular packing problems: a review of mathematical models. Eur. J. Oper. Res. **282**(3), 803–822 (2020). https://doi.org/10.1016/j.ejor.2019.04.045
23. Lodi, A., Martello, S., Monaci, M.: Two-dimensional packing problems: a survey. Eur. J. Oper. Res. **141**(2), 241–252 (2002). https://doi.org/10.1016/S0377-2217(02)00123-6
24. Lourenço, H.R., Martin, O.C., Stützle, T.: Iterated local search. In: Glover, F., Kochenberger, G.A. (eds.) Handbook of Metaheuristics, International Series in Operations Research and Management Science, vol. 57, pp. 320–353. Springer, Boston, MA (2003). https://doi.org/10.1007/0-306-48056-5_11
25. Lourenço, H.R., Martin, O.C., Stützle, T.: Iterated local search: framework and applications, 2nd edn. In: Gendreau, M., Potvin, J.Y. (eds.) Handbook of Metaheuristics, International Series in Operations Research & Management Science, vol. 146, pp. 363–397. Springer, New York (2010). https://doi.org/10.1007/978-1-4419-1665-5_12
26. Luo, C., Su, Z., Lü, Z.: A general heuristic approach for maximum polygon packing. In: 40th International Symposium on Computational Geometry, SoCG 2024. Leibniz International Proceedings in Informatics (LIPIcs), vol. 293, pp. 86:1–86:9 (2024). https://doi.org/10.4230/LIPIcs.SoCG.2024.86
27. Manić, G., Wakabayashi, Y.: Packing triangles in low degree graphs and indifference graphs. Discret. Math. **308**(8), 1455–1471 (2008). https://doi.org/10.1016/j.disc.2007.07.100
28. Mladenović, N., Hansen, P.: Variable neighborhood search. Comput. Oper. Res. **24**(11), 1097–1100 (1997). https://doi.org/10.1016/S0305-0548(97)00031-2
29. Zhang, K., Yu, W., Manhein, M., Waggenspack, W., Li, X.: 3D fragment reassembly using integrated template guidance and fracture-region matching. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2138–2146 (2015). https://doi.org/10.1109/ICCV.2015.247

Mesh Optimization for the One-Way Helmholtz Equation in an Inhomogeneous Medium

Mikhail S. Lytaev(✉)

St. Petersburg Federal Research Center of the Russian Academy of Sciences, 14-th Linia, V.I., No. 39, Saint Petersburg 199178, Russia
mikelytaev@gmail.com

Abstract. This paper presents an algorithm for the automated selection of optimal computational grid parameters in finite-difference Padé approximations of the two-dimensional one-way Helmholtz equation in an inhomogeneous medium. Unlike previous works, where the optimization was performed for a homogeneous medium or accounted only for the propagation angle, the proposed algorithm explicitly takes into account the range of the local wavenumber. The grid steps along the longitudinal and transverse coordinates are selected so as to minimize the computational cost while maintaining a prescribed dispersion error tolerance. A tabulation-based implementation enables real-time grid selection. The algorithm is analyzed on two fundamentally different classes of problems: tropospheric radio wave propagation, where the wavenumber is nearly constant, and underwater acoustics, where it varies significantly between the water column and the sediment. It is demonstrated that ignoring the wavenumber variation leads to grids that are 10–40 times too sparse for underwater acoustics problems, resulting in unacceptable modeling errors.

Keywords: wave propagation · parabolic equation · mesh optimization · Padé approximation · refractive index · underwater acoustics · radio wave propagation

1 Introduction

Computer simulation of wave field propagation in inhomogeneous media is a fundamental problem in mathematical physics, arising in radiophysics [11], computational underwater acoustics (UWA) [5,9], atmospheric acoustics [3], medical ultrasound [18], seismology [7], and optics [2]. One of the most effective and versatile numerical approaches is the parabolic equation (PE) method, originally proposed by Leontovich and Fock [10] for tropospheric radio wave propagation (RWP). The PE method and its wide-angle generalizations, the one-way Helmholtz equation method [8], have become indispensable tools in applied wave physics, owing to their deterministic nature and high computational efficiency.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 144–160, 2026.
https://doi.org/10.1007/978-3-032-30491-9_10

Modern implementations rely on the rational Padé approximation of the pseudo-differential propagation operator [1,6]. The accuracy and efficiency of the resulting numerical scheme depend critically on several computational parameters: the longitudinal grid step Δx, the transverse grid step Δz, the reference propagation constant β, and the order of the Padé approximation. Choosing a grid that is too coarse leads to unacceptable dispersion errors, while an unnecessarily dense grid wastes computational resources. In practice, these parameters are often selected manually, which is error-prone and prevents the integration of wave propagation solvers into automated pipelines.

The problem of automated parameter selection has been addressed in a series of works. In [12], presented at ICCSA 2020, the dispersion analysis of the finite-difference Padé scheme was used to formulate an optimization problem for the two-dimensional homogeneous case. In [14], presented at ICCSA 2023, the approach was extended to the three-dimensional case, including the alternating-direction implicit (ADI) splitting, and a formula for the optimal reference propagation constant was derived. In [13], it was shown that rational interpolation on an interval allows the use of sparser grids than the local Padé approximation. In [15], the mesh optimization problem for underwater acoustics in an inhomogeneous medium was analyzed using a numerical search over the reference sound speed.

However, the algorithms in [12–14] assumed a homogeneous medium ($k = \text{const}$), and the numerical search in [15] did not provide an explicit closed-form procedure or jointly optimize Δx and Δz. The present work addresses these limitations. The main contributions are:

1. An explicit optimization algorithm for the two-dimensional one-way Helmholtz equation that jointly determines β, Δx, and Δz accounting for the full range of the local wavenumber $k(x, z) \in [k_{\min}, k_{\max}]$. The optimization is performed with respect to the reference propagation constant β rather than the reference sound speed.
2. A practical tabulation-based implementation enabling real-time grid selection.
3. A comparative analysis on tropospheric radio wave propagation (where $k_{\min} \approx k_{\max}$) and underwater acoustics (where they differ significantly), demonstrating that the wavenumber range is the dominant factor in the latter case.

2 Problem Statement

Consider the two-dimensional scalar Helmholtz equation

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial z^2} + k^2(x, z)\, \psi = 0, \tag{1}$$

where $\psi(x, z)$ is the wave field, x is the longitudinal coordinate, z is the transverse coordinate, $k(x, z) = 2\pi f / c(x, z)$ is the local wavenumber, f is the frequency, $c(x, z)$ is the spatially varying wave speed, and $\lambda = c/f$ is the wavelength.

The wave field is generated by the initial condition $\psi(0, z) = \psi_0(z)$, where ψ_0 represents the radiation source. The function ψ is subject to boundary conditions on the lower boundary (e.g., impedance) and the upper boundary (e.g., transparent). The wave propagation occurs predominantly in the positive x-direction.

The range of $k(x, z)$ differs fundamentally between the two application domains of interest. In tropospheric radio wave propagation, the refractive index $n = c_0/c(x, z) \approx 1$ to high precision (the modified refractivity deviates from unity by $O(10^{-4})$), so $k_{\min} \approx k_{\max}$. In underwater acoustics, the sound speed varies from $c \approx 1500\,\mathrm{m/s}$ (water) to $c \approx 1700$–$2000\,\mathrm{m/s}$ (sediment), so $k_{\min}/k_{\max}$ can be as low as 0.75, a much larger relative variation.

3 One-Way Helmholtz Equation and Its Numerical Solution

3.1 Transverse Discretization

Rewrite (1) as

$$\frac{\partial^2 \psi}{\partial x^2} + H\psi = 0, \qquad H\psi = \frac{\partial^2 \psi}{\partial z^2} + k^2(x, z)\,\psi. \tag{2}$$

The transverse Laplacian $\partial^2/\partial z^2$ is discretized by the Numerov scheme [17]:

$$\frac{\partial^2 u}{\partial z^2} \approx D_{\Delta z}\, u = \frac{1}{\Delta z^2}\,\delta^2 \left(1 + \alpha\,\delta^2\right)^{-1} u,$$

where $\delta^2 u_j = u_{j-1} - 2u_j + u_{j+1}$ and $\alpha = 0$ for the 2nd order scheme, $\alpha = 1/12$ for the 4th order scheme. Figure 1 shows the transverse discretization error as a function of k_z for both schemes. The 4th order scheme improves accuracy by approximately two orders of magnitude at no additional computational cost.

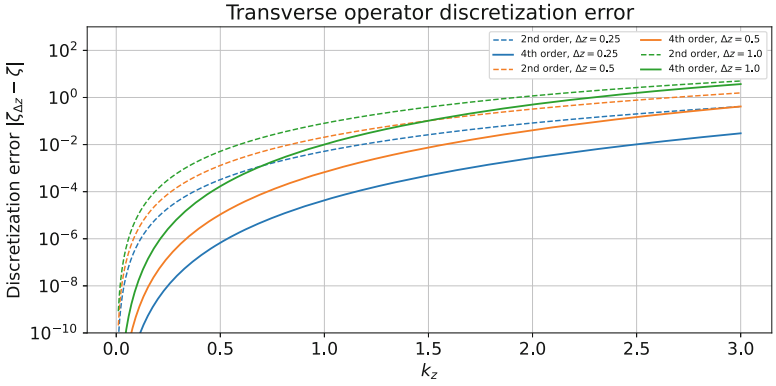


Fig. 1. Transverse operator discretization error for the 2nd and 4th order schemes at various Δz values.

3.2 Fourier Analysis and One-Way Factorization

Substituting the Fourier representation

$$\psi(x,z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \tilde{\psi}(x,k_z)\, e^{ik_z z}\, dk_z$$

into (2), the transverse operator acts on each Fourier component via its spectral symbol $\zeta(k_z)$:

$$e^{-ik_z z} \frac{\partial^2}{\partial z^2} e^{ik_z z} = -k_z^2,$$

$$e^{-ik_z z}\, D_{\Delta z}\, e^{ik_z z} = -\frac{1}{\Delta z^2}\left(4\sin^2\frac{k_z \Delta z}{2} + 16\alpha \sin^4\frac{k_z \Delta z}{2}\right). \tag{3}$$

Denoting the continuous symbol by $\zeta = -k_z^2$ and the discrete symbol by $\zeta_{\Delta z}$, the Helmholtz equation in the spectral domain reads

$$\frac{\partial^2 \tilde{\psi}}{\partial x^2} + \left(k^2(x,z) + \zeta\right)\tilde{\psi} = 0.$$

Factoring into forward- and backward-propagating components [5,8] and retaining the forward-propagating wave, we introduce the slowly varying envelope $u(x,z) = e^{-i\beta x}\psi(x,z)$. The step-by-step solution is

$$\tilde{u}(x + \Delta x, k_z) = P(\xi)\, \tilde{u}(x,k_z),$$

where

$$P(\xi) = \exp\left(i\tilde{\beta}\left(\sqrt{1+\xi} - 1\right)\right), \quad \tilde{\beta} = \beta \Delta x,$$

and

$$\xi = \frac{k^2(x,z) + \zeta}{\beta^2} - 1. \tag{4}$$

Here β is the reference propagation constant, a free parameter optimized by the algorithm (Sect. 4.3).

For the continuous operator, $\zeta = -k_z^2$ and $\xi = (k^2 - k_z^2)/\beta^2 - 1$, where $k_z = k\sin\theta$ for propagation angle θ. For the discrete operator, $\zeta = \zeta_{\Delta z}$ and

$$\xi_d = \frac{k^2(x,z) + \zeta_{\Delta z}}{\beta^2} - 1 = -\frac{4}{\beta^2 \Delta z^2}\left(\sin^2\frac{k_z \Delta z}{2} + 4\alpha \sin^4\frac{k_z \Delta z}{2}\right) + \frac{k^2(x,z)}{\beta^2} - 1. \tag{5}$$

The per-step error thus has two sources: (i) the *rational approximation error*, from approximating $P(\xi)$ (see below), and (ii) the *transverse discretization error*, from $\zeta \neq \zeta_{\Delta z}$, i.e., $\xi \neq \xi_d$.

3.3 Rational Padé Approximation

Following [6], we approximate the propagator by a rational function

$$P(\xi) \approx \tilde{P}(\xi) = \prod_{l=1}^{p} \frac{1 + a_l \xi}{1 + b_l \xi},$$

where a_l, b_l are the Padé coefficients of order $[m/n]$, $p = \max(m, n)$. Applying the inverse Fourier transform, the propagation step reduces to solving p tridiagonal systems:

$$\begin{cases} (1 + b_1 L)\, v_1 = (1 + a_1 L)\, u^{n-1}, \\ \quad \vdots \\ (1 + b_l L)\, v_l = (1 + a_l L)\, v_{l-1}, \qquad l = 2, \ldots, p-1, \\ \quad \vdots \\ (1 + b_p L)\, u^n = (1 + a_p L)\, v_{p-1}, \end{cases}$$

with transverse operator

$$Lu = \frac{1}{\beta^2} D_{\Delta z}\, u + \left(\frac{k^2(x, z)}{\beta^2} - 1 \right) u.$$

Throughout this paper, we use Padé approximations of order $[p-1, p]$, which are L-stable [4] and ensure attenuation of spurious high-frequency components.

4 Optimization Algorithm

4.1 Range of ξ in an Inhomogeneous Medium

From (4), the variable ξ ranges over

$$\xi \in \left[\frac{k_{\min}^2 - k_{z,\max}^2}{\beta^2} - 1,\ \frac{k_{\max}^2}{\beta^2} - 1 \right], \tag{6}$$

where $k_{\min} = \min_{x,z} k(x, z)$, $k_{\max} = \max_{x,z} k(x, z)$, and $k_{z,\max}$ is the maximum transverse wavenumber determined by the source. Note that the algorithm requires only these physical bounds; no artificial reference speed c_0 or reference wavenumber k_0 is needed.

This is the central observation of this paper: in an inhomogeneous medium, the ξ range depends not only on the maximum propagation angle $\theta_{\max}$ (through $k_{z,\max}$) but also on the ratio $k_{\min}/k_{\max}$. In tropospheric applications, $k_{\min} \approx k_{\max}$, so the range is essentially $[-\sin^2 \theta_{\max}, 0]$. In underwater acoustics, the term $k_{\min}^2/\beta^2 - 1$ can be much more negative (for $c_{\text{sed}} = 2000\,\text{m/s}$, $k_{\min}/k_{\max} \approx 0.75$), which dramatically widens the ξ interval. This is illustrated in Table 1.

Table 1. Range of ξ for various propagation angles and media. $\xi \in [\xi_{\min}, 0]$ where $\xi_{\min} = (k_{\min}/k_{\max})^2 - 1 - \sin^2\theta_{\max}$ (assuming $\beta = k_{\max}$).

$\theta_{\max}$	$\sin^2\theta_{\max}$	$\xi_{\min}$		
		RWP ($k_{\min} \approx k_{\max}$)	UWA ($c_{\text{sed}} = 1700$)	UWA ($c_{\text{sed}} = 2000$)
5°	0.008	−0.008	−0.229	−0.445
10°	0.030	−0.030	−0.252	−0.468
20°	0.117	−0.117	−0.338	−0.554
30°	0.250	−0.250	−0.471	−0.688
45°	0.500	−0.500	−0.721	−0.937

4.2 Error of the Rational Approximation

The absolute error of the rational approximation at each step is

$$R(\xi, \tilde{\beta}) = \left| P(\xi) - \prod_{l=1}^{p} \frac{1 + a_l \xi}{1 + b_l \xi} \right|.$$

Since the Padé approximation is centered at $\xi = 0$, the error increases monotonically with $|\xi|$ (Fig. 2). For each $\tilde{\beta}$ and threshold R_0, monotonicity guarantees the existence of bounds $\xi^- < 0 < \xi^+$ such that $R(\xi) \le R_0$ for all $\xi \in [\xi^-, \xi^+]$.

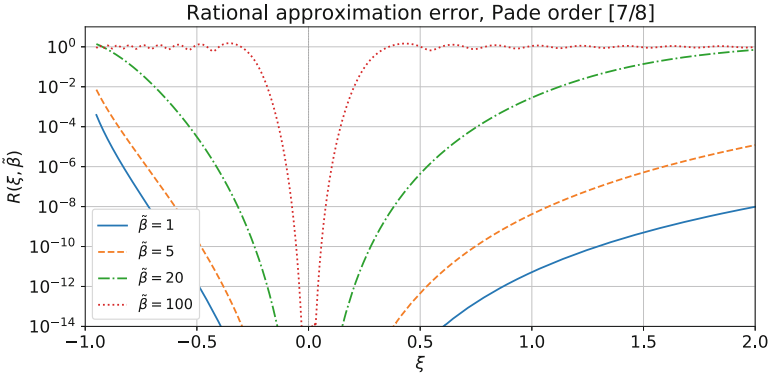


Fig. 2. Rational approximation error $R(\xi, \tilde{\beta})$ for Padé order [7/8]. The error is minimal near $\xi = 0$ and grows monotonically.

4.3 Constraints on β

For the rational approximation to be accurate over the full ξ range (6), we require $[\xi_{\min}, \xi_{\max}] \subseteq [\xi^-, \xi^+]$. This yields:

$$\beta \le \sqrt{\frac{k_{\min}^2 - k_{z,\max}^2}{\xi^- + 1}}, \qquad \beta \ge \sqrt{\frac{k_{\max}^2}{\xi^+ + 1}}. \tag{7}$$

A feasible β exists only when the lower bound does not exceed the upper bound. Among all feasible values, we select the smallest β, since the longitudinal step $\Delta x = \tilde{\beta}/\beta$ is maximized when β is minimized.

4.4 Transverse Discretization Error

From (4) and (5), the difference between the continuous and discrete ξ is

$$\Delta\xi = \xi - \xi_d = \frac{\zeta - \zeta_{\Delta z}}{\beta^2},$$

where ζ and $\zeta_{\Delta z}$ are the spectral symbols defined in (3). Explicitly,

$$|\Delta\xi| = \frac{1}{\beta^2}\left|k_z^2 - \frac{1}{\Delta z^2}\left(4\sin^2\frac{k_z\Delta z}{2} + 16\alpha\sin^4\frac{k_z\Delta z}{2}\right)\right|. \tag{8}$$

The impact of this discretization error on the propagator is estimated via

$$|P(\xi + \Delta\xi) - P(\xi)| \approx |P'(\xi)| \cdot |\Delta\xi|, \quad |P'(\xi)| = \frac{\tilde{\beta}}{2\sqrt{1+\xi}}.$$

The maximum of $|P'|$ on $[\xi^-, \xi^+]$ is at $\xi = \xi^-$. Requiring this error not to exceed R_0:

$$|\Delta\xi| \le \frac{2R_0\sqrt{1+\xi^-}}{\tilde{\beta}}. \tag{9}$$

Combining (8) and (9), the maximum allowable Δz is found by binary search.

4.5 Complete Procedure

The algorithm iterates over a precomputed table of candidate parameters $(\tilde{\beta}, R_0)$ and selects the combination maximizing $\Delta x \cdot \Delta z$ (minimizing computational cost) subject to the accumulated error constraint $\lceil x_{\max}/\Delta x \rceil \cdot R_0 < \varepsilon$.

The precomputed table uses grids $\mathcal{B} = \{0.01, 0.05, 0.1, \ldots, 50000\}$ (~ 50 values) and $\mathcal{R} = \{10^{-10}, \ldots, 10^{-5}\}$ (6 values), totaling $\sim$300 entries. The table depends only on the Padé order and can be reused across problems. The optimization loop performs only elementary arithmetic and at most one binary search per entry, completing in a fraction of a second. The proposed algorithm is implemented in the open-source Python library PyWaveProp [16].

5 Numerical Results

All results were obtained using PyWaveProp [16]. The 4th order Numerov scheme ($\alpha = 1/12$) is used throughout.

Algorithm 1. Grid optimization for the inhomogeneous PE.

Require: $k_{\min}$, $k_{\max}$, $k_{z,\max}$, ε, $x_{\max}$, Padé order $[m/n]$, transverse order α
Ensure: Optimal β, Δx, Δz

Precomputation (once per Padé order):
for all $\tilde{\beta} \in \mathcal{B}$, $R_0 \in \mathcal{R}$ **do**
  Compute $\xi^-(\tilde{\beta}, R_0)$ and $\xi^+(\tilde{\beta}, R_0)$ by binary search on $R(\xi, \tilde{\beta}) = R_0$
end for
Store table $\{(\tilde{\beta}, R_0) \mapsto (\xi^-, \xi^+)\}$

Optimization (for each problem instance):
$(\beta^*, \Delta x^*, \Delta z^*) \leftarrow (\mathrm{NaN}, 0, 0)$
for all $(\tilde{\beta}, R_0)$ in the table **do**
  Compute $\beta_{\max}$, $\beta_{\min}$ from (7)
  if $\beta_{\min} > \beta_{\max}$ **then**
    continue
  end if
  $\beta \leftarrow \beta_{\min}$, $\quad \Delta x \leftarrow \tilde{\beta}/\beta$
  if $\lceil x_{\max}/\Delta x \rceil \cdot R_0 > \varepsilon$ **then**
    continue
  end if
  Compute $\Delta\xi_{\max}$ from (9); find Δz by binary search on (8)
  if $\Delta x \cdot \Delta z > \Delta x^* \cdot \Delta z^*$ **then**
    $(\beta^*, \Delta x^*, \Delta z^*) \leftarrow (\beta, \Delta x, \Delta z)$
  end if
end for
return β^*, Δx^*, Δz^*

5.1 Tropospheric Radio Wave Propagation

Consider $f = 3$ GHz ($\lambda = 0.1\,\mathrm{m}$), $c \approx 3 \times 10^8\,\mathrm{m/s}$, $k_{\min} \approx k_{\max}$. Table 2 shows the optimal grid parameters computed by the algorithm for Padé order [7/8] and $\varepsilon = 10^{-3}$.

Table 2. Optimal grid for tropospheric propagation ($f = 3$ GHz, $n \approx 1$, Padé [7/8], $\varepsilon = 10^{-3}$).

$\theta_{\max}$	$x_{\max}$	Δx (m)	Δz (m)	$\Delta x/\lambda$	$\Delta z/\lambda$
3°	100 km	111.49	0.179	1114.9	1.79
5°	10 km	31.91	0.061	319.1	0.61
10°	10 km	8.03	0.009	80.3	0.09
20°	1 km	1.67	0.004	16.7	0.04
30°	1 km	1.19	0.002	11.9	0.02
45°	1 km	0.38	0.0004	3.8	0.004

For small propagation angles ($\theta_{\max} \leq 5°$), typical of long-range ducted propagation, the algorithm permits very large steps ($\Delta x \sim 300$–1100λ). The grid becomes rapidly denser with increasing angle. Since $k_{\min} \approx k_{\max}$, the wavenumber variation has no measurable effect on the optimization.

5.2 Underwater Acoustics: Full Algorithm vs. Homogeneous Assumption

Consider $f = 500\,\text{Hz}$ ($\lambda = 3\,\text{m}$), $c_{\text{water}} = 1500\,\text{m/s}$, $c_{\text{sed}} = 1700\,\text{m/s}$, $x_{\max} = 10\,\text{km}$, $\varepsilon = 10^{-3}$. Table 3 compares the *full algorithm* (which uses $k_{\min} \neq k_{\max}$) with the *homogeneous assumption* ($k_{\min} = k_{\max}$).

Table 3. Optimal grid for UWA ($f = 500\,\text{Hz}$, $c_{\text{sed}} = 1700\,\text{m/s}$, Padé [7/8], $\varepsilon = 10^{-3}$, $x_{\max} = 10\,\text{km}$). The homogeneous assumption overestimates Δx by a factor of 2–40.

$\theta_{\max}$	Full algorithm		Homogeneous assumption	
	$\Delta x/\lambda$	$\Delta z/\lambda$	$\Delta x/\lambda$	$\Delta z/\lambda$
5°	11.9	1.430	478.4	1.667
10°	10.4	0.485	112.4	1.667
20°	7.2	0.157	16.7	0.202
30°	5.6	0.081	11.9	0.103

The results reveal a dramatic difference. At $\theta_{\max} = 5°$, the homogeneous assumption suggests $\Delta x = 478.4\lambda$, while the full algorithm yields $\Delta x = 11.9\lambda$, a factor of **40**. At $\theta_{\max} = 10°$, the discrepancy is still more than tenfold. At larger angles, the angular contribution to the ξ range begins to dominate, and the discrepancy narrows to a factor of 2.

Using the homogeneous-assumption grid would result in dispersion errors far exceeding the prescribed tolerance, because the rational approximation is evaluated outside its region of accuracy. This error is systematic and cannot be compensated by reducing Δz alone.

Table 4. Same as Table 3, but for $f = 100\,\text{Hz}$ ($\lambda = 15\,\text{m}$).

$\theta_{\max}$	Full algorithm		Homogeneous assumption	
	$\Delta x/\lambda$	$\Delta z/\lambda$	$\Delta x/\lambda$	$\Delta z/\lambda$
5°	15.3	0.333	478.4	0.333
10°	13.7	0.333	112.4	0.333
20°	8.9	0.333	16.9	0.333
30°	5.6	0.182	13.7	0.333

For comparison, Table 4 shows results at $f = 100\,\mathrm{Hz}$ ($\lambda = 15\,\mathrm{m}$). The transverse discretization is less restrictive at this frequency, but the Δx discrepancy remains equally large.

5.3 Field Distribution: Correct vs Incorrect Grid

To demonstrate the practical consequences of ignoring the refractive index inhomogeneity, we compute the acoustic field in a layered shallow-water waveguide at $f = 1500\,\mathrm{Hz}$ ($\lambda = 1\,\mathrm{m}$). The water depth is 200 m with $c_{\mathrm{water}} = 1500\,\mathrm{m/s}$. Below the water column, a 100 m upper sediment layer ($c = 1700\,\mathrm{m/s}$) overlies a 50 m lower sediment layer ($c_{\mathrm{sed}} = 1800\,\mathrm{m/s}$); both sediment layers have $\rho = 1.5\ \mathrm{g/cm^3}$ and absorption $0.05\,\mathrm{dB}/\lambda$. A narrow Gaussian beam (1° width) is placed at 50 m depth with an elevation angle of −45° (steeply downward), giving $\theta_{\max} = 46°$. The propagation distance is $x_{\max} = 1\,\mathrm{km}$ and $\varepsilon = 0.1$.

Three computations are performed using the JAX-based solver [16] with Padé order [7/8]: (i) a reference solution on a fine grid ($\Delta x = 0.78\lambda$, $\Delta z = 0.011\lambda$; 426×10338 output points, downsampled 3× from the computational grid), (ii) the full optimization algorithm accounting for $k_{\min} \neq k_{\max}$ ($\Delta x = 2.4\lambda$, $\Delta z = 0.034\lambda$; 427×10338 points), and (iii) the homogeneous assumption with $k_{\min} = k_{\max}$ ($\Delta x = 5.7\lambda$, $\Delta z = 0.048\lambda$; 177×7309 points). The homogeneous grid is **2.4 times coarser** along the longitudinal coordinate than the correct grid, because the optimizer does not account for the wider ξ range caused by the sediment.

Figure 3 shows the two-dimensional field distribution for all three cases. The reference and the full-optimization solutions are visually indistinguishable, confirming that the optimized grid preserves the accuracy. The homogeneous-assumption solution, however, exhibits visible distortion of the beam pattern: the interference structure differs, and the field in the sediment layers is noticeably degraded.

Figure 4 compares the depth profiles of the field amplitude at three ranges: 250, 500, and 900 m. At 250 m, the three solutions are still close. By 500 m, the homogeneous-assumption curve begins to deviate. At 900 m, after the error has accumulated over many propagation steps, the deviation is pronounced: the interference pattern is shifted, and the amplitude deviates by up to several dB.

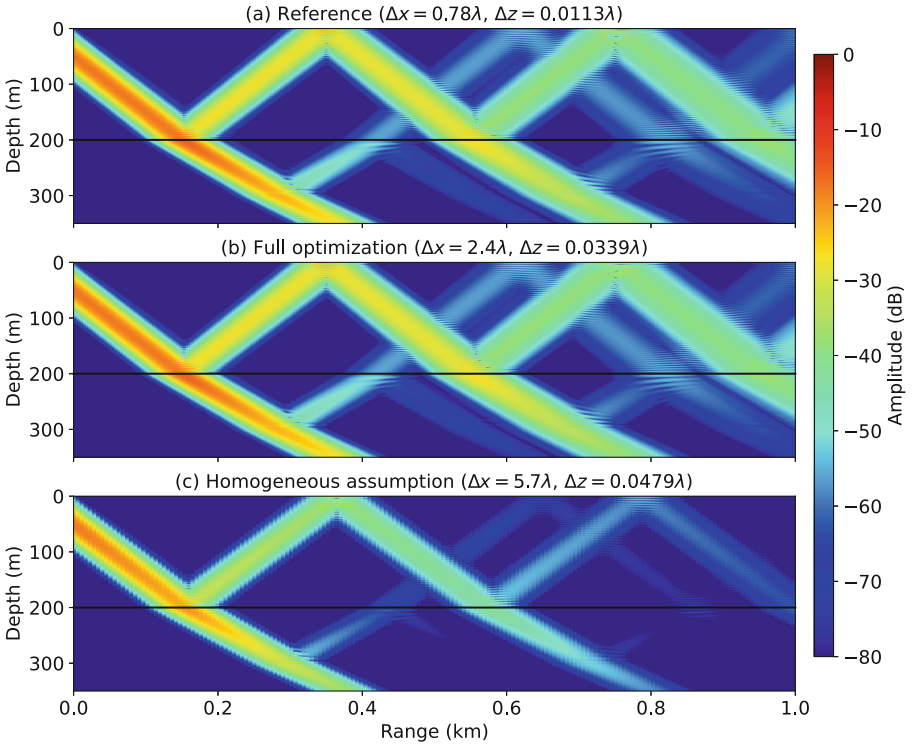


Fig. 3. Acoustic field amplitude (dB) in a layered waveguide ($f = 1500\,\mathrm{Hz}$, c_{sed} up to 1800 m/s, range 1 km, $-45°$ beam). (a) Reference (fine grid). (b) Full optimization. (c) Homogeneous assumption (incorrect grid, 2.4× too sparse). The black line marks the sea bottom.

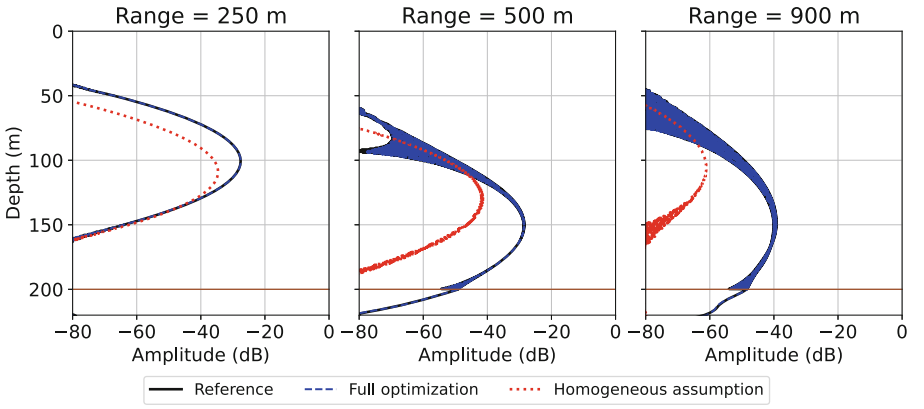


Fig. 4. Field amplitude vs. depth at ranges of 250, 500, and 900 m ($f = 1500\,\mathrm{Hz}$, c_{sed} up to 1800 m/s, Padé [7/8], $-45°$ beam). The full-optimization curve (dashed blue) matches the reference (solid black); the homogeneous-assumption curve (dotted red) progressively diverges with increasing range. (Color figure online)

The range dependence of the error is shown in Fig. 5, which plots the transmission loss at 150 m depth as a function of range. The full-optimization curve tracks the reference over the entire 1 km range. The homogeneous-assumption curve diverges progressively, confirming that the dispersion error accumulates with each propagation step.

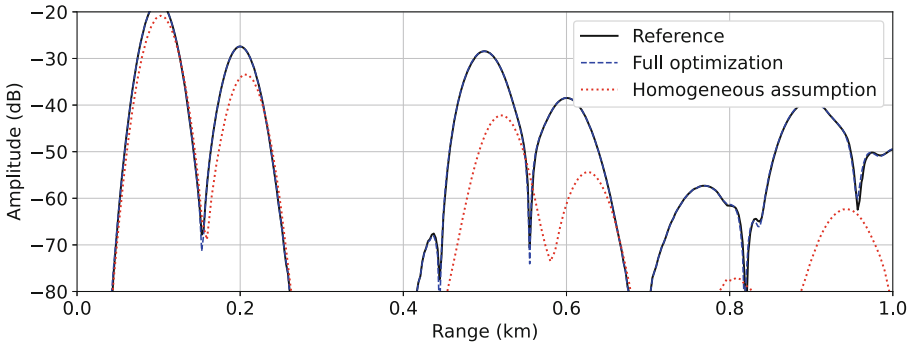


Fig. 5. Transmission loss at 150 m depth vs. range ($f = 1500$ Hz, c_{sed} up to 1800 m/s, Padé [7/8]). The homogeneous-assumption solution (dotted red) progressively diverges from the reference due to accumulated dispersion error. (Color figure online)

5.4 Effect of Sediment Sound Speed

Table 5 shows how the optimal grid depends on the sediment sound speed for $f = 500$ Hz, $\theta_{\max} = 10°$. As c_{sed} increases from 1500 m/s (homogeneous) to 2000 m/s, Δx decreases by a factor of 20. The sharpest drop occurs between 1500 and 1600 m/s, indicating that even a modest wavenumber contrast has a dramatic impact. This dependence is shown graphically in Fig. 6.

Table 5. Dependence of the optimal grid on c_{sed} ($f = 500$ Hz, $\theta_{\max} = 10°$, $x_{\max} = 10$ km).

c_{sed} (m/s)	$k_{\min}/k_{\max}$	$\Delta x/\lambda$	$\Delta z/\lambda$
1500 (homog.)	1.000	112.4	1.667
1600	0.938	16.7	0.557
1700	0.882	10.4	0.485
1800	0.833	7.2	0.435
1900	0.789	5.6	0.403
2000	0.750	5.6	0.397

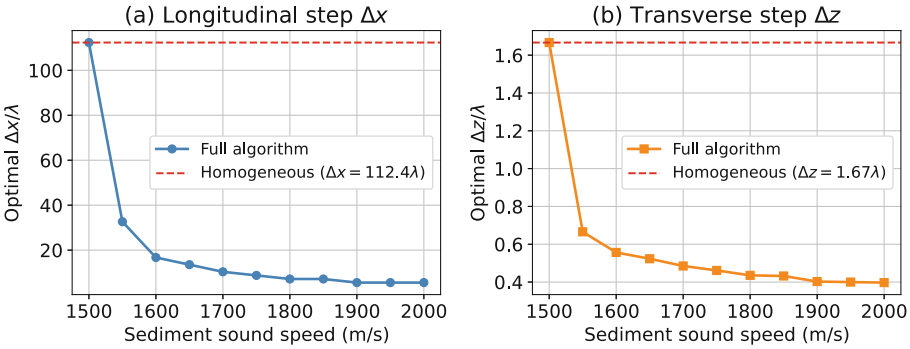


Fig. 6. Optimal grid steps vs. sediment sound speed ($f = 500\,\text{Hz}$, $\theta_{\max} = 10^\circ$, $x_{\max} = 10\,\text{km}$). The dashed line is the homogeneous assumption.

5.5 Optimal Δx as a Function of Propagation Angle

Figure 7 compares $\Delta x/\lambda$ as a function of $\theta_{\max}$ for the tropospheric and UWA cases. In both cases, Δx decreases with increasing angle, but the UWA curve lies significantly below the RWP curve due to the additional ξ-widening from the refractive index. The homogeneous assumption produces overly optimistic (too large) Δx values at all angles.

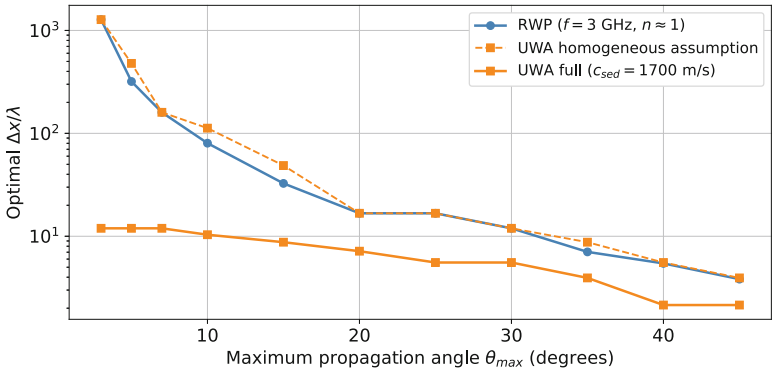


Fig. 7. Optimal $\Delta x/\lambda$ vs. $\theta_{\max}$ for RWP ($f = 3$ GHz) and UWA ($f = 500\,\text{Hz}$, $c_{\text{sed}} = 1700\,\text{m/s}$). Padé order [7/8], $\varepsilon = 10^{-3}$.

5.6 Comparison of Padé Orders

Table 6 shows the efficiency metric $V = p/(\Delta\,\mathrm{x} \cdot \Delta z)$ (lower is better, Δx and Δz in wavelengths) for the tropospheric case at various Padé orders and angles.

Table 6. Efficiency V for various Padé orders (tropospheric, $n \approx 1$, $\varepsilon = 10^{-3}$).

$\theta_{\max}$	$x_{\max}$	Order	$\Delta x/\lambda$	$\Delta z/\lambda$	V
3°	100 km	[1/2]	3.2	0.073	8.59
		[2/3]	31.9	0.231	4.08×10^{-1}
		[5/6]	477.8	0.810	1.55×10^{-2}
		[7/8]	1114.9	1.790	4.01×10^{-3}
10°	10 km	[2/3]	3.2	0.022	4.35×10^{1}
		[5/6]	48.1	0.075	1.66
		[6/7]	64.2	0.081	1.35
		[7/8]	80.3	0.085	1.17
20°	1 km	[2/3]	1.5	0.011	1.81×10^{2}
		[5/6]	14.8	0.036	1.14×10^{1}
		[7/8]	16.7	0.037	1.31×10^{1}

Table 7 presents the analogous comparison for UWA ($f = 500\,\mathrm{Hz}$, $c_{\mathrm{sed}} = 1700\,\mathrm{m/s}$, $\theta_{\max} = 10°$). Notably, orders [1/1] and [1/2] fail entirely in the UWA case, and the efficiency improves monotonically up to [7/8], suggesting that inhomogeneous environments benefit more from higher-order approximations. This is visualized in Fig. 8.

Table 7. Efficiency V for various Padé orders (UWA, $f = 500\,\mathrm{Hz}$, $c_{\mathrm{sed}} = 1700\,\mathrm{m/s}$, $\theta_{\max} = 10°$, $x_{\max} = 10\,\mathrm{km}$, $\varepsilon = 10^{-3}$).

Order	$\Delta x/\lambda$	$\Delta z/\lambda$	V
[2/3]	0.3	0.116	7.47×10^{1}
[3/4]	1.7	0.173	1.35×10^{1}
[4/5]	3.5	0.368	3.91
[5/6]	5.2	0.408	2.82
[6/7]	8.6	0.463	1.76
[7/8]	10.4	0.485	1.59

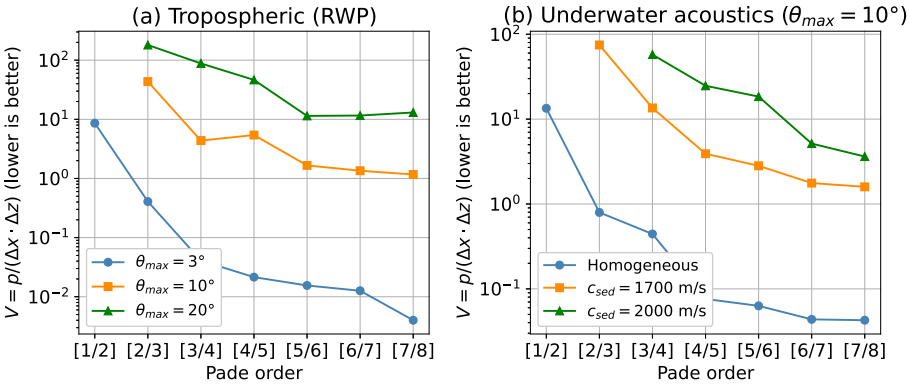


Fig. 8. Efficiency V vs. Padé order. (a) Tropospheric case at various angles. (b) UWA at $\theta_{\max} = 10°$ for different sediment speeds.

6 Conclusion

An algorithm for automated selection of optimal computational grid parameters for the two-dimensional parabolic equation in an inhomogeneous medium has been presented. The key results are as follows.

1. The range of the local wavenumber has been explicitly incorporated into the grid optimization procedure through the analysis of the variable ξ and its dependence on $k_{\min}$ and $k_{\max}$. The algorithm requires only these physical bounds; no artificial reference speed c_0 is needed. This extends the homogeneous-medium approach developed in [12,14].
2. For tropospheric radio wave propagation, where $k_{\min} \approx k_{\max}$, the wavenumber variation has negligible effect, and the optimization is governed by the propagation angle. The results are consistent with those obtained in [12,14].
3. For underwater acoustics, the wavenumber variation between water and sediment is the dominant factor. The homogeneous assumption leads to grids that are 10–40 times too sparse at small propagation angles, resulting in modeling errors far exceeding the prescribed tolerance.
4. A tabulation strategy enables real-time grid selection. The precomputed table depends only on the Padé order and can be reused across different problem instances.
5. Higher Padé orders are more beneficial in inhomogeneous environments: in the UWA case, the efficiency improves monotonically up to order [7/8], whereas in the tropospheric case, the optimal order is problem-dependent and may saturate at [5/6].

Future work includes the extension to three-dimensional inhomogeneous problems, range-dependent grid adaptation, and integration with inverse problem solvers where the wavenumber bounds are specified as an uncertainty interval.

Acknowledgments. This study was supported by the Russian Science Foundation grant No. 25-71-00028.

References

1. Baker, G.A., Graves-Morris, P.: Padé Approximants, 2nd edn. Cambridge University Press (1996)
2. Bekker, E.V., Sewell, P., Benson, T.M., Vukovic, A.: Wide-angle alternating-direction implicit finite-difference beam propagation method. J. Lightwave Technol. **27**(14), 2595–2604 (2009)
3. Bommidala, H., Colas, J., Emmanuelli, A., Dragna, D., Khodr, C., Cotté, B., Stevens, R.J.: Three-dimensional effects of the wake on wind turbine sound propagation using parabolic equation. J. Sound Vib. **608**, 119036 (2025)
4. Butcher, J.C.: Numerical Methods for Ordinary Differential Equations, 3rd edn. Wiley (2016)
5. Collins, M.D., Siegmann, W.L.: Parabolic Wave Equations with Applications. Springer, New York (2019). https://doi.org/10.1007/978-1-4939-9934-7
6. Collins, M.D.: A split-step padé solution for the parabolic equation method. J. Acoust. Soc. America **93**(4), 1736–1742 (1993)
7. Collins, M.D., Siegmann, W.L.: Parabolic equation techniques for seismology, seismo-acoustics, and arctic acoustics. J. Theoret. Comput. Acoust. **29**(02), 2130003 (2021)
8. Fishman, L., McCoy, J.J.: Derivation and application of extended parabolic wave theories. I. The factorized Helmholtz equation. J. Math. Phys. **25**(2), 285–296 (1984)
9. Jensen, F.B., Kuperman, W.A., Porter, M.B., Schmidt, H., Tolstoy, A.: Computational Ocean Acoustics, vol. 2011. Springer, New York (2011). https://doi.org/10.1007/978-1-4419-8678-8
10. Leontovich, M.A., Fock, V.A.: Solution of the problem of propagation of electromagnetic waves along the earth's surface by the method of parabolic equation. J. Phys. USSR **10**, 13–24 (1946)
11. Levy, M.F.: Parabolic Equation Methods for Electromagnetic Wave Propagation. IET (2000)
12. Lytaev, M.S.: Automated selection of the computational parameters for the higher-order parabolic equation numerical methods. In: Gervasi, O., et al. (eds.) ICCSA 2020. LNCS, vol. 12249, pp. 296–311. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58799-4_22
13. Lytaev, M.S.: Rational interpolation of the one-way Helmholtz propagator. J. Comput. Sci. **58**, 101536 (2022)
14. Lytaev, M.S.: Computational grid optimization for the 3D higher-order parabolic equation. In: Gervasi, O., et al. (eds.) Computational Science and Its Applications, ICCSA 2023. LNCS, pp. 170–185, vol. 13956. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-36805-9_12
15. Lytaev, M.S.: Mesh optimization for the acoustic parabolic equation. J. Mar. Sci. Eng. **11**(3), 496 (2023)
16. Lytaev, M.S.: PyWaveProp: Python library for wave propagation modeling (2026). https://github.com/mikelytaev/wave-propagation

17. Lytaev, M.S.: Numerov-Padé scheme for the one-way Helmholtz equation in tropospheric radio-wave propagation. IEEE Antennas Wirel. Propag. Lett. **19**(12), 2167–2171 (2020)
18. Yuldashev, P., et al.: Numerical models of nonlinear acoustic wave propagation in medical ultrasound problems and certain applications of aeroacoustics and underwater acoustics. Mosc. Univ. Phys. Bull. **80**(2), 195–225 (2025)

New Metaheuristic Algorithms to Reduce the Bandwidth of Real-World Graphs

Alexandre Augusto Alberto Moreira de Abreu[1,2,3(✉)] and Sanderson L. Gonzaga de Oliveira[1]

[1] Universidade Federal de São Paulo, São José dos Campos, SP, Brazil
{alexandre.abreu,sanderson.oliveira}@unifesp.br
[2] Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brazil
[3] Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, Canoinhas, SC, Brazil

Abstract. This paper evaluates new heuristics for the bandwidth reduction problem, two BRKGA-based algorithms, and NCHC+, an adaptation of the NCHC heuristic. We experimentally evaluated these heuristics against the most successful graph-theoretic algorithms and state-of-the-art metaheuristic algorithms using a benchmark test of large-scale graphs from the literature. The results confirm that FNCHC+ yields the best results for real-world graphs. Furthermore, the proposed heuristics outperformed DRSA, the previous state-of-the-art metaheuristic for small-sized graphs.

Keywords: Heuristic methods · Bandwidth reduction problem · Biased Random-Key Genetic Algorithm · Sparse graphs · Metaheuristics

1 Introduction

The bandwidth minimization problem is a prominent $\mathcal{NP}$-hard [14,43] combinatorial optimization problem with numerous scientific and engineering applications [7,33]. Garey et al. [13] demonstrated that the problem is $\mathcal{NP}$-hard for trees with maximum degree 3. Additionally, researchers have established particular classes of graphs in which the problem remains $\mathcal{NP}$-hard [44]. Harper [29] examined the problem in the context of error-correcting codes and formally defined it [30].

In an undirected graph, the goal is to minimize the maximum absolute difference between the labels of adjacent vertices. This task belongs to the broader class of graph layout problems, which optimize a specific objective function by assigning distinct natural numbers from the set $\{1, 2, \ldots, |V|\}$ to the vertices of a graph $G = (V, E)$.

To accelerate linear system solvers in sparse numerical linear algebra, practitioners typically employ low-cost graph-theoretic algorithms [21,25–28]. These methods relabel the vertices of a graph to improve the non-null patterns in

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 161–178, 2026.
https://doi.org/10.1007/978-3-032-30491-9_11

the associated matrix. Specifically, the Reverse Cuthill-McKee (RCM) [15] and Gibbs-Poole-Stockmeyer (GPS) [17] methods are the most successful graph-theoretic algorithms for this purpose. In contrast, this study targets domains lacking stringent runtime constraints, making them highly suitable for metaheuristic algorithms. Examples of such applications include VLSI design, network survivability, and efficient sparse graph representation [37], compression of large hypertext media [4], small-world networks [31], visual dataset analysis [42], graph entropy rate minimization [5], mesh layout optimization [8], symbolic model checking [2], and the seriation problem [9]. Particularly in big data, low-memory storage formats are essential for processing large-scale graphs, improving data locality, and avoiding inefficient computations [39].

Consequently, this paper focuses on metaheuristic algorithms for bandwidth reduction. For large-scale graphs (with orders exceeding one million and sizes exceeding millions), FNCHC+ [21,24] is the state-of-the-art metaheuristic algorithm [22]. Conversely, DRSA [50] is the leading metaheuristic algorithm for smaller graphs (of orders up to 1,000 and sizes up to 50,000). However, a previous study [22] indicated that DRSA relies on an outdated benchmark test, and this paper evaluates the heuristic in large-scale graphs.

This paper proposes two heuristics based on the BRKGA metaheuristic employing a new constructive method. The new BRKGA-based heuristics differ chiefly in the fitness functions adopted. The first uses bandwidth as the only fitness function. The second fitness function uses each individual's bandwidth and the number of critical vertices as a tiebreaker. Additionally, the paper evaluates an adaptation of the NCHC heuristic [36], named NCHC+.

This work experimentally evaluates three new heuristics for bandwidth reduction. We compare the proposed heuristics against state-of-the-art metaheuristic algorithms and the most successful graph-theoretic algorithms (RCM [15] and GPS [17]) using a benchmark test of large-scale graphs from the literature [22].

We organized the remainder of the paper as follows. Section 2 defines the problem, and Sect. 3 reviews the most important related work. Section 4 introduces the two novel BRKGA-based heuristics. Section 5 explains the new constructive method used in the BRKGA-based heuristics. Section 6 details the experiment, including an explanation of the NCHC+ heuristic in Sect. 6.4. Subsequently, Sect. 7 reports the computational results. Finally, Sect. 8 presents the conclusions.

2 Problem Definition

Gonzaga de Oliveira [22] defined the vertex labeling of an undirected graph, the bandwidth of a vertex, and the bandwidth of an undirected graph as follows.

Definition 1 (Vertex labeling of an undirected graph). *A vertex labeling L for an undirected graph $G = (V, E)$ is a bijective mapping $L : V \rightarrow \{1, 2, \ldots, |V|\}$. It assigns a unique integer identifier $l(v) \in [1, |V|]$ to each vertex $v \in V$, resulting in the set $L = \{l(v_1), l(v_2), \ldots, l(v_{|V|})\}$.*

Definition 2 (Bandwidth of a vertex in undirected graph). *Given a vertex labeling L, the bandwidth of a vertex v in $G_L = (V, E)$ is $\beta_v(G_L) = 0$ if $\nexists u \mid u \in Adj(G_L, v) \wedge l(u) < l(v)$; otherwise, $\beta_v(G_L) = \max_{u \in Adj(G_L,v) \wedge l(u) < l(v)} [l(v) - l(u)]$, where $Adj(G_L, v) = \{u : \{v, u\} \in E\}$ denotes the adjacent vertices of v.*

Definition 3 (Bandwidth in undirected graph). *The bandwidth of the undirected graph $G_L = (V, E)$ with a vertex labeling L is $\beta(G_L) = \max_{v \in V} [\beta_v(G_L)]$.*

The bandwidth minimization problem seeks to minimize the maximum absolute difference between the labels of adjacent vertices [22]. Figure 1 illustrates this reduction process on an undirected graph. Initially, vertices $v_1, \ldots, v_5$ are labeled as $l(v_i) = i$ for $i \in \{1, \ldots, 5\}$, resulting in $\beta(G_L) = 4$. A subsequent relabeling to $l(v_1) = 5, l(v_2) = 2, l(v_3) = 1, l(v_4) = 3$, and $l(v_5) = 4$ reduces the bandwidth to $\beta(G_L) = 2$.

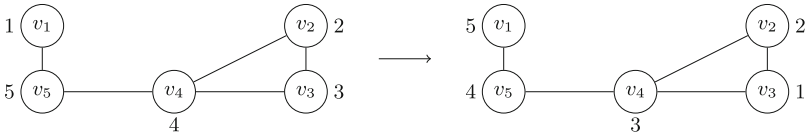


Fig. 1. Bandwidth reduction of an undirected graph from $\beta(G_L) = 4$ (left) to $\beta(G_L) = 2$ (right).

3 Related Work

Since the 1990s, researchers have developed numerous metaheuristic algorithms for bandwidth reduction [22]. Following a recent literature review [22], this section presents the most prominent and recent metaheuristic algorithms for this problem.

Dueck and Jeffs [11] proposed a Simulated Annealing heuristic. Esposito and Tarricone [12] introduced a heuristic based on Tabu Search. Martí et al. [40] proposed another Tabu Search (TS) algorithm that incorporates a frequency-based restart strategy for diversification alongside the tabu list. For vertex label swapping, the heuristic selects from a set of near-critical vertices defined as $C(L) = \{v : \beta_v(G_L) \geq \lambda \cdot \beta(G_L)\}$, where $\lambda \in (0, 1)$. Specifically, the heuristic considers v a critical vertex if its bandwidth is the maximum in the graph G_L, i.e., $\beta_v(G_L) = \beta(G_L)$. In 2001, the TS algorithm became the state-of-the-art metaheuristic algorithm for bandwidth reduction. However, the authors evaluated TS using small-sized graphs with up to 1,000 vertices and 50,000 edges. A recent review [22] classified these graphs as outdated and unrepresentative of modern problem scales and applications. Furthermore, modern computational capabilities make execution times on such small graphs indistinguishable. Consequently,

researchers must evaluate the heuristics on larger-scale graphs to verify whether a heuristic maintains its performance advantage as the problem size increases [32].

Piñana et al. [45] introduced a heuristic based on the Greedy Randomized Adaptive Search Procedures with Path Relinking (GRASP-PR) metaheuristic. The authors compared the bandwidth solutions on 31 graphs with up to 199 vertices and 80 graphs with up to approximately 1,000 vertices. A recent review [22] provided a detailed description of these graphs. The GRASP-PR heuristic emerged as the state-of-the-art method in 2004.

Lim et al. [37] proposed a Genetic Algorithm (GA) and the Node Shift (NS) heuristic, which incorporates a Hill Climbing procedure. After evaluating these heuristics on the benchmark test of Martí et al. [40,45] and 14 additional graphs, the NS algorithm yielded the best bandwidth solutions, becoming the state-of-the-art algorithm. Lim et al. [35] proposed a heuristic based on PSO.

The Node Centroid Hill Climbing (NCHC) and Fast Node Centroid Hill Climbing (FNCHC) heuristics were published in 2007 by Lim et al. [36]. The NCHC heuristic iteratively generates an initial solution via a Breadth-First Search starting from an arbitrary vertex. Subsequently, the heuristic improves the initial solution using a global optimization operator called Node Centroid (NC) and a Hill Climbing (HC) local search. The algorithm applies both the NC and HC procedures NC_{max} times, where NC_{max} is an algorithm parameter.

The NC procedure relabels all vertices toward a central position (centroid) relative to their neighborhood. For this purpose, it computes the metric $\omega(v) = \frac{\sum_{u \in B_\lambda(v)} l(u)}{|B_\lambda(v)|}$ for each vertex $v \in V$, where $B_\lambda(v) = (Adj(v) \cap \{u \in V : |l(u) - l(v)| \geq \lambda \cdot \beta(G_L)\}) \cup \{v\}$. Afterward, it relabels the vertices in non-decreasing order of their $\omega(v)$ values. The HC algorithm swaps labels of near-critical vertices using the candidate list of moves proposed by Martí et al. [40] to reach a local optimum. Similar to the GRASP-PR heuristic [45], NCHC considers the reduction in the number of critical vertices as the acceptance criterion for label swaps. Due to its high computational cost, the algorithm executes the HC procedure only during odd iterations. FNCHC is a faster variant of NCHC, optimized by fine-tuning the parameter λ that defines the near-critical vertex set. In computational tests on the benchmark test of Martí et al. [40,45], NS achieved the best bandwidth results, whereas FNCHC demonstrated lower running times.

Rodriguez-Tello et al. [46] introduced another Simulated Annealing heuristic (SA) in 2008, called SA–δ. Campos et al. [6] introduced a heuristic based on Scatter Search and Tabu Search.

Mladenović et al. [41] introduced the Variable Neighbourhood Search for bandwidth reduction (VNS-band). The heuristic became the state-of-the-art metaheuristic algorithm, achieving superior bandwidth reduction for graphs with orders up to 1,000 and sizes up to 50,000. The authors evaluated the algorithm using the benchmark test of Martí et al. [40,45]. In 2011, Koohestani and Poli [34] employed genetic programming to evolve the RCM algorithm. Although

analysts published several heuristics in subsequent years (e.g., [23]), VNS-band achieved superior results.

In 2015, Torres-Jimenez et al. [50] proposed the Dual Representation Simulated Annealing (DRSA) algorithm. When evaluated on the benchmark test composed of very small graphs from Martí et al. [40,45], DRSA outperformed VNS-band. Its superior results established it as the state-of-the-art metaheuristic algorithm on graphs with orders up to 1,000 and sizes up to 50,000 edges. While subsequent studies introduced other metaheuristic algorithms, such as a hybrid Iterated Local Search-Tabu Search approach [1], DRSA continued to yield superior results.

Silva et al. [47] introduced the first heuristic based on the Biased Random-Key Genetic Algorithm (BRKGA) framework [19] for bandwidth reduction. The authors used randomly generated individuals and the parameterized uniform crossover [48]. In each iteration, new solutions are randomly generated and introduced into the population to increase diversity. DRSA outperformed BRKGA [47] on 43 of the small-sized graphs from the benchmark test of Martí et al. [40,45]. The two novel BRKGA-based heuristics proposed in this paper, named BRKGA* and BRKGA+, differ from the BRKGA-based heuristic introduced by Silva et al. [47] mainly in four aspects: the constructive phase, the use of an encoder, the creation of new solutions, and the fitness function. Specifically, BRKGA* and BRKGA+ employ a novel randomized semi-greedy constructive algorithm (Sect. 5) to generate high-quality solutions at a low computational cost for both the initial population and the newly generated individuals. Afterward, a proposed encoder generates the random keys for these high-quality individuals. In contrast, Silva et al. [47] used randomly generated individuals for the initial population and new solutions, a process that does not require an encoder. Furthermore, while the BRKGA [47] and BRKGA+ heuristics both use the bandwidth value as their fitness function, BRKGA* employs a two-tiered fitness function that prioritizes the bandwidth value but uses the number of critical vertices as a tie-breaking criterion.

In 2022, Gonzaga de Oliveira and Carvalho [24] proposed the ILS-based heuristic for bandwidth reduction (ILS-band), the first metaheuristic algorithm to solve large-scale graphs (exceeding one million vertices) within an acceptable execution time of 1,200 s. Although ILS-band achieved bandwidth reductions comparable to GPS, the former state-of-the-art for large-scale graphs, it demanded longer execution times. In the same study, the authors introduced FNCHC+, a faster refinement of FNCHC [36] that restricts the invocation of the computationally expensive Hill Climbing local search to promising solutions. Computational tests on graphs ranging from 5,940 to 112,985 vertices demonstrated that FNCHC+ [24], GPS [18], FNCHC [36], and ILS-band [24] surpassed VNS-band [41] and DRSA [50]. Furthermore, comprehensive tests on the other 76 large-scale graphs (with orders up to 9,845,725 and sizes up to 57,156,537) established FNCHC+ as the state-of-the-art metaheuristic algorithm for large-scale graphs, despite GPS initially exhibiting shorter running times. Lastly, a subsequent study [21] used 32 large-scale graphs (with orders up to 23,947,347

and sizes up to 130,641,318) and confirmed that FNCHC+ consistently outperforms GPS in both solution quality and computational cost.

In short, DRSA [50] is the state-of-the-art metaheuristic algorithm for reducing the bandwidth of small graphs (i.e., with orders of up to 1,000 and sizes of up to 50,000). FNCHC+ heuristic [21,24] is the state-of-the-art metaheuristic algorithm for large-scale graphs.

4 Two Novel Heuristics Based on the Biased Random-Key Genetic Algorithm Metaheuristic

This section introduces two new bandwidth reduction heuristics based on the Biased Random-Key Genetic Algorithm (BRKGA) metaheuristic [19]. BRKGA employs arrays of real numbers, named random-key arrays. These arrays represent possible solutions to a given problem. A deterministic decoding algorithm associates a random-key array with a feasible solution. It allows the algorithm to calculate the objective function (fitness) for the problem. The method is biased because one of the parents in BRKGA is always selected from the elite group and has a higher probability of passing on its genes.

Algorithm 1 receives the graph $G(V, E)$, the parameters $|P| > 0$, $e > 0$, $r > 0$, $init \in \{0, 1\}$, $\alpha \in [0, 1]$, $p > 0.5$, and the maximum execution time t_{max}. The parameters $|P|$, e, and r define, respectively, the population size, the size of the elite solution set, and the size of the new solution set. The parameters $init$ and α determine the initial vertex and the level of randomness considered in the constructive method (Algorithm 2). The parameter p represents the probability that a child receives genes from a parent selected from the elite set.

Algorithm 1: The pseudocode of two heuristics based on the Biased Random-Key Genetic Algorithm metaheuristic (BRKGA+ and BRKGA*).

Input: graph $G(V, E)$, parameters $|P|$, e, r, α, $init$, p, and maximum runtime t_{max}.
Output: the bandwidth of the best solution found.
Generate an initial population P (Algorithm 2), with parameters $init$ and α;
Encode the individuals of the population P;
Sort the population P;
while (execution time is less than t_{max}) **do**
    Copy the best e individuals from P into P';
    for ($i = 1$ **to** ($|P| - e - r$)) **do**
        Select a random parent from the elite set of P;
        Select a random parent from the population P;
        Generate a single new individual by parameterized uniform crossover with probability p;
        Decode the new individual;
        Insert the new individual into the population P';
    end for
    Insert r new individuals into P' (Algorithm 2), with parameters $init$ and α;
    Encode the r new individuals inserted into the population P';
    Sort the population P';
    $P \leftarrow P'$; `// the new population` P' `becomes population` P
end while
return the bandwidth of the best individual in P;

We implemented two variations of Algorithm 1. We called the first variation BRKGA+. It uses only each individual's bandwidth as a fitness function. The fitness function of the second variation, called BRKGA*, is the bandwidth value of each individual. However, for individuals with the same bandwidth, BRKGA* uses the number of critical vertices as a tiebreaker. This approach is the first difference between Algorithm 1 and the BRKGA heuristic proposed by Silva et al. [47].

In line 1 of Algorithm 1, an initial population P of size $|P|$ is generated by Algorithm 2, presented in Sect. 5, with parameters *init* and α. This new initial population is the second difference between this approach and the approach proposed by Silva et al. [47]. Specifically, Silva et al. [47] employed a set of random solutions as the initial population.

Line 2 encodes the generated individuals (solutions). An individual is a labeling L in an array of random keys. The encoder assigns the value $l(i)/|V|$, where $i = 1, 2, \cdots, |V|$, to each entry i of the random-key array. This approach is the third difference between Algorithm 1 and the BRKGA heuristic [47]. In the BRKGA heuristic [47], individuals' random keys in the initial population are generated randomly in the interval $[0, 1)$. Therefore, BRKGA [47] does not use an encoder that translates a solution (labeling) to the domain of random keys. Line 3 sorts the generated individuals in ascending order of their fitness values.

Algorithm 1 executes lines from 5 to 16 while the execution time is less than t_{max}. Line 5 copies the best e individuals from population P to the next population P'. This approach ensures the survival of the best e individuals.

Algorithm 1 executes lines 7–11 $|P| - e - r$ times. Line 7 randomly selects a parent from the elite set. Line 8 randomly selects another parent from the population P. Line 9 generates a single offspring from the selected parents. The parameterized uniform crossover [48] performs the crossover operation. For each entry in the array of random keys (gene), the algorithm randomly decides from which parent to copy the key value to the offspring. However, it is a biased decision because the algorithm defines a higher probability $p > 0.5$ of selection for the parent from the elite group. This bias encourages the propagation of characteristics from the most promising solutions throughout the evolutionary process.

In line 10, the decoder decodes the new individual. Line 11 inserts it into the population P'. In this study, the decoder sorts the random keys in ascending order and assigns the index corresponding to the original entry of the i-th key before sorting to the label of vertex i, where $i = 1, 2, \ldots, |V|$. Thus, the resulting array is a permutation array corresponding to a labeling for the graph's vertices. This decoder is the simple decoder proposed by Bean [3] and widely adopted in combinatorial optimization problems.

In line 13, the constructive method presented in Algorithm 2 with parameters *init* and α generates a set of r new solutions. This step aims to bring diversity to the population. This approach is the fourth difference with respect to the BRKGA proposed by Silva et al. [47], which generates the new solutions randomly. The encoder encodes the generated r new individuals in line 14. Line

15 sorts the population P' by the value of the fitness function, and it becomes the current population P in line 16. Finally, in line 18, the bandwidth of the best individual of the population P is returned.

5 Constructive Method

Algorithm 2 details the semi-greedy randomized constructive algorithm that the new BRKGA-based heuristics use to create high-quality individuals at low computational cost. Algorithm 1 employs this procedure in lines 1 and 13. The proposed constructive algorithm is a variant of the RCM [15] and the George-Liu [16] algorithms. In addition to the graph $G = (V, E)$, the algorithm requires the parameters *init* and α, which determine the initial vertex and the degree of randomness, respectively.

Algorithm 2: Randomized Semi-Greedy Constructive Method.

```
   Input: graph G = (V, E), parameters init and α.
   Output: labeling L.
 1 Q ← ∅;   visited ← [false, ..., false];
 2 if (init = 0) then  v ← G.RandomVertex();
 3 else  v ← G.GL(G.RandomVertex());
 4 Q.add(v);   visited[v] ← true;
 5 l(v) ← 1;   i ← 2;
 6 while (Q ≠ ∅) do
 7 |   u ← Q.remove();
 8 |   foreach (vertex w ∈ Adj(G, u), sorted by degree in a randomized semi-greedy order) do
 9 |   |   if (¬visited[w]) then
10 |   |   |   Q.add(w);   visited[w] ← true;
11 |   |   |   l(w) ← i;   i ← i + 1;
12 |   |   end if
13 |   end foreach
14 end while
15 return L;
```

In line 1, the constructive algorithm initializes a vertex queue Q and a boolean array *visited* used in its breadth-first search strategy. Depending on the parameter *init*, the procedure either randomly selects a starting vertex $v \in V$ (line 2) or determines a pseudoperipheral vertex v via the George-Liu algorithm [16] (line 3). Subsequently, line 4 enqueues v into Q and marks it as visited. Line 5 then assigns the initial label 1 to v and sets the next available label counter i to 2.

Algorithm 2 iterates through lines 7–13 until the queue Q becomes empty. In line 7, the procedure dequeues the first element and assigns it to u. Line 8 sorts the neighborhood of u in a semi-greedy randomized order based on vertex degrees. Specifically, the algorithm first sorts the adjacent vertices by degree. Then it applies a random shuffle to the initial $\lfloor |Adj(G, u)| \cdot \alpha \rfloor$ elements of the resulting ordered list. Consequently, setting $\alpha = 1$ results in a completely random traversal of the adjacent vertices. If $\alpha = 0.5$, the algorithm randomly shuffles

the first half of the sorted neighbors. To prevent greedy ordering when $\alpha = 0$, the algorithm assigns a random value in the range $(0, 1)$ to α at each iteration.

Following this partial sorting process, line 9 evaluates whether each adjacent vertex w remains unvisited. For every unvisited vertex w, line 10 enqueues it into Q and marks it as visited. Then, line 11 assigns the current label i to w and increments i by one. Finally, line 15 returns the labeling L.

6 Description of the Tests

This study conducted the experiments on a machine equipped with an Intel® Xeon® Gold 6330 processor and 32 GiB of RAM (DIMM DDR4 3200 MT/s). The operating system was Ubuntu 24.04 LTS (Linux kernel 6.8.0-45-generic).

The BRKGA, BRKGA+, BRKGA*, RCM, and DRSA heuristics employed the Compressed Sparse Row (CSR) format [49]. This study implemented all eight algorithms in C++ and compiled using g++ 13.2.0 with the -O3 optimization flag.

This study evaluates the algorithms in a benchmark test comprising 23 graphs from the literature [22]. These graphs originate from real-world applications of metaheuristic algorithms for bandwidth reduction. Moreover, these graphs are publicly available in the SuiteSparse Matrix Collection [10].

We employed Irace [38] to configure the parameters automatically of all evaluated metaheuristic algorithms. This tool efficiently explores the parameter space to identify satisfactory parameter configurations. Irace executes iterated runs, discarding inferior configurations based on statistical tests. To avoid premature convergence, the tool incorporates a restart mechanism.

We kept most Irace settings at their default values, modifying only the statistical elimination test (testType) and the experiment budget (maxExperiments). We selected the Friedman test and fixed the maximum number of experiments at 200. This value represents the minimum budget recommended by Irace to parameterize the DRSA heuristic. We executed Irace independently for each of the 23 benchmark graphs.

Defining an acceptable timeout for metaheuristics is critical. Extended execution times are often impractical. Following an earlier work [24], we established a maximum execution time of 1,200 s as the only stopping criterion for all algorithms evaluated in this study.

We evaluated the best bandwidth and the average of four solutions per graph. This count corresponds to the minimum number of executions performed by Irace under the used configuration. The subsequent sections detail the parameter tuning for each metaheuristic. Section 6.1 addresses the BRKGA heuristic [47], while Sect. 6.2 introduces the proposed BRKGA* and BRKGA+ heuristics. Section 6.3 presents the parameters configured by Irace for DRSA [50], and Sect. 6.4 details the parameterization of NCHC+ (firstly used in the present study) and FNCHC+ [24].

6.1 BRKGA [47]

The parameters configured by Irace for the BRKGA heuristic [47] were $|P| \in [50, 1000]$ (integer), $e \in [10, 400]$ (integer), $r \in [5, 200]$ (integer), and $p \in [0.55, 0.99]$ (real). Furthermore, we configured Irace to prevent parameters that violate the following rules: $e > 0.4 \cdot |P|$ and $r > 0.2 \cdot |P|$.

We defined the range of the parameter $|P|$ based on the population size reported by Silva et al. [47], and after preliminary tests. Silva et al. [47] reported the parameters $|P| = 1000$, $e = 500$, and $r = 500$. Thus, the population incorporates only individuals belonging to the elite set and newly generated individuals in each new generation in the BRKGA heuristic [47]. Therefore, we defined bounds for the parameters e and r to ensure the existence of descendants produced by crossover operations in which at least one parent belongs to neither the elite set nor the newly generated individuals from the constructive method. Furthermore, we defined the ranges for the parameters e, r, and p based on values recommended by Gonçalves and Resende [20], as well as on preliminary tests. Table 1 shows the best parameter configurations found by Irace for the BRKGA heuristic [47]. We have abbreviated the names of graphs Fashion_MNIST_norm_10NN, kmnist_norm_10NN, MISKnowledgeMap, k49_norm_10NN, preferentialAttachment, vsp_barth5_1Ksep_50in_5Kout, vsp_befref_fxm_2_4_air02, vsp_bump2_e18_aa01_model1_crew1, vsp_c-30_data_data, vsp_data_and_seymourl, vsp_finan512_scagr7-2c_rlfddd, vsp_model1_crew1_cr42_south31, vsp_msc10848_300sep_100in_1Kout, vsp_p0291_seymourl_iiasa, vsp_sctap1-2b_and_seymourl, vsp_south31_slptsk, and vsp_vibrobox_scagr7-2c_rlfddd in the subsequent tables.

6.2 BRKGA+ and BRKGA*

The parameters configured by Irace for the BRKGA+ and BRKGA* heuristics were $\alpha \in [0.0, 1.0]$ (real), $init \in \{0, 1\}$ (categorical), $|P| \in [10, 100]$ (integer), $e \in [2, 40]$ (integer), $r \in [1, 10]$ (integer), and $p \in [0.55, 0.99]$ (real). Preliminary tests showed that populations of size $|P| > 100$ yielded unpromising results due to the computational cost associated with generating and maintaining a larger population.

We defined the ranges for the parameters e, r, and p based on values recommended by Gonçalves and Resende [20], as well as on preliminary tests. Table 1 shows the best parameter settings found by Irace for the BRKGA+ and BRKGA* heuristics.

6.3 DRSA [50]

We implemented DRSA in C++ strictly following the original description by Torres-Jimenez et al. [50]. Because this algorithm is a direct application of the Simulated Annealing metaheuristic, its implementation is very straightforward. We configured the Irace to tune the DRSA parameters within the following ranges: $T_0 \in [500, 10000]$ (integer), $T_f \in [1 \times 10^{-8}, 0.1]$ (real), $\alpha \in [0.9, 0.99]$

Table 1. The best parameter settings found by Irace for the BRKGA [47], BRKGA+, and BRKGA* heuristics in 23 graphs.

Graph	BRKGA [47]				BRKGA+						BRKGA*					
	$\|P\|$	e	r	p	$\|P\|$	e	r	α	$init$	p	$\|P\|$	e	r	α	$init$	p
boyd1	793	271	54	0.60	70	20	5	0.94	1	0.77	75	17	6	0.96	1	0.70
Fashion_MNIST	930	45	152	0.75	66	3	8	0.90	0	0.57	66	5	3	0.85	0	0.70
har_10NN	647	71	107	0.82	64	17	4	0.81	1	0.75	52	15	10	0.91	1	0.98
k49_norm	927	18	140	0.66	86	28	7	0.61	1	0.87	67	20	9	0.48	1	0.74
kmnist_norm	887	97	80	0.80	77	6	9	0.85	0	0.59	94	13	8	0.82	0	0.91
lp1	706	71	43	0.96	33	7	7	0.28	1	0.63	97	38	9	0.85	0	0.96
MISKnowledge	959	32	91	0.55	71	23	6	0.93	0	0.73	71	25	4	0.31	0	0.64
power	849	91	146	0.94	62	10	6	0.85	0	0.75	56	22	5	0.87	0	0.70
preferentialAtt	476	46	69	0.57	74	8	10	0.99	1	0.79	20	3	4	0.69	0	0.70
smallworld	713	72	109	0.86	57	19	9	0.66	0	0.89	76	21	8	0.59	1	0.78
vsp_barth5	761	50	99	0.84	76	24	7	0.36	0	0.69	44	5	4	0.60	0	0.74
vsp_befref	904	141	137	0.95	45	15	5	0.65	0	0.74	60	22	7	0.70	0	0.64
vsp_bump2	661	33	79	0.80	59	18	10	0.34	0	0.92	87	14	10	0.16	0	0.70
vsp_c-30	973	74	167	0.73	100	30	8	0.87	0	0.58	79	19	3	0.89	0	0.70
vsp_data	743	166	139	0.96	76	7	8	0.53	0	0.66	83	18	8	0.58	0	0.97
vsp_finan512	662	262	34	0.72	32	5	7	0.55	0	0.69	69	19	10	0.45	0	0.70
vsp_model1	808	21	93	0.63	75	15	6	0.26	0	0.62	46	2	7	0.37	0	0.64
vsp_msc10848	438	27	67	0.92	60	12	7	0.31	0	0.86	68	24	7	0.09	0	0.57
vsp_p0291	888	199	96	0.87	87	6	5	0.27	0	0.97	87	9	7	0.29	0	0.98
vsp_sctap1-2b	831	26	112	0.80	42	2	7	0.75	0	0.84	74	20	8	0.31	0	0.79
vsp_south31	923	133	166	0.80	63	18	6	0.34	0	0.71	45	3	5	0.37	0	0.75
vsp_vibrobox	635	39	35	0.84	92	14	10	0.01	0	0.99	43	9	6	0.14	0	0.66
worms20_10NN	982	313	104	0.57	61	8	10	0.87	0	0.61	87	14	7	0.47	0	0.91

(real), $L \in [30, 50]$ (integer), $L_f \in [1, 20]$ (integer), $P_{rex} \in [0.0, 0.7]$ (real), and $P_{nex} \in [P_{rex}, 0.9]$ (real). While the original authors [50] fixed $L_f = 10 \cdot |V| \cdot |E|$, our study scales the L_f parameter by multiplying it by $|V| \cdot |E|$ before executing the heuristic. Consequently, providing the interval $[1, 20]$ allows Irace to explore L_f values both substantially lower and higher than the original baseline. We applied this same logic to establish the intervals for all other DRSA parameters. Furthermore, Sect. 6.4 details how we adopted an identical approach to configure the parameters for the NCHC+ and FNCHC+ algorithms. Finally, Table 2 presents the best parameter configuration found by Irace for the DRSA heuristic [50].

Table 2. The best parameter settings found by Irace for the DRSA heuristic [50] in 23 graphs.

Graph	α	L	L_f	P_{nex}	P_{rex}	T_0	T_f
boyd1	0.95	44	4	0.52	0.23	1680	3×10^{-2}
Fashion_MNIST	0.91	34	11	0.11	0.69	1412	4×10^{-2}
har_10NN	0.95	33	20	0.17	0.26	6692	1×10^{-8}
k49_norm	0.98	33	7	0.35	0.51	3424	1×10^{-8}
kmnist_norm	0.95	38	16	0.29	0.60	1083	2×10^{-2}
lp1	0.92	30	2	0.13	0.63	3819	5×10^{-2}
MISKnowledge	0.92	30	11	0.19	0.40	4018	2×10^{-2}
power	0.94	34	20	0.49	0.16	1409	1×10^{-8}
preferentialAtt	0.93	34	15	0.60	0.25	3130	1×10^{-2}
smallworld	0.98	44	2	0.01	0.63	3695	6×10^{-2}
vsp_barth5	0.90	47	1	0.04	0.46	825	2×10^{-2}
vsp_befref	0.95	36	10	0.12	0.57	6412	1×10^{-2}
vsp_bump2	0.96	31	5	0.65	0.14	670	1×10^{-8}
vsp_c-30	0.96	44	13	0.40	0.45	3175	1×10^{-8}
vsp_data	0.94	35	14	0.29	0.28	7125	1×10^{-8}
vsp_finan512	0.93	30	12	0.16	0.58	6636	1×10^{-8}
vsp_model1	0.94	46	16	0.15	0.53	7660	1×10^{-2}
vsp_msc10848	0.98	32	5	0.34	0.41	2683	1×10^{-8}
vsp_p0291	0.93	32	8	0.52	0.03	6133	1×10^{-8}
vsp_sctap1-2b	0.93	30	8	0.17	0.36	5594	5×10^{-2}
vsp_south31	0.92	32	7	0.16	0.54	4015	1×10^{-8}
vsp_vibrobox	0.95	39	3	0.08	0.56	2386	1×10^{-8}
worms20_10NN	0.92	50	9	0.11	0.58	612	7×10^{-2}

6.4 NCHC+ and FNCHC+ [21,24]

The NCHC+ algorithm is a variation of the NCHC heuristic [36]. NCHC used a single value for λ. Unlike NCHC, NCHC+ varies the parameter λ, such as in the FNCHC+ heuristic [21,24]. NCHC+ also uses the condition with κ [21,24] to restrict calls to the Hill Climbing (HC) local search, which has a high execution cost. Thus, NCHC+ is feasible for large-scale graphs in a reasonable runtime [21,24]. The present study applies the NCHC+ heuristic for the first time.

Irace configured the following parameters of the NCHC+ and FNCHC+ heuristics: $\kappa \in [1, 200]$ (integer) and $NC_{max} \in [1, 200]$ (integer). Additionally, Irace configured $\lambda \in [0.85, 1.0]$ (real) for the NCHC+ heuristic. We defined the ranges used for the parameters κ and NC_{max} (NCHC+ and FNCHC+) based on the parameters reported by Gonzaga de Oliveira and Carvalho [24], and on

preliminary tests. We defined the range used for the parameter λ based on the results reported by Lim et al. [36] and after preliminary tests.

Table 3 shows the best parameter settings found by Irace for the NCHC+ and FNCHC+ heuristics. The present study employed the FNCHC+ implementation of Gonzaga de Oliveira and Carvalho [24]. This study modified this implementation to use the NCHC+ heuristic. Table 3 shows that the 23 graphs used in this study have dimensions ranging from 2,427 to 534,388 rows and columns and 13,188 to 2,442,056 non-zero elements.

Table 3. The best parameter settings found by Irace for the NCHC+ heuristic and FNCHC+ [24] in 23 graphs.

Graph	$\|V\|$	$\|E\|$	NCHC+			FNCHC+	
			λ	NC_{max}	κ	κ	NC_{max}
boyd1	93,279	1,211,231	0.88	137	110	66	80
Fashion_MNIST	10,000	158,304	0.88	15	124	131	93
har_10NN	10,299	151,736	0.99	63	176	130	51
k49_norm	38,547	618,158	1.00	30	42	151	153
kmnist_norm	10,000	156,932	0.99	19	181	123	184
lp1	534,388	11,643,420	0.89	138	111	92	17
MISKnowledge	2,427	57,022	0.89	78	113	73	185
power	4,941	13,188	0.89	185	178	75	198
preferentialAtt	100,000	999,970	1.00	92	149	90	196
smallworld	100,000	999,996	1.00	161	110	113	194
vsp_barth5	32,212	203,610	1.00	43	121	108	85
vsp_befref	14,109	196,448	1.00	91	165	156	188
vsp_bump2	56,438	601,602	1.00	31	190	192	66
vsp_c-30	11,023	124,368	1.00	177	90	181	193
vsp_data	9,167	111,732	1.00	60	96	178	137
vsp_finan512	139,752	1,104,040	1.00	62	108	50	164
vsp_model1	45,101	379,952	1.00	131	198	88	200
vsp_msc10848	21,996	2,442,056	1.00	41	172	146	21
vsp_p0291	10,498	107,736	0.99	178	99	41	99
vsp_sctap1-2b	40,174	281,662	1.00	64	59	130	114
vsp_south31	39,668	379,828	1.00	175	196	188	110
vsp_vibrobox	77,328	871,172	0.97	11	38	169	191
worms20_10NN	20,055	240,826	0.89	4	171	120	181

7 Results and Analysis

This section evaluates the bandwidth results of the BRKGA+, BRKGA*, and NCHC+ heuristics against the GPS [17], RCM [15], DRSA [50], BRKGA [47], and FNCHC+ [21,24] heuristics. Table 4 shows the bandwidth results of the eight evaluated heuristics.

Table 4 displays the number of best results and the Sum of Relative Deviations (SRD) over all graphs. The Relative Deviation (RD) for a given graph, calculated as $\frac{\beta(G_L)-\beta(G_{L^*})}{\beta(G_{L^*})}$, represents the extent to which the bandwidth of a solution L deviates from the bandwidth of the best solution L^* found. The GPS algorithm failed to solve within the 1,200 s time limit when evaluated on the k49_norm_10NN, lp1, and vsp_vibrobox graphs. Consequently, we used the initial bandwidth value of these graphs to calculate the RD.

Table 4. The bandwidth results from 8 heuristics applied to 23 graphs [22].

Graph	FNCHC+		NCHC+		BRKGA*		BRKGA+		RCM	GPS	DRSA		BRKGA	
	best	avg.	best	avg	best	avg.	best	avg.			best	avg.	best	avg.
boyd1	**46634**	**46634**	46636	46636	48227	77045	87936	88855	89508	93276	46901	46919	48072	48443
Fashion_MNIST	**1422**	**1422**	1541	1541	1578	1602	1571	1591	1890	5138	9698	9717	9787	9794
har_10NN	**942**	**942**	1101	1101	1122	1126	1123	1128	1804	3434	8811	9014	10076	10105
k49_norm	**11668**	**11668**	11913	11913	12827	12945	13040	13080	16783	†	37935	37949	38170	38196
kmnist_norm	**2271**	**2271**	2323	2323	2988	2992	2988	2993	3660	7561	9825	9841	9799	9805
lp1	**284546**	**284546**	284641	284641	284550	284550	284645	284645	284744	†	532668	532766	530265	530499
MISKnowledge	**684**	**684**	934	934	1031	1031	1016	1018	1346	1994	2339	2351	2333	2336
power	230	230	**196**	**196**	272	272	272	272	294	1088	513	563	4490	4531
preferentialAtt	**50849**	**50849**	61877	61877	73427	73439	73339	73448	73802	99962	99165	99414	99378	99427
smallworld	**44502**	**44502**	45521	45521	53691	53702	53691	53719	54556	96435	99623	99642	99281	99303
vsp_barth5	**8984**	**8984**	**8984**	**8984**	11778	11783	11779	11789	12365	30232	31582	31621	31708	31732
vsp_befref	**3431**	**3431**	3546	3546	5647	5655	5647	5647	9924	13736	13730	13755	13613	13690
vsp_bump2	**17316**	**17316**	20453	20453	23916	23966	23898	23942	35336	53706	55516	55569	55839	55878
vsp_c-30	**2284**	**2284**	2300	2300	3676	3679	3687	3693	4207	9770	8365	8465	10676	10700
vsp_data	**3149**	**3149**	**3149**	**3149**	4449	4455	4459	4480	5367	9022	6390	6525	8892	8917
vsp_finan512	**39676**	**39676**	40239	40239	43712	44292	44302	44385	68124	128618	139027	139129	138159	138159
vsp_model1	**14315**	**14315**	14509	14509	17631	17631	17631	17631	21267	39133	44469	44540	44266	44316
vsp_msc10848	**6706**	**6706**	8173	8173	8221	8275	8221	8277	14817	21396	21813	21843	21887	21893
vsp_p0291	3987	3987	**3796**	**3796**	4420	4421	4299	4392	6620	9741	8312	8432	10166	10185
vsp_sctap1-2b	**12935**	**12935**	**12935**	**12935**	17426	17734	17418	17461	18991	38849	39663	39706	39588	39616
vsp_south31	**15749**	**15749**	**15749**	**15749**	17631	17632	17631	17632	20321	38106	38082	38306	38946	38972
vsp_vibrobox	**31685**	**31685**	45908	45908	43725	44096	43727	44146	49677	†	76530	76621	76686	76723
worms20_10NN	**699**	**699**	748	748	744	744	741	743	1024	2939	19751	19769	18501	18501
SRD	**0.2**	**0.2**	1.9	1.9	6.3	7.0	7.1	7.2	15.2	48.8	78.5	79.2	100.7	101.0
No. of bests results	**21**	**21**	6	6	0	0	0	0	0	0	0	0	0	0

Table 4 shows that FNCHC+ yielded 21 of the best overall bandwidth results. The new heuristic NCHC+ returned six of the best results. FNCHC+ (NCHC+) achieved a Sum of Relative Deviations (SRD) of 0.2 (1.9) for the best and average results across all 23 graphs. In contrast, the other six heuristics failed to yield competitive solutions.

BRKGA* and BRKGA+ outperformed GPS [17], RCM [15], DRSA [50], and BRKGA [47]. BRKGA* produced slightly better results than BRKGA+ because the former employs the number of critical vertices as a tiebreaker criterion (see Sect. 4). The SRD values achieved by BRKGA* (BRKGA+) for the best and average solutions were 6.3 (7.1) and 7.0 (7.2), respectively. The RCM, GPS, DRSA, and BRKGA heuristics yielded considerably worse SRD values.

Table 4 shows that BRKGA [47] achieved a considerably worse SRD than DRSA [50]. Additionally, DRSA yielded 14 better results than BRKGA, while BRKGA returned 9 better results than DRSA. Conversely, FNCHC+ [21,24], NCHC+, BRKGA*, BRKGA+, and RCM [15] outperformed DRSA.

8 Summary and Conclusions

This work introduced novel heuristics for the Bandwidth Reduction Problem: two based on the Biased Random-Key Genetic Algorithm, BRKGA* and BRKGA+, and an adaptation of the NCHC algorithm [36], NCHC+. We assessed the performance of these heuristics against classical and state-of-the-art algorithms on a benchmark of real-world applications composed of large-scale graphs from the literature [22].

Experimental results demonstrated that BRKGA* and BRKGA+ effectively outperformed DRSA [50], the former state-of-the-art metaheuristic for small-sized graphs. However, FNCHC+ [21,24] and NCHC+ outperformed BRKGA* and BRKGA+. Therefore, this study corroborates FNCHC+ as the current state-of-the-art metaheuristic algorithm for the bandwidth reduction of real-world graphs. In future research, we intend to explore the application of new heuristics based on other metaheuristics to this optimization problem.

Acknowledgments. The first author acknowledges IFSC (Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina). The second author acknowledges the Brazilian funding agency CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) for financial support with grant 402599/2023-3. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. The authors also thank the anonymous referees for their careful reading, insightful comments, and suggestions.

Disclosure of Interests. The authors declare that they have no competing interests.

References

1. Abreu, A.A.A.M., Gonzaga de Oliveira, S.L.: Iterated local search with Tabu search for the bandwidth reduction problem in graphs. In: et al., O.G. (ed.) ICCSA 2024. LNCS, vol. 14813, pp. 125–136. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-64605-8_9
2. Amparore, E.G., Beccuti, M., Donatelli, S.: Gradient-based variable ordering of decision diagrams for systems with structural units. In: D'Souza, D., Kumar, K.N. (eds.) ATVA 2017. LNCS, vol. 10482, pp. 184–200. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68167-2_13

3. Bean, J.C.: Genetic algorithms and random keys for sequencing and optimization. ORSA J. Comput. **6**(2), 154–160 (1994). https://doi.org/10.1287/ijoc.6.2.154
4. Berry, M.W., Hendrickson, B., Raghavan, P.: Sparse matrix reordering schemes for browsing hypertext. In: Renegar, J., Shub, M., Smale, S. (eds.) Lectures in Applied Mathematics, vol. 32, pp. 99–123. American Mathematical Society Press, Park City, Utah, USA (1996), the Mathematics of Numerical Analysis (1996)
5. Bolanos, M.E., Aviyente, S., Radha, H.: Graph entropy rate minimization and the compressibility of undirected binary graphs. In: Proceedings of the IEEE Statistical Signal Processing Workshop (SSP), Ann Arbor, MI, pp. 109–112. IEEE (2012). https://doi.org/10.1109/SSP.2012.6319634
6. Campos, V., Piñana, E., Martí, R.: Adaptive memory programming for matrix bandwidth minimization. Ann. Oper. Res. **183**, 7–23 (2011)
7. Cavero, S., Pardo, E.G., Duarte, A., Martí, R.: Graph Layout Problems. Springer, Cham (2025). https://doi.org/10.1007/978-3-319-07153-4_45-2
8. Chen, H.K.: Evaluation of triangular mesh layout techniques using large mesh simplification. Multimedia Tools Appl. **76**(23), 25391–25419 (2017). https://doi.org/10.1007/s11042-017-4607-z
9. Concas, A., Fenu, C., Rodriguez, G.: PQser: a Matlab package for spectral seriation. Numerical Algorithms **80**(3), 879–902 (2019). https://doi.org/10.1007/s11075-018-0510-6
10. Davis, T.A., Hu, Y.: The University of Florida sparse matrix collection. ACM Trans. Math. Software **38**(1), 1–25 (2011). https://doi.org/10.1145/2049662.2049663
11. Dueck, G.W., Jeffs, J.: A heuristic bandwidth reduction algorithm. J. Comb. Math. Comb. Comput. 97–108 (1995)
12. Esposito, A., Tarricone, L.: Matrix bandwidth reduction with Tabu search: parallel implementation on Cray T3D and applications to microwave circuit design. In: Voli, E. (ed.) Science and Supercomputing at CINECA: Report, Bologna, Italy, pp. 440–450. CINECA (1996)
13. Garey, M.R., Graham, R.L., Johnson, D.S., Knuth, D.: Complexity results for bandwidth minimization. SIAM J. Appl. Math. **34**, 477–495 (1978)
14. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, New York (1979)
15. George, A., Liu, J.W.H.: Computer Solution of Large Sparse Positive Definite Systems. Prentice-Hall, Englewood Cliffs (1981)
16. George, A., Liu, J.W.H.: An implementation of a pseudoperipheral node finder. ACM Trans. Math. Software **5**(3), 284–295 (1979). https://doi.org/10.1145/355841.355845
17. Gibbs, N.E., Poole, W.G., Stockmeyer, P.K.: An algorithm for reducing the bandwidth and profile of a sparse matrix. SIAM J. Numer. Anal. **13**(2), 236–250 (1976). https://doi.org/10.1137/0713023
18. Gibbs, N.E., Poole, W.G., Stockmeyer, P.K.: Comparison of several bandwidth and profile reduction algorithms. ACM Trans. Math. Software **4**(1), 322–330 (1976)
19. Gonçalves, J.F., Resende, M.G.C.: Biased random-key genetic algorithms for combinatorial optimization. J. Heuristics **17**(5), 487–525 (2011). https://doi.org/10.1007/s10732-010-9143-1
20. Gonçalves, J.F., Resende, M.G.C.: Random-key genetic algorithms. In: Martí, R., Pardalos, P.M., Resende, M.G.C. (eds.) Handbook of Heuristics, chap. 13, pp. 703–715. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-07124-4_30

21. Gonzaga de Oliveira, S.L.: An evaluation of heuristic methods for the bandwidth reduction of large-scale graphs. Revista Pesquisa Operacional **43**(e268255), 1–22 (2023). https://doi.org/10.1590/0101-7438.2023.043.00268255
22. Gonzaga de Oliveira, S.L.: A survey of heuristics for matrix bandwidth reduction. Comput. Sci. Rev. **56**, 100724 (2025). https://doi.org/10.1016/j.cosrev.2025.100724
23. Gonzaga de Oliveira, S.L., de Abreu, A.A.A.M., Robaina, D., Kischinhevsky, M.: A new heuristic for bandwidth and profile reductions of matrices using a self-organizing map. In: Gervasi, O., et al. (eds.) ICCSA 2016. LNCS, vol. 9786, pp. 54–70. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42085-1_5
24. Gonzaga de Oliveira, S.L., Carvalho, C.: Metaheuristic algorithms for the bandwidth reduction of large-scale matrices. J. Comb. Optim. **43**, 727–784 (2022). https://doi.org/10.1007/s10878-021-00801-6
25. Gonzaga de Oliveira, S.L., Silva, L.M.: An ant colony hyperheuristic approach for matrix bandwidth reduction. Appl. Soft Comput. **94**, 106434 (2020). https://doi.org/10.1016/j.asoc.2020.106434
26. Gonzaga de Oliveira, S.L., Silva, L.M.: Evolving reordering algorithms using an ant colony hyperheuristic approach for accelerating the convergence of the ICCG method. Eng. Comput. **36**(4), 1857–1873 (2019). https://doi.org/10.1007/s00366-019-00801-5
27. Gonzaga de Oliveira, S.L., Bernardes, J.A.B., Chagas, G.O.: An evaluation of reordering algorithms to reduce the computational cost of the incomplete Cholesky-conjugate gradient method. Comput. Appl. Math. **37**(3), 2965–3004 (2017). https://doi.org/10.1007/s40314-017-0490-5
28. Gonzaga de Oliveira, S.L., Bernardes, J.A.B., Chagas, G.O.: An evaluation of low-cost heuristics for matrix bandwidth and profile reductions. Comput. Appl. Math. **37**(2), 1412–1471 (2016). https://doi.org/10.1007/s40314-016-0394-9
29. Harper, L.H.: Optimal assignments of numbers to vertices. J. Soc. Ind. Appl. Math. **12**, 131–135 (1964)
30. Harper, L.H.: Optimal numberings and isoperimetric problems on graphs. J. Comb. Theory **1**(3), 385–393 (1966)
31. Higham, D.J.: Unravelling small world networks. J. Comput. Appl. Math. **158**(1), 61–74 (2003). https://doi.org/10.1016/S0377-0427(03)00471-0
32. Johnson, D.S.: A theoretician's guide to the experimental analysis of algorithms. In: Data Structures, Near Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Implementation Challenges, vol. 59, Providence, RI, pp. 215–250. American Mathematical Society (AMS) (2002)
33. Koohestani, B.: On the solution of the graph bandwidth problem by means of search methods. Appl. Intell. **53**, 7988–8004 (2023). https://doi.org/10.1007/s10489-022-03802-y
34. Koohestani, B., Poli, R.: A hyper-heuristic approach to evolving algorithms for bandwidth reduction based on genetic programming. In: Research and Development in Intelligent Systems XXVIII, pp. 93–106. Springer, London (2011). https://doi.org/10.1007/978-1-4471-2318-7_7
35. Lim, A., Lin, J., Xiao, F.: Particle Swarm optimization and Hill Climbing for the bandwidth minimization problem. Appl. Intell. **3**(26), 175–182 (2007)
36. Lim, A., Rodrigues, B., Xiao, F.: A fast algorithm for bandwidth minimization. Int. J. Artif. Intell. Tools **16**(3), 537–544 (2007). https://doi.org/10.1142/s0218213007003394
37. Lim, A., Rodrigues, B., Xiao, F.: Heuristics for matrix bandwidth reduction. Eur. J. Oper. Res. 69–91 (2006). https://doi.org/10.1016/j.ejor.2005.02.066

38. López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Birattari, M., Stützle, T.: The Irace package: Iterated racing for automatic algorithm configuration. Oper. Res. Perspect. **3**, 43–58 (2016). https://doi.org/10.1016/j.orp.2016.09.002
39. Marichal, R., Dufrechou, E., Ezzatti, P.: Optimizing sparse matrix storage for the big data era. In: Naiouf, M., Rucci, E., Chichizola, F., De Giusti, L. (eds.) Communications in Computer and Information Science (JCC-BD&ET 2021: Cloud Computing, Big Data & Emerging Topics), vol. 1444, pp. 121–135. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-84825-5_9
40. Martí, R., Laguna, M., Glover, F., Campos, V.: Reducing the bandwidth of a sparse matrix with Tabu search. Eur. J. Oper. Res. **135**(2), 450–459 (2001). https://doi.org/10.1016/S0377-2217(00)00325-8
41. Mladenovic, N., Urosevic, D., Pérez-Brito, D., García-González, C.G.: Variable neighbourhood search for bandwidth reduction. Eur. J. Oper. Res. **200**(1), 14–27 (2010). https://doi.org/10.1016/j.ejor.2008.12.015
42. Mueller, C., Martin, B., Lumsdaine, A.: A comparison of vertex ordering algorithms for large graph visualization. In: Proceedings of the 6th International Asia-Pacific Symposium on Visualization (APVIS'07), Sydney, Australia, pp. 141–148 (2007). https://doi.org/10.1109/APVIS.2007.329289
43. Papadimitriou, C.: The NP-completeness of bandwidth minimization problem. Comput. J. **16**, 177–192 (1976). https://doi.org/10.1007/BF02280884
44. Petit, J.: Addenda to the survey of layout problems. Bull. EATCS **105**, 177–201 (2011)
45. Piñana, E., Plana, I., Campos, V., Martí, R.: GRASP and path relinking for the matrix bandwidth minimization. Eur. J. Oper. Res. **153**(1), 200–210 (2004). https://doi.org/10.1016/S0377-2217(02)00715-4
46. Rodriguez-Tello, E., Kao, H.J., Torres-Jimenez, J.: An improved simulated annealing algorithm for bandwidth minimization. Eur. J. Oper. Res. **185**, 1319–1335 (2008)
47. Silva, P.H.G., Brandão, D.N., Morais, I.S., Gonzaga de Oliveira, S.L.: A biased random-key genetic algorithm for bandwidth reduction. In: Gervasi, O., et al. (eds.) ICCSA 2020, Part VI. LNCS, vol. 12249, pp. 312–321. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58799-4_23
48. Spears, W., De Jong, K.: On the virtues of parametrized uniform crossover. In: Proceedings of the 4th International Conference on Genetic Algorithms, San Diego, CA, USA, July 1991. pp. 230–236 (1991). https://doi.org/10.21236/ADA293985
49. Tinney, W., Walker, J.: Direct solutions of sparse network equations by optimally ordered triangular factorization. Proc. IEEE **55**(11), 1801–1809 (1967). https://doi.org/10.1109/PROC.1967.6011
50. Torres-Jimenez, J., Izquierdo-Marquez, I., Garcia-Robledo, A., Gonzalez-Gomez, A., Bernal, J., Kacker, R.N.: A dual representation simulated annealing algorithm for the bandwidth minimization problem on graphs. Inf. Sci. **303**, 33–49 (2015). https://doi.org/10.1016/j.ins.2014.12.041

Shelf Space Allocation with Rectangular Arrangement: A Comparative Study of Three MILP Formulations

Daniel Martins[1(✉)], Maria João Alves[1,2], and Cristela Bairrada[1]

[1] Faculty of Economics, Univ Coimbra, CeBER, Av Dias da Silva 165, 3004-512 Coimbra, Portugal
danielmmail@gmail.com, {mjalves,cristela.bairrada}@fe.uc.pt
[2] INESC Coimbra, Coimbra, Portugal

Abstract. In this work, we compare three mixed-integer linear programming formulations for the shelf space allocation problem with rectangular arrangement: an original model proposed in the literature and two new formulations. The new proposed models differ in how they enforce rectangular arrangements with respect to the original model: one is a compact-variable formulation, reducing the number of variables at the expense of additional constraints, while the other keeps a similar size to the original model but uses a more direct representation of the rectangularity conditions. Using selected real instances, we evaluate solution times and best objective values with a general-purpose solver (Gurobi) under the original setting, where only row effectiveness is considered, and settings that also include column attractiveness under several patterns. Overall, when column attractiveness is included, the new models consistently outperformed the original one, with the third model achieving the best performance in most cases. The experiments confirm that the problem is very difficult to solve at optimality when the size of instances increases, and explicitly modeling both row and column attractiveness can significantly affect the solution landscape and performance of the models.

Keywords: Shelf space allocation · Retailing · Mixed-integer linear programming

1 Introduction

Shelf space allocation has been a central topic in retail research for decades, given its direct impact on sales and profitability. Earlier studies have shown that demand responds to the amount of space provided (space elasticity) and to where items are placed (location elasticity) [1, 2]. More recent studies further link layout to the effectiveness of marketing instruments [3] and provide comprehensive reviews of shelf planning models and rules [4]. The challenge lies in optimally distributing limited shelf space among numerous products to maximize returns, considering space elasticity, location effects and merchandising rules such as rectangular arrangement [5].

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 179–191, 2026.
https://doi.org/10.1007/978-3-032-30491-9_12

In this work we focus on the Shelf Space Allocation Problem with Rectangular Arrangement (SSAP-RA) introduced by Gencosman and Begen [5], who proposed a mixed-integer linear programming (MILP) formulation to address the problem.

The SSAP-RA model is intended to determine the number of facings (defined as the number of product units visible at the front of the shelf) for each product, in both the vertical and horizontal dimensions, and their allocation in a rectangular shelf layout. The objective of the retailer is to maximize profit considering space-elasticity, i.e., the effect of the number of facings on the demand, and location-elasticity, i.e., the effect of products' position on the demand of products. In this model, only row position effectiveness is considered. As the use of a general MILP solver for the SSAP-RA model has not allowed to solve realistic instances in acceptable times, two exact solution approaches were developed in [5]. The first method is based on logic-based Benders decomposition, which decomposes the optimization problem into a master problem and a sub-problem. The master problem contains only some decision variables and the values found for these variables are parameters to the subproblem to obtain possible solutions. Feasibility cuts and optimality cuts are iteratively imposed until an optimal solution is found. Although this method can handle larger instances compared to solving the MILP formulation, it still fails to solve some instances. Therefore, inspired by the previous method, a two-stage iterative approach was built by the authors, which separates the MILP model into two simple models considering row and column consecutiveness conditions separately. This approach has proven to be effective and could deal with larger instances. However, these methods are specific to the model addressed and rely on the assumption that the column position of the product has no effect on the optimization problem, that is, attractiveness varies only across rows. As a result, these approaches do not apply if placing a product in one column has a different effect on demand than placing it in another. The vertical dimension, however, also influences product, with the eye and hand level positions identified in the literature as the best locations [2, 6]. With respect to the horizontal dimension some authors support the best placement on the edges and others advocate for placement in the central columns of a cabinet [3, 6].

Our goal is to consider both row and column attractiveness, comparing the SSAP-RA formulation with two new more compact formulations - Shelf Space Allocation Problem with Rectangular Fulfilment Arrangement (SSAP-RFA) and Shelf Space Allocation Problem with Interval-Based Rectangular Arrangement (SSAP-IBRA). Both models accommodate horizontal (column) and vertical (row) attractiveness. SSAP-RFA model reduces the number of variables but significantly increases the number of constraints. SSAP-IBRA model has a comparable number of variables and constraints to the SSAP-RA model, but uses a more direct representation of the rectangularity conditions, which is easier to understand. Using selected real instances from [5], we evaluate solution times and best objective values with a general-purpose solver (Gurobi) under the original setting, where only row effectiveness is considered, and settings that also include column attractiveness under several patterns (center-most and edge-most columns, and an alternative multiplicative row $\times$ column attractiveness model). Computational experiments have shown that the new models consistently outperform the original model when both row and column attractiveness are both considered. The experiments also confirmed that

solving the problem at optimality is very difficult and different row/column attractiveness scenarios influence the performance of the models.

The remainder of the paper is organized as follows: Sect. 2 describes the original model (SSAP-RA). Section 3 presents our new models – SSAP-RFA and SSAP-IBRA models. Section 4 reports and discusses the computational results. Section 5 concludes the paper and outlines future work.

2 The SSAP-RA Model

The SSAP-RA model by Gencosman and Begen [5] aims to decide how products should be displayed on shelves to maximize the total expected profit. The products should be placed in rectangular blocks (consecutive rows and columns), which reflects practical merchandising practices. The key variables represent the number of facings (units visible on the shelf) of each product and where each product is placed on the shelf. Since the motivation for this study has been an application to a local bookstore, the *facings* represent the number of copies of a book title that are displayed on the shelf so that their front side is visible to customers.

The problem addressed in [5] considers K cabinets. Each cabinet k includes R_k rows and C_k columns, so each cabinet has $R_k \times C_k$ slots. A slot is the space where the thinnest book can be allocated. Thus, a facing of a product can be allocated to more than one slot, depending on the thickness of each product. Empty slots are not allowed. All the facings of a product must be placed in the same cabinet in a rectangular layout. Without loss of generality, the present work considers only one cabinet ($K = 1$), meaning that the k index can be omitted from certain parameters and variables. The original model only considers the effect of vertical position on demand (i.e., the effectiveness of the rows A_i) but not that of the columns. In the formulation below, we have adapted the model to also incorporate the impact of horizontal position (replacing A_i with A_{ij} and adjusting the objective function accordingly). Table 1 shows the notation of this model.

For the sake of completeness, we now present the full MILP formulation of SSAP-RA. Further explanations can be found in [5].

$$\max \text{Profit} = \sum_{d \in D} \sum_{i \in I} \sum_{j \in J} V_d A_{ij} t_{dij} / T_d \tag{1}$$

s.t.

$$\sum\nolimits_{\mathrm{d} \in \mathrm{D}} t_{dij} = 1, \forall i \epsilon I, j \epsilon J \tag{2}$$

$$\sum\nolimits_{j \in J} t_{dij} = n_{di} T_d, \qquad \forall d \epsilon D, i \epsilon I \tag{3}$$

$$\sum\nolimits_{i \in I} n_{di} = m_d, \qquad \forall d \epsilon D \tag{4}$$

$$\sum\nolimits_{i \in I} t_{dij} = p_{dj}, \qquad \forall d \epsilon D, j \epsilon J \tag{5}$$

$$L_d \leq m_d \leq U_d, \qquad \forall d \epsilon D \tag{6}$$

Table 1. Notation for the SSAP-RA model

Sets and indices:	*Parameters:*
D- Set of products, $D = \{1,2, \ldots, D_{max}\}$	D_{max}- Maximum number of products
I- Set of rows, $I = \{1,2, \ldots, R\}$	R- Number of rows in the cabinet
J- Set of columns, $J = \{1,2, \ldots, C\}, j \in J$	C- Number of columns in the cabinet
d- Index of products, $d \in D$	A_{ij}- Effectiveness of slot (i, j)
i- Index of rows, $i \in I$	V_d- Profit of product d
j- Index of columns, $j \in J$	L_d, U_d - Lower and upper bound for the number of facings of product d
	T_d- Thickness of product d
Decision variables:	
$y_{di} \in \{0,1\}$: 1, if product d is assigned to row i; 0, otherwise	
$z_{dj} \in \{0,1\}$: 1, if product d is assigned to column j; 0, otherwise	
$t_{dij} \in \{0,1\}$: 1, if product d is assigned to row i and column j; 0, otherwise	
m_d: Total number of facings of product d	
n_{di}: Number of facings of product d in row i	
p_{dj}: Number of slots occupied by product d in column j (which also represents the number of facings of product d in column j)	
w_d: : Width of the rectangle (number of slots = columns) assigned to product d	
h_d: : Height of the rectangle (number of rows) assigned to product d	

$$\sum\nolimits_{d \in D} n_{di} T_d = C, \qquad \forall i \epsilon I \tag{7}$$

$$\sum\nolimits_{d \in D} p_{dj} = R, \qquad \forall j \epsilon J \tag{8}$$

$$w_d - n_{di} T_d \leq C(1 - y_{di}), \qquad \forall d \epsilon D, i \epsilon I \tag{9}$$

$$n_{di} T_d \leq C y_{di}, \qquad \forall d \epsilon D, i \epsilon I \tag{10}$$

$$h_d - p_{dj} \leq R\big(1 - z_{dj}\big), \qquad \forall d \epsilon D, j \epsilon J \tag{11}$$

$$p_{dj} \leq R z_{dj}, \qquad \forall d \epsilon D, j \epsilon J \tag{12}$$

$$n_{di} T_d \leq w_d, \qquad \forall d \epsilon D, i \epsilon I \tag{13}$$

$$\sum\nolimits_{i \in I} y_{di} \leq h_d, \qquad \forall d \epsilon D \tag{14}$$

$$y_{d,i+1} + w_d \geq y_{di} + n_{d,i+u}T_d, \qquad \forall d\epsilon D, i\epsilon I, u\epsilon I | 1 \leq u \leq R-1; \&i+u \leq R \quad (15)$$

$$z_{d,j+1} + h_d \geq z_{dj} + p_{d,j+u}, \qquad \forall d\epsilon D, j\epsilon J, u\epsilon J | 1 \leq u \leq C-1; \&j+u \leq C \quad (16)$$

$$y_{di}, z_{dj}, t_{dij}\epsilon\{0,1\}, \qquad \forall d\epsilon D, i\epsilon I, j\epsilon J \quad (17)$$

$$m_d, n_{di}, p_{dj}, w_d, h_d\epsilon\mathbb{Z}_0^+, \qquad \forall d\epsilon D, i\epsilon I, j\epsilon J \quad (18)$$

The objective function (1) maximizes the expected profit of displaying products. Since $\sum_{j\in J} t_{dij} = n_{di}T_d$, then the variable n_{di} in the objective function in [5] is replaced by $\sum_{j\in J} t_{dij}/T_d$, to also include the position of the facing in column. Constraints (2) assign exactly one facing unit to each slot. Constraints (3) compute the number of facings of each product d in each row i, n_{di}, and (4) calculate the total number of facings of each product d, m_d; (5) determine the number of facings of each product d in each column j, p_{dj}; (6) restrict the number of assigned facings of each product d with respect to its lower and upper bounds, L_d and U_d, respectively; (7) and (8) limit the display quantities of products with the dimensions of the cabinet that contains C columns and R rows; (9) and (10) ensure that, if product d exists in a row, the number of slots allocated to product d should be equal to the width of product d, w_d; (11) and (12) ensure that the number of facings of a product in different columns is equal to each other, h_d; (13) and (14) determine the width w_d and the height h_d of each product d, respectively; (15) guarantee that the product facings in a row are allocated consecutively and (16) do the same for columns, ensuring a rectangular arrangement of products. The expressions (17) and (18) define the solution space of the decision variables.

3 The New Proposed Models

We propose two more compact MILP formulations – SSAP-RFA model and SSAP-IBRA model – which differ in the way the rectangular arrangement is ensured. SSAP-RFA keeps only the assignment variables to slots and the row-wise facing counts, while enforcing rectangularity using rectangle fulfilment constraints. SSAP-IBRA introduces additional variables that define the rows and columns used by each product, as well as the start/end row/column for each rectangle. The aim is to compare the performance of these models with that of the original one (SSAP-RA). The sets, indices and parameters for these models are the same as presented in Table 1. The decision variables of SSAP-RA are only the following:

$t_{dij} \in \{0,1\}$: 1, if product d occupies slot (i, j); 0, otherwise.

$n_{di} \geq 0$integer: Number of facings of product d in row i.

SSAP-RFA and SSAP-IBRA both have the same objective function (19) and the set of constraints (20)–(26). These correspond to the objective function (1) and constraints (2), (3), (6)–(8) and (17)–(18) of SSAP-RA. Each model also includes specific constraints, which will be presented and explained below.

$$\max \text{Profit} = \sum_{d\in D}\sum_{i\in I}\sum_{j\in J} V_d A_{ij} t_{dij}/T_d$$

s.t. Exactly one product per slot (19)

$$\sum_{d \in D} t_{dij} = 1, \qquad \forall i \epsilon I, j \epsilon J \tag{20}$$

Facings per row

$$\sum_{j \in J} t_{dij} = n_{di} T_d, \qquad \forall d \epsilon D, i \epsilon I \tag{21}$$

Facing bounds

$$L_d \leq \sum_{i \in I} n_{di} \leq U_d, \qquad \forall d \epsilon D \tag{22}$$

Capacity in number of columns

$$\sum_{d \in D} n_{di} T_d = C, \qquad \forall i \epsilon I \tag{23}$$

Capacity in number of rows

$$\sum_{d \in D} \sum_{i \in I} t_{dij} = R, \qquad \forall j \epsilon J \tag{24}$$

Type of variables

$$t_{dij} \epsilon \{0,1\}, \qquad \forall d \epsilon D, i \epsilon I, j \epsilon J \tag{25}$$

$$n_{di} \epsilon \mathbb{Z}_0^+, \qquad \forall d \epsilon D, i \epsilon I \tag{26}$$

To ensure the fulfillment of the rectangle for each product, the following principle is followed in SSAP-RFA: if slots (i_1, j_1) and (i_2, j_2) are occupied by the same product d, as shown in Fig. 1, then all the slots between these two corners should also be occupied by product d. The idea is to interpret any pair of occupied slots as defining the corners of a potential rectangle and then impose the same product within this region.

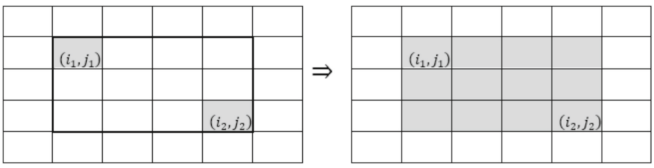


Fig. 1. Example of filling in the area from the top left to the bottom right.

Constraints (27) ensure the fulfillment illustrated in Fig. 1, in which $(i_2 - i_1 + 1)(j_2 - j_1 + 1)$ represents the area of the rectangle; the condition is redundant if either $t_{di_1j_1} = 0$ or $t_{di_2j_2} = 0$ (or both) and it must be satisfied if $t_{di_1j_1} = t_{di_2j_2} = 1$. M is a sufficiently large constant. Although the SSAP-RFA formulation relies on Big-M constraints, which can generally affect computational performance, in this case M can

be set explicitly to $R \times C$. This is valid because the area of the rectangle is never greater than $R \times C$. Therefore, the usual difficulty of determining appropriate Big-M values is avoided.

$$\sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} t_{dij} \geq (i_2 - i_1 + 1)(j_2 - j_1 + 1) - M\left(2 - t_{di_1j_1} - t_{di_2j_2}\right),$$

$$\forall d \epsilon D, i_1, i_2 \epsilon I, j_1, j_2 \epsilon J, i_2 \geq i_1, j_2 \geq j_1 \tag{27}$$

Constraints (27) search to the right and downward from each slot. However, searching in only one direction is not sufficient to cover all possible corner configurations, which is why a symmetric set of constraints is also necessary. To fill in the rectangles in all situations, we also need to search in another direction, downwards and to the left (as illustrated in Fig. 2) or upwards and to the right (corresponding to search from (i_2, j_2) to (i_1, j_1) in Fig. 2). Either one ensures that, together with (27), all cases are considered. Constraints (28) implement the former (in Fig. 2).

$$\sum_{i=i_1}^{i_2} \sum_{j=j_2}^{j_1} t_{dij} \geq (i_2 - i_1 + 1)(j_1 - j_2 + 1) - M\left(2 - t_{di_1j_1} - t_{di_2j_2}\right),$$

$$\forall d \epsilon D, i_1, i_2 \epsilon I, j_1, j_2 \epsilon J, i_2 \geq i_1, j_2 \leq j_1 \tag{28}$$

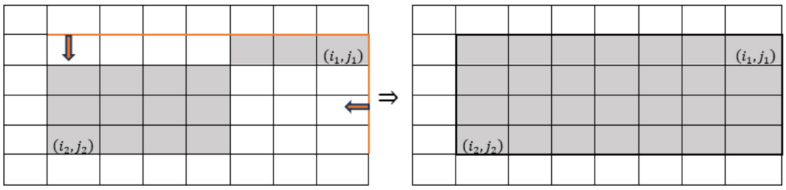


Fig. 2. Example of filling in the area from the top right to the bottom left.

The **SSAP-RFA** model is composed of (19)–(28).

The SSAP-IBRA model includes all the variables of SSAP-RFA, i.e., t_{dij} and n_{di}, and also the following binary variables:

$r_{di} \in \{0,1\}$: 1, if product d uses row i; 0, otherwise.

$c_{dj} \in \{0,1\}$: 1, if product d uses column j; 0, otherwise.

$r_{di}^1, r_{di}^2 \in \{0,1\}$: 1, if row i is the start/end interval row for product d.

$c_{dj}^1, c_{dj}^2 \in \{0,1\}$: 1, if column j is the start/end interval column for product d.

SSAP-IBRA explicitly specifies which rows and columns are occupied by each product, allowing the rectangle to emerge from choosing contiguous rows and columns.

A principle like the one in SSAP-RFA is considered to force each product d to occupy a single solid rectangle in $R \times C$ cabinet, but it is implemented in a different way: if (i_1, j_1) and (i_2, j_2) are assigned to d, then all (i, j) with $i \in [\min(i_1, i_2), \max(i_1, i_2)]$ and $j \in \left[\min(j_1, j_2), \max(j_1, j_2)\right]$ must also be assigned to d. The rectangular arrangement constraints ensure:

- Linkage between t_{dij} (use of slot (i, j) by product$d$) andr_{di},c_{dj}, such that$t_{dij} = r_{di}.c_{dj}$; these products between binary variables are linearized in (29);
- Contiguity in rows linking variables of using row$i$, r_{di}, to the start/end rows for$d$, r^1_{di}, r^2_{di}, which is modelled in constraints (30);
- Contiguity in columns, linking variables of using column$j$, c_{dj}, to the start/end columns for$d$,c^1_{dj}, c^2_{dj},, which is modelled in constraints (31).

Linearization of $t_{dij} = r_{di}.\, c_{dj}$

$$\begin{aligned} &t_{dij} \le r_{di}, \qquad \forall d \epsilon D, i \epsilon I, j \epsilon J \\ &t_{dij} \le c_{dj}, \qquad \forall d \epsilon D, i \epsilon I, j \epsilon J \\ &t_{dij} \ge r_{di} + c_{dj} - 1, \qquad \forall d \epsilon D, i \epsilon I, j \epsilon J \end{aligned} \tag{29}$$

Contiguity in rows.

$$\begin{aligned} &\textstyle\sum_{i \in I} r^1_{di} \le 1, \sum_{i \in I} r^2_{di} \le 1, \sum_{i \in I} r^1_{di} = \sum_{i \in I} r^2_{di}, \forall d \in D \\ &r_{di} = \textstyle\sum_{k \in I : k \le i} r^1_{dk} - \sum_{k \in I : k < i} r^2_{dk}, \forall d \in D,\ i \in I \end{aligned} \tag{30}$$

Contiguity in columns.

$$\begin{aligned} &\textstyle\sum_{j \in J} c^1_{dj} \le 1, \sum_{j \in J} c^2_{dj} \le 1, \sum_{j \in J} c^1_{dj} = \sum_{j \in J} c^2_{dj}, \qquad \forall d \in D \\ &c_{dj} = \textstyle\sum_{h \in J : h \le j} c^1_{dh} - \sum_{kh \in J : h < j} c^2_{dh}, \qquad \forall d \in D,\ j \in j \end{aligned} \tag{31}$$

$$\begin{aligned} &r_{di}, r^1_{di}, r^2_{di} \epsilon \{0,1\}, \qquad \forall d \epsilon D, i \epsilon I \\ &c_{dj}, c^1_{dj}, c^2_{dj} \epsilon \{0,1\}, d \epsilon D, j \epsilon J \end{aligned} \tag{32}$$

Constraints (30) ensure contiguity in rows by imposing that (i) there is at most one start and one end per product ($\sum_{i \in I} r^1_{di} \le 1, \sum_{i \in I} r^2_{di} \le 1$), ($ii$) the start and end are balanced ($\sum_{i \in I} r^1_{di} = \sum_{i \in I} r^2_{di}$) and ($iii$) the row usage r_{di} is defined as the following: before the start row, $r_{di} = 0$(because$\sum_{k \in I : k \le i} r^1_{dk} = \sum_{k \in I : k < i} r^2_{dk} = 0$);; from the start row up to the end row (inclusive), $r_{di} = 1$(because$\sum_{k \in I : k \le i} r^1_{dk} = 1$ and$\sum_{k \in I : k < i} r^2_{dk} = 0$); and after the end row, $r_{di} = 0$(because$\sum_{k \in I : k \le i} r^1_{dk} = \sum_{k \in I : k < i} r^2_{dk} = 1$). Thus, the rows used by product d form one contiguous interval. Constraints (31) ensure similar contiguity in columns.

The **SSAP-IBRA** model is composed of (19)–(26), (29)–(32).

4 Computational Results

First, we compare the performance of the three models in instances R1 to R9 of the *RealisticData* in [5] derived from bookstore data. As these instances only consider one cabinet ($K = 1$), the SSAP-RA model presented herein is exactly the same as the one in [5] when instantiated with these problems. The original SSAP-RA only considers the effect of vertical position on demand, i.e., the effectiveness of rows (A_i). Thus, for the

first comparison with the original instances R1-R9, we have set $A_{ij} = A_i$ for all column j. The parameters A_{ij} were then modified to consider column attractiveness under several patterns. A total of 44 instances were tested. The number of rows is 8 for all instances, while the number of books and columns varies.

Table 2 reports the number of decision variables and constraints for the instances in sets R1–R9 under each model.

Table 2. Number of variables and constraints of each set of problems

Prob	SSAP-RA		SSAP-RFA		SSAP-IBRA	
	Variables	Constraints	Variables	Constraints	Variables	Constraints
R1, R2, R3	990	1240	720	26100	1200	2320
R4, R5, R6	1980	2400	1440	52120	2400	4560
R7, R8, R9	4170	5096	3120	168896	4920	9836

We used Gurobi 12.0.3 with the AMPL modelling language. All runs were carried out on a standard laptop with an Intel Core i5-1335U@1.30 GHz, 16 GB RAM. A maximum time limit of 1800s per run was imposed for all formulations (as in the original paper). The tables below show the objective values and the running times (in seconds). The best times are highlighted in bold. If the time limit of 1800s is reached with an MIP gap larger than the default Gurobi value, this is also shown in the result tables.

A - Rows-Only Effectiveness

Table 3 reports the results obtained by the three models for the original R1-R9 instances. When columns are neutral (no horizontal attractiveness), the original model remains competitive and is sometimes faster than the alternative formulations. SSAP-RFA shows the weakest performance on this set of instances, with the least consistent running times. SSAP-IBRA achieves the best running times in 7 of the 9 instances.

Regarding column attractiveness, there is a lack of agreement in the literature about the best position along the horizontal axis within a cabinet. Some authors support the best placement on the edges, while others advocate for placement in the center. In contrast, in the vertical dimension, eye and hand level is often referred to as the best location [2, 6].

In the first scenario considered, the middle column is the most effective and effectiveness decreases towards the columns furthest to the left and right of the cabinet [6, 3]. To facilitate identification, the original dataset's instances have been labelled with "_B".

B - Center-Most Columns Preferred

Table 4 shows the results for the three models in the "_B" instances. These instances appear to be more challenging than the previous ones, typically requiring longer running times. For this and the subsequent experiments, we tested seven instances, from R1 to R7, since R7_B ran out of time on SSAP-RA and SSAP-RFA without ensuring that the optimal solution had been found.

In this test set, SSAP-IBRA outperformed the other models in 6 of the 7 instances.

Table 3. Comparison results considering rows-only effectiveness

Prob	# Books	#Cols	SSAP-RA		SSAP-RFA		SSAP-IBRA	
			Obj	Time	Obj	Time	Obj	Time
R1	10	8	2260	**0.20**	2260	0.66	2260	0.31
R2	10	8	1996	0.89	1996	0.69	1996	**0.47**
R3	10	8	1652	14.34	1652	139.39	1652	**13.84**
R4	20	8	2241	**7.91**	2241	125.00	2241	18.34
R5	20	8	2316	4.64	2316	18.78	2316	**1.61**
R6	20	8	2432	1.11	2432	2.22	2432	**0.39**
R7	30	12	3646	5.69	3646	8.11	3646	**4.70**
R8	30	12	2965	13.64	2965	13.58	2965	**5.17**
R9	30	12	2791	8.02	2791	10.97	2791	**6.69**

Table 4. Comparison results considering center-most columns preferred

Prob	SSAP-RA		SSAP-RFA		SSAP-IBRA	
	Obj	Time	Obj	Time	Obj	Time
R1_B	4456	157.45	4456	85.05	4456	**46.77**
R2_B	3903.6	195.41	3903.6	76.95	3903.6	**47.72**
R3_B	3235.2	195.41	3235.2	349.75	3235	**53.55**
R4_B	4391.9	496.09	4391.9	212.16	4391.9	**37.06**
R5_B	4538.1	230.59	4538.1	**41.61**	4538.1	52.45
R6_B	4752	670.91	4752	164.55	4752	**64.45**
R7_B	8783.5 *mipgap* = 0.14%	1800	**8787** *mipgap* = 0.03%	1800	**8787**	**1027.47**

B2 - Center-Most Columns Preferred with $T_d = 2$.

With the aim of undertaking further analysis and determining whether product thickness can affect model performance, at this stage we consider the same A_{ij} values as in study B, but assume that the thickness of all products is 2 ($T_d = 2, \forall d \epsilon D$). This change has been made to the previous dataset. The instances are identified as "_B2". Table 5 shows the comparison results for the three models.

In this test set, SSAP-RFA and SSAP-IBRA produced the best results, with SSAP-IBRA achieving better overall results than SSAP-RFA. SSAP-RA was not the best in any instance.

C2 - Edges preferred with $T_d = 2$.

Table 5. Comparison results considering center-most columns preferred with $T_d = 2$

Prob	SSAP-RA		SSAP-RFA		SSAP-IBRA	
	Obj	Time	Obj	Time	Obj	Time
R1_B2	4719.5	16.56	4719.5	16.97	4719.5	**6.03**
R2_B2	4441	324.64	4441	143.91	4441	**46.66**
R3_B2	3919.5	21.53	3919.5	76.27	3919.5	**16.89**
R4_B2	4756.5	63.27	4756.5	11.27	4756.5	**5.70**
R5_B2	4769.5	160.78	4769.5	**10.38**	4769.5	16.02
R6_B2	4572.5	181.44	4572.5	**16.83**	4572.5	32.27
R7_B2	8652 *mipgap* = 0.05%	1800	8652	827.5	8652	**247.39**

In another scenario, the columns are considered to be most effective at the far left and right edges of the cabinet, and less effective in the middle column [6]. We tested this by changing A_{ij} in the B2 instances, where the thickness of all products is 2 ($T_d = 2, \forall d \epsilon D$). Table 6 shows the comparison results for the three models. The performance of the models in this test set is similar to that observed in experiment B2, with SSAP-IBRA showing an even stronger advantage over the other models. SSAP-IBRA ensured the optimal solution in all instances and did not hit the time limit in any case. Moreover, it achieved shorter running times in all but one instance.

Table 6. Comparison results considering edges preferred with $T_d = 2$

Prob	SSAP-RA		SSAP-RFA		SSAP-IBRA	
	Obj	Time	Obj	Time	Obj	Time
R1_C2	4714.5	26.36	4714.5	99.11	4714.5	**8.64**
R2_C2	4440	81.99	4440	78.50	4440	**21.83**
R3_C2	3915.5	104.89	3915.5	85.29	3915.5	**16.36**
R4_C2	4756.5	111.20	4756.5	28.42	4756.5	**14.94**
R5_C2	4767.5	205.72	4767.5	59.10	4767.5	**31.34**
R6_C2	4572.5	210.67	4572.5	**34.02**	4572.5	37.44
R7_C2	8643 *mipgap* = 0.13%	1800	8643 *mipgap* = 0.08%	1800	8643	**580.20**

D - Multiplicative Row × Column Attractiveness

In this scenario, the slot attractiveness is considered to be the product of row and column factors, $A_{ij} = A_i \bullet A_j$, where A_i are the original parameters and A_j are the parameters for the center-most columns scenario. The results are shown in Table 7. These results

corroborate previous findings, showing the superiority of SSAP-IBRA in all cases but one.

Table 7. Comparison results considering $A_i \times A_j$ (center-most columns preferred)

Prob	SSAP-RA		SSAP-RFA		SSAP-IBRA	
	Obj	Time	Obj	Time	Obj	Time
R1_D	14785	63.83	14785	48.34	14785	**25.20**
R2_D	13069	83.41	13069	68.67	13069	**14.75**
R3_D	10915	90.91	10915	79.06	10915	**32.36**
R4_D	14956.5	860.66	14956.5	**119.02**	14956.50	291.22
R5_D	15275	245.47	15275	43.73	15275	**33.14**
R6_D	16200.5	623.70	16200.5	103.27	16200.5	**43.92**
R7_D	35947.67	1515.73	35947.67	290.09	35947.67	**87.47**

D2 - Multiplicative row × column attractiveness with $T_d = 2$.

The last experiment used the previous test set, but considering $T_d = 2, \forall d \epsilon D$. Table 8 shows the comparison results. This experiment reinforces the conclusions of experiments B2 and C2, showing that, when all products have the same thickness, SSAP-RFA performs better than in other scenarios, sharing the best times with SSAP-IBRA.

Table 8. Comparison results considering $A_i \times A_j$ (center-most columns preferred) with $T_d = 2$

Prob	SSAP-RA		SSAP-RFA		SSAP-IBRA	
	Obj	Time	Obj	Time	Obj	Time
R1_D2	15188.5	32.13	15188.5	69.47	15188.5	**15.00**
R2_D2	14762	561.52	14762	113.80	14762	**36.80**
R3_D2	13035.5	76.02	13035.5	38.92	13035.5	**8.92**
R4_D2	15725	86.39	15725	**6.91**	15725	8.56
R5_D2	15820.5	94.67	15820.5	**9.47**	15820.5	15.06
R6_D2	15216	235.44	15216	35.77	15216	**27.11**
R7_D2	34363.5 $mipgap = 0.05\%$	1800	**34364**	568.99	**34364**	**208.05**

5 Conclusions

This study compared three MILP formulations for a shelf space allocation problem with rectangular arrangement. It extended the original model SSAP-RA in [5] to incorporate both row and column attractiveness and proposed two new alternative formulations -

SSAP-RFA and SSAP-IBRA models. Ignoring column effects can lead to biased allocations by considering equivalent positions on shelves that, in practice, result in different demand responses. However, including these effects may increase the difficulty of the problems, change the performance of the solvers and make specialized methods not applicable.

Computational experiments showed that the new models consistently outperformed the original model when both row and column attractiveness are considered. Overall, the SSAP-IBRA model achieved the best performance by providing the optimal solution in the shortest time in 35 of the 44 instances. The SSAP-RFA model was the second best, providing the fastest solution in 7 of the 44 instances. The original SSAP-RA model performed best in only 2 instances.

Computational experiments also confirm that solving this problem to optimality is very challenging. Due to the large number of variables and constraints of the MILP formulations, the use of general solvers is limited to small problem sizes. Future work will focus on two directions: (i) linking optimization to behavioral insights by estimating average attractiveness indices from planograms and consumer surveys [7], a pathway that we have already started to explore; (ii) developing metaheuristics or hybrid approaches to handle larger instances efficiently.

Acknowledgments. This work has been funded by national funds through FCT – Fundação para a Ciência e a Tecnologia, I.P., through project UIDB/05037/2025 with DOI: https://doi.org/10.54499/UIDB/05037/2025.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Corstjens, M., Doyle, P.: A Model for Optimizing Retail Space Allocations. Manage. Sci. **27**, 822–833 (1981). https://doi.org/10.1287/mnsc.27.7.822
2. Drèze, X., Hoch, S.J., Purk, M.E.: Shelf management and space elasticity. J. Retail. **70**, 301–326 (1994). https://doi.org/10.1016/0022-4359(94)90002-7
3. van Nierop, E., Fok, D., Franses, P.H.: Interaction between shelf layout and marketing effectiveness and its impact on optimizing shelf arrangements. Mark. Sci. **27**, 1065–1082 (2008). https://doi.org/10.1287/mksc.1080.0365
4. Bianchi-Aguiar, T., Hübner, A., Carravilla, M.A., Oliveira, J.F.: Retail shelf space planning problems: a comprehensive review and classification framework. Eur. J. Oper. Res. **289**, 1–16 (2021). https://doi.org/10.1016/j.ejor.2020.06.018
5. Gencosman, B.C., Begen, M.A.: Exact optimization and decomposition approaches for shelf space allocation. Eur. J. Oper. Res. **299**, 432–447 (2022). https://doi.org/10.1016/j.ejor.2021.08.047
6. Bai, R.: An investigation of novel approaches for optimising retail shelf space allocation (2005). https://eprints.nottingham.ac.uk/10153
7. Rooderkerk, R.P., Lehmann, D.R.: Incorporating Consumer Product Categorizations into Shelf Layout Design. J. Mark. Res. **58**, 50–73 (2021). https://doi.org/10.1177/0022243720964127

Architecture Foundations and Deployment Scalability for CWMP: A Technical Review

Marco Marinello(✉)

Qnets, "Internet that Just Works", 20143 Milano, MI, Italy
marco@marinello.bz.it

Abstract. The complexity of the Internet requires the use of sophisticated management protocols capable of handling heterogeneous hardware at different functional levels. The CPE WAN Management Protocol (CWMP) serves as the basic framework for the management of subscriber devices from afar, emerging from the initial TR-069 specification to the more complex data model in TR-181. This paper provides an overview of the architecture behind CWMP, specifically focusing on the importance of separating the SOAP-based communication infrastructure from the data models themselves, enabling the management of various hardware devices, such as home gateways, Access Points, and VoIP phones. Unlike existing literature that focuses primarily on theoretical scalability or high-level autonomic loops, this paper contributes a practical analysis of "zero-touch" lifecycle management within high-concurrence ISP environments. We examine the deployment of these protocols through the lens of GenieACS, an open-source implementation that leverages TypeScript and MongoDB to handle high-concurrency ISP workloads. The originality of this work lies in its systematic review of modular ACS design and its identification of the critical documentation and implementation gaps that hinder the adoption of Free and Open Source Software (FOSS) solutions in production. Finally, we discuss the transition toward emerging methodologies, such as the User Services Platform (USP/TR-369), in the context of the lessons learned from two decades of TR-069 lifecycle management.

Keywords: TR-069 · CWMP · GenieACS · Device Management · Zero-Touch Provisioning · Internet · ISP Operations · FOSS

1 Introduction

The evolution of the Internet from a collection of static workstations to a massive ecosystem of billions of heterogeneous endpoints has fundamentally reshaped network management. In the contemporary Internet of Things (IoT) and broadband landscape, Customer Premises Equipment (CPE) – ranging from high-speed home gateways to smart utility meters – is increasingly managed externally by service providers. This "zero-touch" paradigm requires seamless remote provisioning, configuration, and maintenance without end-user intervention.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 192–203, 2026.
https://doi.org/10.1007/978-3-032-30491-9_13

To address the operational impossibility of manual configuration at scale, the Broadband Forum[1] introduced the CPE WAN Management Protocol (CWMP) [1], designated as TR-069[2]. By defining a standardized communication layer over HTTP/SOAP between the CPE and an Auto Configuration Server (ACS), TR-069 provides a robust framework for firmware management, status monitoring, and diagnostic execution.

While being one of the most widely used solutions in global telecommunications infrastructure, implementation techniques, as well as operational peculiarities of TR-069-based ACS are not widely discussed in the literature, where more modern approaches like MQTT and CoAP take center stage. Moreover, many previous works often neglect the challenges associated with implementation of ACS in a high concurrence environment. This paper aims to fill that void through a systematic review of the structure of TR-069-based ACS deployments. In particular, we analyze the process of CWMP session initialization, explore key features of the RPC mechanism used for managing the devices, and present a modular design of ACS implementation architecture.

2 Related Work

Research into remote device management has grown since TR-069 appeared. However, most papers focus on high-level architecture or network theory, often skipping the messy details of protocol implementation. Many studies prioritize mathematical scalability models for large carrier networks [2]. These help define what autoconfiguration servers can theoretically do, but they usually ignore the technical friction found in real-world operations.

Rachidi and Karmouch [3] offer one example of this theoretical focus. They designed autonomic devices using IBM's MAPE-K loop, showing how CWMP can reconfigure Set-Top Boxes when quality drops. Their work clarifies autonomic logic but largely overlooks the protocol's actual constraints.

Stusek et al. [4] broadened the scope by applying TR-069 to smart meters and health monitors. While their early work stayed abstract, their extended study [5] introduced a modular implementation using OSGi bundles and GenieACS. This shift from theory to a functional smart home gateway architecture highlights the protocol's practical utility in heterogeneous IoT environments.

Our study bridges theoretical design and operational practice. We analyze the ACS architecture through a strictly implementation-oriented lens. Instead of abstract models, we focus on session initiation and RPC management within GenieACS. This technical decomposition draws directly from experience managing large-scale production fleets. It addresses the practical friction points that theoretical papers often overlook.

[1] https://www.broadband-forum.org, accessed 8 May 2026.
[2] https://www.broadband-forum.org/pdfs/tr-069-1-6-1.pdf, accessed 8 May 2026.

3 Technical Architecture and Protocol Stack

CWMP is essentially a two-way protocol based on SOAP/HTTP that aims at facilitating communication between the Customer Premises Equipment and Auto Configuration Server. The following subsections discuss the architecture and protocol stack of the management framework.

3.1 Communication Stack and SOAP Encapsulation

The CWMP protocol stack relies on standard Internet protocols that ensure interoperability across diverse network hardware. At the transport layer, TCP/IP provides reliable delivery, while HTTP/S acts as the application carrier. TR-069 utilizes SOAP[3] to encapsulate management commands and responses in XML format.

A typical session consists of an HTTP Post request containing a SOAP envelope. This overhead, while significant for low-power IoT devices, allows for a highly structured and extensible command set, enabling the ACS to manipulate complex data models like TR-181.

3.2 Decoupling Protocol Logic from the Data Model

The primary strength of the TR-069 framework lies in the strict separation between the transport mechanism (CWMP) and the information payload. By keeping the SOAP/HTTP envelope agnostic to the XML content it carries, the protocol supports a data model that is both fluid and highly extensible. This abstraction ensures the management interface remains remarkably versatile, adapting not only to diverse device categories – such as VoIP gateways and DSL modems – but also to the incremental feature sets introduced across different firmware versions of the same hardware.

This flexibility is operationalised through standardised root objects, most notably `Device.` and the legacy `InternetGatewayDevice.` framework. When a manufacturer introduces new functionality via a firmware update, they simply extend the existing data model tree with new parameter paths. Crucially, the ACS does not require a protocol-level upgrade to support these additions; it merely needs to be aware of the new nodes within the hierarchy. Such inherent forward compatibility allows the ACS to function as a universal controller, managing a heterogeneous fleet of legacy and next-generation hardware through a single, consistent interface.

However, this extensibility often leads to a proliferation of vendor-specific parameters that diverge from the standard TR-181 definitions. Manufacturers frequently use custom paths to expose proprietary features or hardware-specific configurations or parameters. A query conducted on a production GenieACS

[3] Simple Object Access Protocol https://www.w3.org/TR/2000/NOTE-SOAP-20000508/, accessed 8 May 2026.

instance reveals the scale of this fragmentation. For instance, a TP-Link appliance contains 259 custom parameters, representing 18% of its total data model. In contrast, a ZTE device holds 491 (25%), while Huawei hardware reaches a significant 1127 (56%). Other vendors like AVM and Genexis show 767 (39%) and 61 (4%) custom parameters respectively. These variations, illustrated in Figs. 1, 2, 3, 4, and 5, force the ACS to handle thousands of unique paths that are not documented in the official TR-069 specifications.

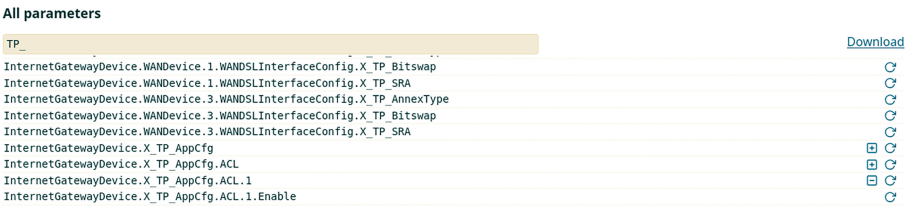


Fig. 1. Custom parameters: TP-Link

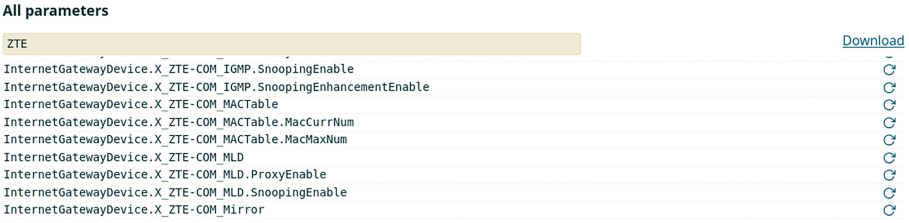


Fig. 2. Custom parameters: ZTE

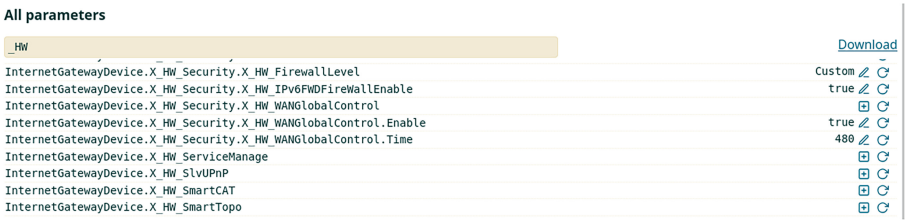


Fig. 3. Custom parameters: Huawei

Fig. 4. Custom parameters: AVM

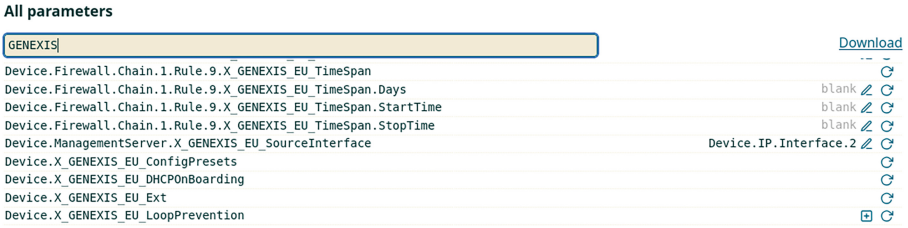


Fig. 5. Custom parameters: Genexis

3.3 Security and CPE Authentication

The security architecture of TR-069 is designed to ensure both data confidentiality and the authenticity of the managing entities, particularly when operating over public wide-area networks. At the transport layer, the protocol relies on TLS[4] to establish an encrypted tunnel. This prevents the interception of sensitive configuration parameters – such as Wi-Fi, PPPoE or SIP credentials – which would otherwise be transmitted in plain-text XML within the SOAP envelope.

Beyond transport encryption, the protocol implements an identity verification process via HTTP Digest Authentication[5]. A critical aspect of the CWMP lifecycle is the bidirectional nature of this trust: the ACS must authenticate the CPE to ensure it is managing, and sending sensitive data to, a legitimate device, while the CPE must authenticate the ACS to prevent "rogue server" attacks.

3.4 Session Lifecycle and Inform Mechanism

The operational lifecycle of a CWMP session is fundamentally "CPE-initiated", a design choice intended to navigate the complexities of NAT[6] both at a tenant-level, to manage devices behind the main router and, at a carrier level, for ISPs that implement CGNAT[7].

[4] Transport Layer Security – RFC 8446 https://datatracker.ietf.org/doc/html/rfc8446, accessed 8 May 2026.
[5] RFC 7617 (obsoletes 2617) https://datatracker.ietf.org/doc/html/rfc7617, accessed 8 May 2026.
[6] Network Address Translation.
[7] Carrier Grade NAT – RFC 6888 https://datatracker.ietf.org/doc/html/rfc6888, accessed 8 May 2026.

Every interaction between the management system and the device begins when the device sends an HTTP POST request to the ACS. The message serves as a state report that includes the device's identity[8] along with a set of "Event codes" indicating why the connection was established.

3.5 Remote Procedure Calls and Managed Download Process

On the functional layer, TR-069 leverages a well-defined collection of Remote Procedure Calls (RPCs), which should be used in order to change the parameters' tree of the device. The usual status polling and dynamic configuration are performed via `GetParameterValues` and `SetParameterValues` calls. However, Download is used as the primary way of performing more significant lifecycle changes. Unlike the simple change of a single parameter, a download operation is a stateful process that involves several stages and is tuned to transfer large volumes of information, such as firmware images or complex configuration blocks.

The effectiveness of this process is guaranteed by its asynchronous nature: in contrast with the conventional transfer method, where ACS would need to embed the file itself into the SOAP message, it uses another technique: it supplies the target CPE with the URI and authentication details. This allows CPE to accept the instruction and close the management session before initiating the file retrieval process on its own. In turn, the endpoint of the whole process occurs when the CPE finishes validation of the firmware and enters its new state; this fact is reported to the ACS with the means of a special `TransferComplete` RPC call. Offloading the actual content transfer part onto some external servers or even CDN[9] helps keep the ACS lean as the mere controller of this procedure.

4 Deployment Architecture and Implementation

While the CWMP protocol provides the guidelines on how communication should be conducted, the actual implementation of an ACS requires a powerful software architecture that is able to process highly concurrent requests. In this case, this paper focuses on GenieACS[10], as it represents the only existing FOSS solution that is currently being developed and used by the community.

As an open-source project, GenieACS provides a powerful engine but suffers from a fragmented documentation landscape. The official documentation[11] is sparse; it totals roughly 51 pages. Critical sections are incomplete. For instance, the "Roles and permissions" page contains only a "TODO" placeholder. This forces engineers to rely on the community forum[12] or tribal knowledge found

[8] defined by the Manufacturer, typically the device model, serial number, firmware version, etc.
[9] Content Delivery Network.
[10] https://genieacs.com/, accessed 8 May 2026.
[11] https://docs.genieacs.com/en/latest/, accessed 10 May 2026.
[12] https://forum.genieacs.com/, accessed 8 May 2026.

in the project's GitHub Wiki[13]. While the code is highly scalable, this lack of formal guidance remains the primary hurdle for production-grade adoption.

4.1 Architecture for Implementation and Deployment

GenieACS architecture represents the latest designs for ACSs, moving away from the monolithic systems to become a modular system based on services. The most critical component of the system is its being written in TypeScript, which provides the opportunity to apply strong type checking necessary for dealing with the complicated and often uncertain XML messages used by CWMP. Using the asynchronous nature of Node.js, GenieACS manages to process many management sessions at once without making closing SOAP connections a bottleneck.

Operational intelligence on the platform involves several specialized operations where the CWMP engine becomes the most significant one. CWMP engine acts as the protocol interface controlling the lifecycle of the session by monitoring and facilitating state transitions. The protocol interface receives `Inform` messages that need interpretation, and CWMP acknowledges them. Unlike many other software implementations, GenieACS decouples its protocol logic from the underlying storage. In this regard, GenieACS uses MongoDB as the main database for storing data. Such an option allows for accommodating different Data Models within one server. Since the parameters of devices represent trees (e.g., TR-181), using a document-oriented database like MongoDB allows storing such hierarchical data as JSON objects. This approach eliminates the "impedance mismatch", which occurs when using SQL databases. This way, ACS can handle and store different parameters from other manufacturers without a fixed structure.

Interactions with the system can further be categorized into two separate interfaces, which are referred to as the User Interface (UI) and the Northbound Interface (NBI). The User Interface is meant to provide a visual dashboard that is supposed to be used by administrators in diagnosing problems and taking manual actions. On the other hand, the NBI represents the functional part of ISP connectivity. As a RESTful interface, the NBI allows external logic such as billing services to control the ACS.

Finally, the FS module functions as a special storage place for the "Data Plane" of management, containing all of the firmware binaries and configuration blobs which are served through the Download RPC API. This particular module, as stated in 3.5, may be swapped out for any other FTP/HTTP server or even for some custom services. These modules provide a loosely coupled architecture in which all layers may scale independently, thus allowing the support of large-scale production network deployments.

[13] https://github.com/genieacs/genieacs/wiki/Example-of-a-Provisioning-Flow, accessed 8 May 2026.

4.2 Programmable Logic via Provisioning Scripts

GenieACS handles the complexity of heterogeneous device fleets through its scripting engine. This system lets engineers write TypeScript code, called "Provisions", that the ACS executes during each management session. Instead of relying on static configuration templates, the engine evaluates logic in real-time. Engineers define specific parameters and conditions to trigger these Provisions. For example, a Provision might only activate if the device belongs to a specific manufacturer or if a particular data model value is missing. This conditional execution ensures the ACS doesn't waste cycles on irrelevant tasks: a script for VoIP settings is triggered only if the device actually reports VoIP-capable hardware. By using TypeScript, the platform gives developers a familiar environment with strong typing. This helps prevent runtime errors that could brick a hardware unit. Such programmability enables "zero-touch" automation at scale. The ACS adapts its behavior based on what the CPE reports, handling vendor-specific quirks without needing a hardcoded core for every new hardware model.

4.3 Network Topology and Security Considerations

The location of the ACS in relation to the provider's network infrastructure can affect the effectiveness of the protocol and its vulnerability to attacks. In contemporary telecommunication networks, two main strategies are widely adopted. One of the most popular deployment schemes involves locating the ACS in a private management network owned by the carrier, which is located beyond CGNAT. Such a strategy shields the ACS from volumetric attacks but presents difficulties in terms of communicating commands to the Devices, since the ACS must have the proper routes and firewall permissions to communicate with them, without the need to wait for an `Inform`.

The ACS can be put online to enable device management within various network boundaries as well. Although it allows for greater flexibility, it necessitates the application of a zero-trust model for the protocol's security.

No matter what the topology of the network is, the integrity of the CWMP communication process will always be vital. Since the ACS acts as a central point of control, it becomes a highly valuable target for identity spoofing. One weakness that exists in the system is when the ACS only uses the serial number or OUI of the device to validate its identity. Without mutual authentication using TLS with certificate-based authentication or, at least, HTTP Digest Authentication, the attacker can easily spoof the identity of a legitimate device.

These implications can be quite severe. For one thing, a successful attack can give the intruder access to the full configuration payload that is stored on the targeted device. In many cases of residential deployment, the configuration payload contains confidential SIP trunking data used for VoIP communication or the passwords to access the gateway device. Apart from breaching the confidentiality of the individual, it is possible that a successful attack of this sort will also enable the attacker to gain access to the network infrastructure of the service provider itself. Consequently, the deployment of an ACS in a production

environment necessitates a rigorous security posture where the authenticity of both the CPE and the ACS are validated before any sensitive parameters are exchanged.

5 Lifecycle Management: from Zero-Touch Provisioning to Operational Maintenance

5.1 ACS Discovery and Bootstrapping Mechanisms

For a "zero-touch" deployment to succeed, the CPE must be able to resolve the ACS URI without manual configuration. In production environments, three primary mechanisms are utilized to "inform" the CPE of its management destination:

1. Factory-preset configurations: in many vertically integrated ISP models, the CPE is shipped with a hardcoded ACS URL embedded within the default firmware. While this is the most seamless method, it lacks flexibility, as migrating the device fleet to a new ACS infrastructure would require a coordinated firmware update or complex DNS redirection at the network level.
2. DHCP option 43[14] and PPPoE extensions: a more dynamic approach involves utilizing the network's access infrastructure to provide management metadata. During the initial IP address acquisition, the DHCP server can include option 43, which carries the ACS URL. Similarly, in DSL environments, vendor-specific tags within PPPoE discovery packets can serve the same purpose. This method is highly effective for "open-market" devices, as the management authority is determined by the network to which the device connects rather than the hardware's origin.
3. Manual intervention: while counter-intuitive to the goal of automation, manual entry remains a necessary fallback. This typically occurs via the built-in web management interface after logging in with elevated credentials. This method is often the only choice for small-scale businesses who cannot afford either the firmware personalization by the vendor nor direct control over the network.

5.2 ACS Admission Control and Device Adoption

The transition from a device "calling home" to being actively managed is governed by the ACS's admission control policy. In a production environment, the server must decide whether to treat an incoming `Inform` request as a legitimate registration or a potential intrusion. This is often managed through a choice between automated discovery and pre-authorized orchestration.

In the most flexible deployment models, the ACS acts as a dynamic listener, adopting any device that successfully negotiates the HTTP Basic or Digest

[14] DHCP Options and BOOTP Vendor Extensions – RFC 2312 https://datatracker.ietf.org/doc/html/rfc2132, accessed 8 May 2026.

Authentication handshake. While this "open enrollment" simplifies logistics – especially in high-volume consumer rollouts – it relies heavily on the integrity of shared secrets. Should these credentials be compromised or reverse-engineered from the firmware, the ACS becomes susceptible to "shadow" devices that could drain system resources or probe for management vulnerabilities.

To mitigate these risks, many operators moved toward a more rigorous pre-provisioning model, where the ACS is instructed to reject any CPE not explicitly registered in its database. This creates a "closed-loop" ecosystem where the device's hardware identity[15] must be white-listed before the first contact occurs. In this architecture, the GenieACS NBI becomes the primary facilitator of trust. Rather than the device initiating the relationship, the operator's CRM/ERP or billing system "introduces" the device to the ACS via a RESTful API call at the moment of subscriber activation. If an unlisted device attempts to initiate a session, the ACS rejects the request, refusing to proceed with the CWMP handshake. This integration ensures that the management plane remains strictly coupled to the business logic, guaranteeing that sensitive configuration data are reserved exclusively for verified assets.

5.3 First Provisioning

Following a successful adoption, the ACS must transform the CPE from a generic hardware state into a service-ready device. This initial provisioning phase is rarely a single-parameter push; rather, it is a nuanced synthesis of two distinct data sets. The server relies on the scripting engine (Sect. 4.2) to execute the logic required for this transition. First, the ACS applies ISP-defined global configurations, such as standardized firewall rules, DNS server assignments, and firmware baseline requirements. Second, it injects user-specific parameters retrieved from the CRM, which can include the subscriber's specific WAN configuration, Wi-Fi localization and personalization, and SIP trunking details for voice services.

In many high-security or complex carrier environments, this provisioning is handled through a dual-infrastructure handover. In this model, the device first contacts a "Provisioning ACS" located within a restricted, local provisioning network. This initial server's sole purpose is to perform the base-lining of the device, ensuring it has the minimum required firmware, security certificates and configuration. Once this internal stage is validated, the Provisioning ACS pushes a final command to update the `ManagementServer.URL` parameter, effectively handing over the CPE to the Production ACS. Considerations about the security of the Production ACS made in 4.3 remain valid.

5.4 Remote Diagnostics and Orchestration

The longevity value of a TR-069 implementation lies in the fact that after a one-time setup, it moves into a continuous period of maintenance and reactiveness. This interaction is maintained via periodic `Inform`, which is essentially the

[15] specifically its serial number and OUI.

heartbeat of the whole process that guarantees the ACS stays aware of what is happening on the device side. In this regard, forcing the CPE to check-in periodically gives MSP a longitudinal view of network status, thus providing the ACS the capability to act as an oracle and orchestrator of all activities on the device.

As for problem resolution, such check-ins provide the necessary information for resolving customer complaints without having to make an actual physical trip to their location. If a customer complains about slow connection, the ACS can refer back to previous telemetry collected during the check-in, including the noise margin of DSL lines, wireless interference levels, and high memory usage. This allow technicians to differentiate between a localized hardware failure and a broader environmental issue.

However, the effectiveness of these diagnostics is often hampered by vendor-specific implementation quirks. For example, while standardized RPCs for diagnostics exist, some manufacturers only expose a subset of the results, or worse, report a successful test even when hardware faults are present. Others may use proprietary parameter paths for standard metrics like optical signal strength (RSSI), forcing the ACS to maintain complex mapping tables for each vendor. Furthermore, the ACS can trigger real-time diagnostic RPCs, such as SelfTest-Diagnostics, effectively "virtually presence" the technician inside the customer's LAN to pinpoint bottlenecks in the service delivery chain.

Beyond individual support, the ACS manages the orchestrated evolution of the entire fleet through firmware and configuration roll-outs. Managing a mass upgrade is a delicate balancing act; pushing a new firmware to a hundred thousand devices simultaneously would create a thundering herd effect, potentially crippling the carrier's bandwidth and the ACS's own processing capacity. Another benefit of a slow roll-out is the increased chance to find peculiar configuration that causes the upgrade to fail and/or cause a service disruption. Deploying to uncorrelated setups helps testing the reliability of the new configuration and fix – if any – related issues. GenieACS addresses this by allowing operators to schedule these transitions across a distributed timeline. The ACS monitors the success of these campaigns through `TransferComplete` events, ensuring that any device failing the upgrade is flagged for manual review rather than being left in an insecure or non-functional state. This ensures that the network maintains its flexibility to evolve past initial factory settings.

6 Conclusions

From the TR-069 standards up to the current state of affairs, there have been significant changes in scope, as reflected in the increase in data models from TR-098 to TR-181, among other developments. Whereas this development allows for the increasing complexity of hardware being controlled, it brings about a considerable level of interoperability issues. CWMP may be considered a standardized technology, but its implementation in real-world scenarios depends on a lot of lab testing. There are specific vendor parameters and behaviours that do not conform to standards, which require thorough preproduction testing for every piece

of hardware or firmware update since there could be variations under the hood despite the existence of standards.

Nonetheless, it is through the adaptability of CWMP that it has found itself rising as the default engine behind the wide array of business management systems. The application has uses that extend beyond the traditional residential gateway to include managed Wi-Fi hotspots, Voice over IP phones, and many other kinds of devices. In this regard, GenieACS offers a modular, open-source alternative to proprietary systems. It handles the "thundering herd" of concurrent sessions through its asynchronous architecture and document-oriented storage. However, the documentation gap remains the primary barrier to wider FOSS adoption. With only 51 pages of official documentation and placeholders, users are forced to scavenge for implementation details on forums and wikis.

As for future trends, while TR-069 remains the industry standard for remote management systems, it is clear that it will soon be succeeded by more modern solutions such as the User Services Platform[16]. These modern frameworks aim to meet the needs of continuous communication and multiple controllers.

References

1. Zheng, L., Hu, Y., Chen, S.: Research and application of cwmp in distributed network management system. In: 2012 International Conference on Computer Science and Service System, pp. 647–650 (2012). https://doi.org/10.1109/CSSS.2012.167
2. Barczak, B., Kafel, T., Magliocca, P.: Network approaches and strategic management: exploration opportunities and new trends. J. Entrepreneurship Manage. Innov. **17**, 7–35 (2021). https://doi.org/10.7341/20211731
3. Rachidi, H., Karmouch, A.: A framework for self-configuring devices using TR-069. In: 2011 International Conference on Multimedia Computing and Systems (2011), pp. 1–6 (2011). https://api.semanticscholar.org/CorpusID:264024769
4. Stusek, M., Masek, P., Kovac, D., Ometov, A., Hosek, J., Kröpfl, F., Andreev 0001, S.: Remote management of intelligent devices: using TR-069 protocol in IoT. In: 39th International Conference on Telecommunications and Signal Processing, TSP 2016, Vienna, Austria, June 27–29, 2016, pp. 74–78. IEEE (2016). isbn: 978-1-5090-1288-6. https://doi.org/10.1109/TSP.2016.7760832
5. Stusek, M., et al.: A novel application of CWMP: an operatorgrade management platform for IoT. Int. J. Adv. Telecommun. Electrotechn. Signals Syst. **5**(3), 171–177 (2016)

[16] https://www.broadband-forum.org/pdfs/tr-369-1-3-0.pdf, accessed 8 May 2026.

Less is More: XAI-Driven Feature Selection for Efficient Molecular Property Prediction

Jonas Lucas Durão[1], Luís G. Dias[2], and Marcos G. Quiles[1(✉)]

[1] Institute of Science and Technology, Federal University of São Paulo, Av. Cesare Lattes, 1201, São José dos Campos, SP 12247-014, Brazil
{jonas.durao,quiles}@unifesp.br

[2] Chemistry Department, FFCLRP, University of São Paulo, Ribeirão Preto, SP 14040–901, Brazil
lgdias@ffclrp.usp.br

Abstract. Molecular descriptors are commonly employed to build predictive models for molecular properties; however, their usage comes with several challenges. The exponential growth of available descriptors and the complexity of the required algorithms often result in opaque, 'black-box' models that lack interpretability. Traditional feature selection methods focus on statistical optimization, but they often fail to capture the physicochemical drivers and complex interactions central to the chemical domain. In this work, we propose an explainability-driven feature selection framework that uses Shapley Additive exPlanations (SHAP) to identify the minimal necessary subset of descriptors for predicting some molecular properties of the QM9 dataset. This framework serves not only as a dimensionality reduction tool but also as a validation mechanism that ensures the model's logic is consistent with the laws of chemistry. By iteratively reducing the initial set of 1057 Mordred descriptors, we show that feature selection using SHAP values might outperform traditional methods in several scenarios, such as predicting Rotational Constants C and B, Internal Energy at 0 K, and Electronic Spatial Extent. For other molecular properties, the SHAP-guided feature selection delivers similar performance to baseline feature selection methods, such as LASSO, Random Forest, CFS, as well as Pearson and Spearman correlations. Our findings show that the feature space can be compressed by more than 90% with no loss in predictive capability. For some thermodynamic and electronic properties, the SHAP-guided feature reduction has also improved generalization performance due to the elimination of non-informative features. More importantly, the variables surviving this selection process are not simply mathematical conveniences. They map directly onto recognized physicochemical theory, pinpointing the precise electronic and topological mechanisms responsible for molecular behavior.

Keywords: Explainable Artificial Intelligence · XAI · Feature Selection · QM9 dataset · Machine Learning · Molecular Property Prediction

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 204–217, 2026.
https://doi.org/10.1007/978-3-032-30491-9_14

1 Introduction

One of the challenges in machine learning is data dimensionality; over the years, the number of features has increased, and with that came the need for algorithms and tools to address this problem [6]. Therefore, the main objective of feature selection is to reduce the number of input features and lower computational costs while enhancing predictive performance and generalization capabilities of models [15,16].

Traditional feature selection approaches, such as filter, wrapper, and embedded methods [3], often face significant limitations. They are effective in optimizing error metrics, but traditional methods disregard the model's internal logic, failing to reveal the behavior and meaning behind the decisions of the "black box" models [1,2]. Conventional protocols frequently fail to delineate intricate, non-linear interactions across the feature space. Compounding this limitation is the absence of a cohesive metric capable of reconciling raw predictive power with structural transparency [4].

In the Chemistry domain, that requires the understanding of the physical drivers behind a prediction, the lack of interpretation of features selected by traditional feature selection methods makes the final model difficult to understand. This gap can be filled with a shift towards methods that can provide explanations of the results to justify the inclusion of specific descriptors in the input data [5].

In that sense, Explainable Artificial Intelligence (XAI) methods have emerged as powerful tool for feature selection [8]. The integration of XAI based methods into the feature selection phase allows the usage of model explanations to find significant features, and this makes the selection problem more understandable [10]. In this work, we employ SHAP [7] as the core XAI method for feature selection. The choice of SHAP is not only because its a traditional method in the XAI literature, but unlike other methods, SHAP provides a mathematically fair allocation of importance, specifically because of its properties: Efficiency, Symmetry, Null Player, and Linearity [13].

The extensive characterization of molecular topology and physicochemistry that Mordred offers generates a vast feature space (1,613 2D and 213 3D descriptors) populated by redundancy and multicollinearity. Consequently, separating the contribution of each descriptor is a difficult challenge. In this context, because of the SHAP properties, it can act as a critical filter to identify the ideal subset of chemical descriptors necessary for predicting each property. Leveraging these properties allows us to fairly discriminate between important and irrelevant features within the initial feature set.

The application of XAI driven feature selection advances property prediction by providing a different approach to identifying descriptors that not only can improve the final prediction but also show the chemically relevant descriptors for properties. This work bridges the gap between the high accuracy of complex "black-box" models and the necessity for chemically meaningful interpretations of the outputs. By adopting this "Less is More" idea, we ensure an ideal trade-off

between predictive performance, computational efficiency, and the interpretability required for scientific discovery in Chemistry [14].

2 Methodology

2.1 QM9 Dataset and Molecular Representation

For building the models, we used the QM9 dataset [12]. This dataset is widely recognized as a gold standard in the fields of quantum chemistry and machine learning. It consists of approximately 134,000 stable organic molecules, composed of up to nine heavy atoms of four species (C, O, N, F), in addition to hydrogen.

The physicochemical properties of each molecule in QM9 were originally calculated using Density Functional Theory (DFT) at the B3LYP/6-31G(2df,p) level of theory. In this work, all available energetic, electronic, and thermodynamic properties were considered, with the exception of heat capacity (C_v).

2.2 Featurization

The numerical representation of chemical structures was performed using the Mordred library [9]. Unlike approaches based solely on binary fingerprints, Mordred calculates continuous physicochemical descriptors and topological indices based on graph theory. This enhances the model's ability to capture valuable information about the molecules [11].

For this study, an initial vector containing 1057 descriptors was generated for each molecule of QM9. The full 1826 descriptors from the Mordred was not used because some exhibited errors or NaN values when calculating, so they were discarded remaining 1057 descriptors. This set served as the baseline for subsequent feature selection and optimization processes.

2.3 Experimental Setup

To rigorously evaluate the proposed explainability-driven feature selection pipeline, we established a robust validation framework. Prior to model training, a strict preprocessing protocol was applied to the feature space $\mathcal{X}$. Features were standardized to zero mean and unit variance (z-score normalization) to mitigate scale disparity issues inherent in heterogeneous physicochemical descriptors. The dataset was randomly split into training 80%, validation 10%, and test 10% sets using a fixed random seed to ensure reproducibility.

All feature selection methods shared the same underlying predictive network. The model employed for property prediction is a Multilayer Perceptron (MLP). To ensure a correct assessment of feature importance, regardless of the target property's complexity, we treated the model architecture as a hyperparameter search problem rather than a fixed topology. We employed the Tree-structured Parzen Estimator (TPE) algorithm, via the Optuna framework, to optimize the architecture and training hyperparameters for each of the 15 properties independently. The search space encompassed:

1. **Architecture:** Depth (1 to 4 hidden layers) and width (64 to 1024 neurons per layer).
2. **Optimization:** Learning rate (10^{-5} to 10^{-1}), batch size (128 to 1024), and optimizer selection (Adam, RMSprop, SGD, AdamW).
3. **Regularization:** Dropout rates (0.0 to 0.75) and weight decay (L_2 regularization).
4. **Objective Function:** A categorical choice between L_1 (MAE), MSE, and Huber (Smooth L_1) loss functions.

Early stopping with a patience of 10 epochs was utilized to prevent overfitting during the optimization trials. All experiments were conducted on a workstation equipped with two 12-core Intel Xeon Silver 4214R processors, 256 GB of RAM, and 2 NVidia RTX A6000 GPUs, each with 48 GB of memory.

2.4 Feature Selection and Degradation Protocol

We implemented an iterative feature degradation protocol Fig. 1 to assess the explanation of the identified feature subsets. The core utilization of this protocol is to determine the minimal subset of features $S \subset \mathcal{D}$ required to maintain predictive performance comparable to the full feature set, in which $\mathcal{D}$ represents the full feature set. The iterative pruning process is defined by:

$$S_k = S_{k-1} \setminus \{d_j \in S_{k-1} \mid \Omega_j^{(k)} < \tau_k\} \tag{1}$$

where S_k is the feature subset at iteration k, and $\Omega_j^{(k)}$ represents the importance score of descriptor d_j calculated by the chosen method. Following each pruning step, the model f is retrained on the reduced subspace $\mathcal{X}_{S_k}$ until the termination criterion is met.

The degradation process was terminated at 20 features for all methods based on the importance score, features ranked below the 20th position exhibited importance values less than 1%, of the top-ranked feature, indicating negligible contribution to model predictions.

XAI-Driven Selection. We used SHAP global explanations from the toolbox ChemXAI. For a given trained MLP model f, the importance of feature j is quantified as the mean absolute SHAP value across the training cohort:

$$I_j^{SHAP} = \frac{1}{N} \sum_{i=1}^{N} |\phi_j(f, x^{(i)})| \tag{2}$$

where ϕ_j represents the contribution of feature j to the prediction of sample $x^{(i)}$. In each iterative step, we keep the top 90% of the features with the highest global importance.

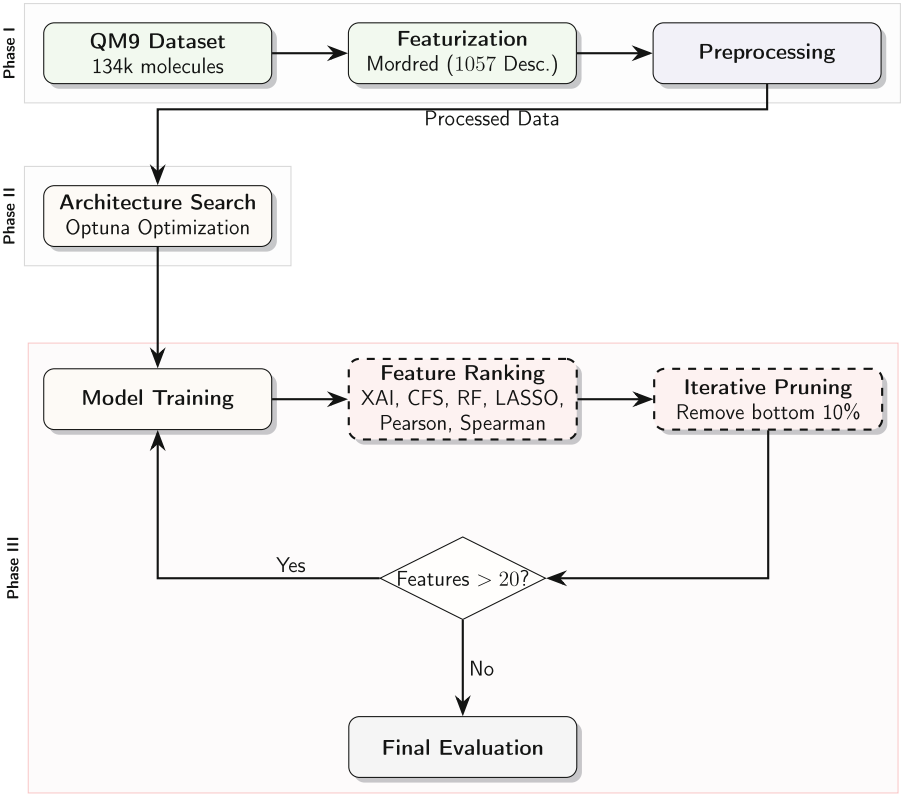


Fig. 1. Iterative Feature Pruning and Model Re-training Workflow.

Baseline Methods. To validate the efficacy of the SHAP-based selection, we benchmark against five well-established feature selection baselines representing filter and embedded methods:

LASSO (Least Absolute Shrinkage and Selection Operator): An embedded method that enforces sparsity through L1 regularization. To ensure computational efficiency in the high-dimensional Mordred space, we utilized a fixed regularization parameter $\alpha = 0.001$ and a maximum of 2,000 iterations. Features are ranked based on the absolute values of their coefficients.

Random Forest Importance (RF): A non-linear embedded method using an ensemble of decision trees. We utilized 50 estimators with a maximum depth of 10 to capture non-linear interactions while maintaining memory safety. Feature importance is computed based on the Mean Decrease in Impurity (MDI).

Pearson Correlation: A linear filter method that measures the strength and direction of the linear relationship between each descriptor and the target property. This serves as a baseline for detecting strictly linear dependencies.

Spearman Rank Correlation: A non-parametric filter method that assesses monotonic relationships. It is implemented by calculating the Pearson correlation

on the rank-transformed variables, allowing for the detection of non-linear but monotonic dependencies.

CFS (Correlation-based Feature Selection): Functioning as a proxy for CFS, this method utilizes the F-statistic via univariate linear regression (f_regression). It ranks features by evaluating the individual effect of each regressor on the target property.

3 Results and Discussion

For the results presented below, we evaluate the global effectiveness of the XAI-based feature selection methodology across all molecular properties; then, we compare our approach against traditional baseline methods and provide detailed case studies demonstrating the chemical interpretability of the selected features.

3.1 Global Reduction Analysis

With feature degradation using SHAP as the selection method, 11 of the 14 properties (Fig. 2) exhibited improved performance compared to the full feature set of 1057 descriptors. The applied methodology was effective in enhancing model performance by reducing test error, and it successfully eliminated redundant features that added no predictive information while increasing model complexity.

The results depicted in Fig. 2 reveal a heterogeneous response of molecular properties to the XAI feature selection. This result shows a clear improvement for thermodynamic properties that suggests a presence of non-essential features for these particular targets. Acting as a filter, SHAP distills the descriptors into an optimal subset that actively isolates the primary energetic drivers. In contrast, electronic properties such as Dipole Moment (μ) and orbital energies ϵHOMO and ϵLUMO showed no improvement. This indicates a higher intrinsic dimensionality for electronic targets, where a more diverse set of descriptors is needed to account for the variability of the electron density in the dataset, making them less susceptible to performance gains via aggressive feature pruning.

For the bottom 6 properties in Table 2, the method achieved more than a 50% reduction over the original descriptor set without sacrificing predictive capacity, illustrating that the degradation process can act as a denoising step when guided by SHAP. When the loss ratio of first degradation to current degradation stays above unity during feature removal, the XAI-based selection exceeds the performance of the initial full-feature baseline. Figure 3a illustrates this process for the property Rotational Constant C. These results confirm that the proposed approach operates as an active, targeted filter, distinctly moving beyond blind statistical dimensionality reduction.

3.2 Evolution of Test Loss

The optimized models achieved low test losses across all properties, demonstrating robust generalization. The evolution of test loss with feature reduction, however, depends strongly on the nature of the target property (Table 1, Table 2).

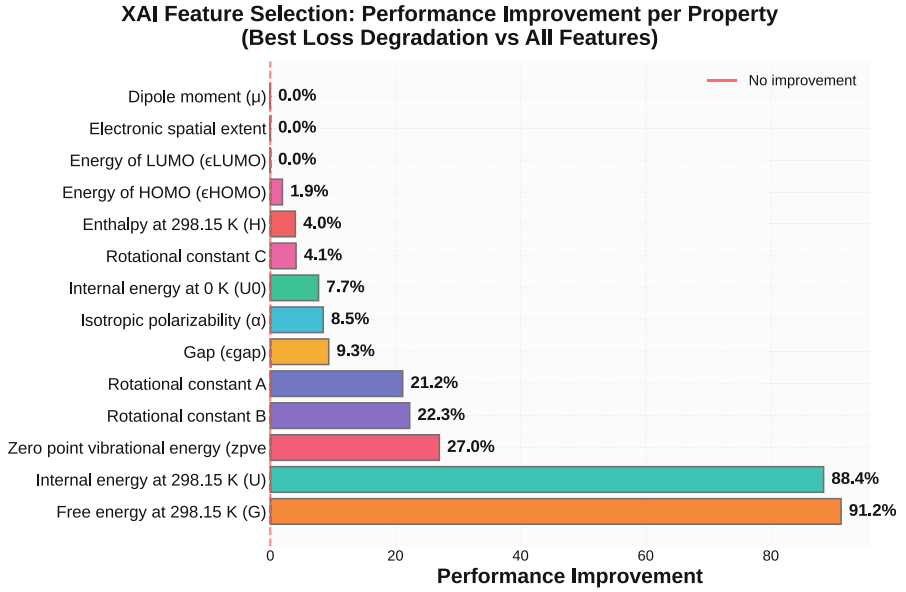
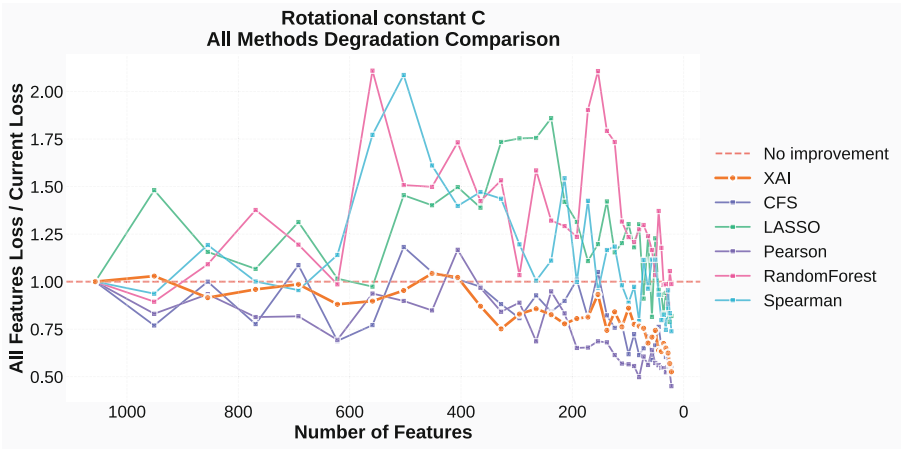


Fig. 2. A comparative performance gains of XAI-driven feature selection across all QM9 properties. The chart illustrates the improvement when moving from the full feature set (1057 descriptors) to the optimal feature subset identified during the iterative degradation protocol.

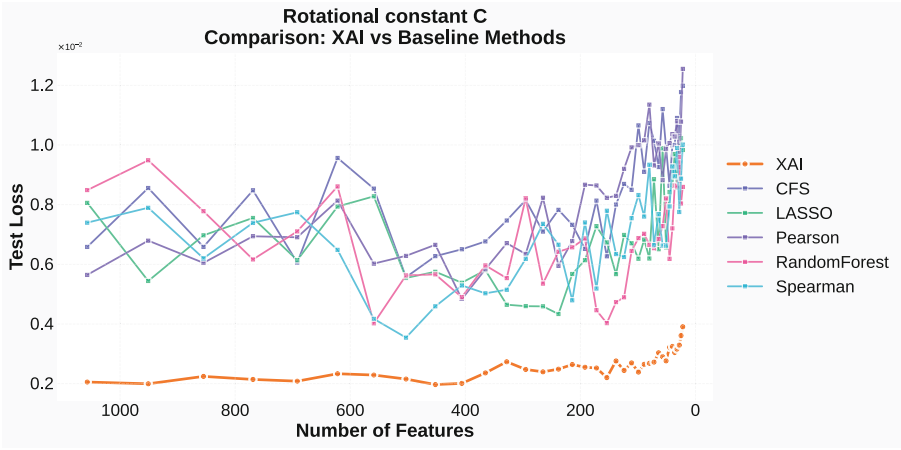
Electronic and vibrational properties (e.g., HOMO-LUMO gap, zero-point vibrational energy) exhibit a clear dependence on descriptor count; the test loss tends to rise as features are pruned, and a larger subset is needed to capture their behavior.

In contrast, thermodynamic and rotational properties reach an elbow point in the end of the degradation process. For these, the test loss stabilizes or oscillates at a low level even under drastic dimensionality reduction, often requiring fewer than 30 features to achieve the best performance. This suggests that thermodynamic and rotational behavior can be captured by a compact set of descriptors. The XAI approach showed that it can offer a favorable trade-off for vibrational and structural targets by retaining a larger subset than some baselines while prioritizing chemically meaningful descriptors and maintaining predictive accuracy at the elbow of the curve. Figure 3b illustrates this behavior for Rotational Constant C, where XAI attains the lowest test loss among all methods and has a more stable oscillation as the feature set is reduced.

In the detailed analysis of the Fig. 3, the superiority of the XAI approach is observed at two levels. In plot Fig. 3a, the XAI curve consistently remains above the 'No improvement' baseline, reaching efficiency peaks even under severe dimensionality reduction. Most notably, plot Fig. 3b demonstrates that the test losses for the baselines increase drastically upon crossing the 200 features threshold; the SHAP based selection maintains a low and stable test error, increasing



(a)



(b)

Fig. 3. Comparative analysis of feature selection impact on the Rotational Constant C prediction. (a) Displays the relative performance expressed as the test loss ratio ($Loss_{full}/Loss_{reduced}$) as features are iteratively removed; (b) Illustrates the absolute performance evolution of the Test Loss across the degradation process for XAI and baseline methods. The shaded areas represent the standard deviation across multiple experimental runs.

at a lower rate than the baselines. This indicates that the algorithm identified geometrically fundamental descriptors that purely statistical methods discarded prematurely.

Table 1. Comparison of Minimum Test Loss (Mean ± Std. Dev.).

Sym.	XAI	CFS	LASSO	Pearson	RF	Spearman
μ [$\times 10^{-1}$]	1.178 ± 0.003	1.164 ± 0.029	$\mathbf{1.146 \pm 0.013}$	1.151 ± 0.005	1.153 ± 0.010	1.155 ± 0.015
R_e [$\times 10^{-2}$]	$\mathbf{1.19 \pm 0.02}$	1.20 ± 0.01	1.20 ± 0.05	1.20 ± 0.03	1.21 ± 0.03	1.21 ± 0.05
ϵ_H [$\times 10^{-2}$]	3.77 ± 0.12	3.51 ± 0.05	3.63 ± 0.06	3.62 ± 0.19	$\mathbf{3.50 \pm 0.02}$	3.63 ± 0.02
ϵ_L [$\times 10^{-3}$]	9.93 ± 0.31	9.83 ± 0.11	$\mathbf{9.60 \pm 0.24}$	9.70 ± 0.07	9.63 ± 0.05	9.78 ± 0.14
H [$\times 10^{-5}$]	5.63 ± 0.21	0.99 ± 0.06	3.45 ± 0.59	1.09 ± 0.26	1.78 ± 0.51	$\mathbf{0.64 \pm 0.04}$
G [$\times 10^{-5}$]	2.65 ± 0.63	$\mathbf{0.51 \pm 0.11}$	2.00 ± 0.85	1.00 ± 0.73	2.02 ± 0.44	0.93 ± 0.87
ϵ_{gap} [$\times 10^{-2}$]	1.60 ± 0.11	1.61 ± 0.13	1.60 ± 0.08	1.65 ± 0.12	$\mathbf{1.54 \pm 0.05}$	1.65 ± 0.13
U_0 [$\times 10^{-4}$]	$\mathbf{4.86 \pm 0.53}$	5.06 ± 0.44	5.16 ± 0.12	5.81 ± 0.29	5.31 ± 0.13	5.73 ± 0.32
U [$\times 10^{-5}$]	2.87 ± 1.26	3.37 ± 0.25	$\mathbf{0.71 \pm 0.86}$	4.28 ± 1.85	2.66 ± 1.53	3.76 ± 0.51
α [$\times 10^{-3}$]	5.46 ± 0.37	5.66 ± 0.49	$\mathbf{5.32 \pm 0.21}$	5.85 ± 0.19	5.71 ± 0.14	5.61 ± 0.40
A [$\times 10^{-7}$]	9.07 ± 0.03	3.20 ± 0.05	2.90 ± 0.03	2.97 ± 0.02	3.03 ± 0.02	$\mathbf{2.87 \pm 0.01}$
B [$\times 10^{-3}$]	$\mathbf{3.20 \pm 0.21}$	4.82 ± 0.81	3.93 ± 0.75	4.44 ± 0.29	4.22 ± 1.15	5.29 ± 0.64
C [$\times 10^{-3}$]	$\mathbf{1.97 \pm 0.09}$	5.57 ± 2.03	4.33 ± 1.27	4.84 ± 0.95	4.02 ± 0.77	3.54 ± 0.51
E_{zpve} [$\times 10^{-4}$]	3.3 ± 0.2	3.2 ± 0.2	3.2 ± 0.0	$\mathbf{2.9 \pm 0.1}$	3.2 ± 0.3	3.0 ± 0.2

Properties: μ: Dipole moment; R_e: Electronic spatial extent; ϵ_H: Energy of HOMO; ϵ_L: Energy of LUMO; H: Enthalpy (298.15 K); G: Free energy (298.15 K); ϵ_{gap}: Gap energy; U_0: Internal energy (0 K); U: Internal energy (298.15 K); α: Isotropic polarizability; A, B, C: Rotational constants; E_{zpve}: Zero point vib. energy. **Note:** Multipliers in brackets (e.g., [$\times 10^{-2}$]) apply to both mean and std. dev.

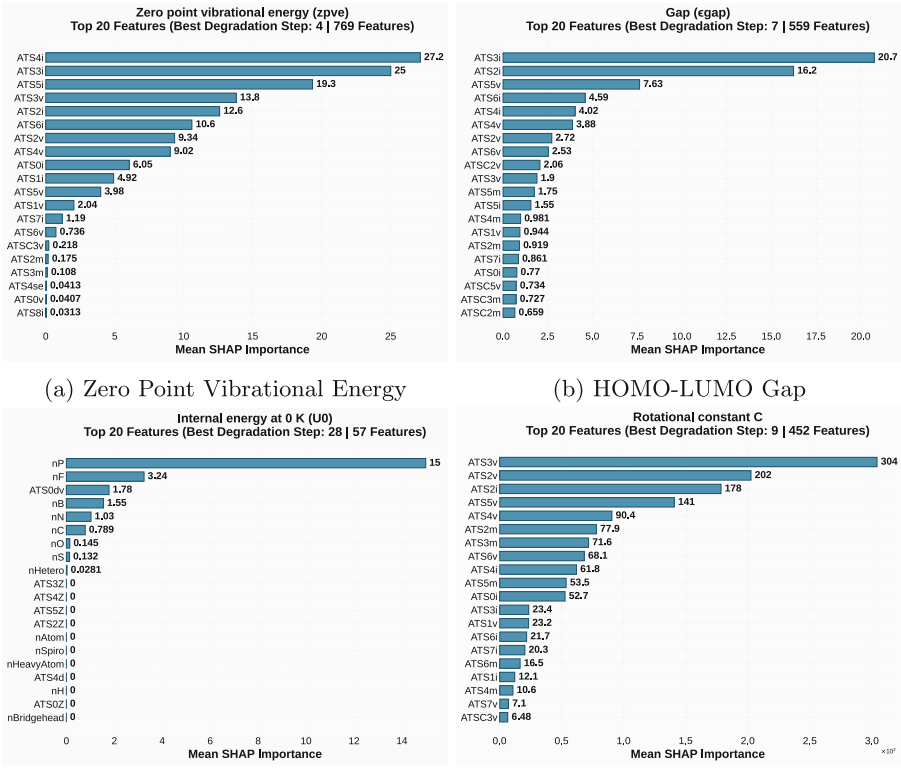
3.3 Case Study

The SHAP values across Fig. 4 reveal a big disparities in sensitivity scales among the property classes. HOMO-LUMO Gap and ZPVE, exhibit importance scores between 20 and 30. In comparison, the Rotational Constant C values are on a scale roughly ten times larger, with the top descriptor reaching a peak value of 304.

This order of magnitude difference highlights the extreme sensitivity of molecular inertia to specific topological volume distributions. This means that, for rotational constants, the volume of the molecule, $ATSXv$, reflects the prediction more directly than other properties, which require more descriptors to explain their behavior. A similar trend appears for Internal Energy at 0 K, where the phosphorus count, nP, is nearly five times as influential as fluorine. These steep gradients in importance justify the degradation of the features. Since a small set of high valuable descriptors captures the target, the model can undergo aggressive feature pruning without significant accuracy loss.

The top 20 SHAP rankings (Fig. 4) also reveal how the selected descriptors align with physicochemical intuition. For electronic and vibrational targets (Fig. 4a, Fig. 4b), the model attributes high importance to the Autocorrelation of Topological Structure (ATS) descriptors, weighted by ionization potential ($ATSXi$) and van der Waals volume ($ATSXv$). The prominence of $ATS3i$, $ATS2i$, and $ATS4i$ is consistent with the sensitivity of gap and zero-point energy to charge distribution and molecular size at short topological distances. Thermo-

dynamic properties are driven by descriptors that reflect elemental composition and electronic distribution. For Internal Energy at 0 K (Fig. 4c), heteroatom counts (nP, nF, nB, nN) rank among the most important features, in line with the dependence of formation energy on atomic identity and bond contributions.



(a) Zero Point Vibrational Energy

(b) HOMO-LUMO Gap

(c) Internal Energy at 0 K (U_0)

(d) Rotational Constant C

Fig. 4. Global feature importance ranking based on Mean SHAP values for four selected QM9 properties. Each panel illustrates the top 20 descriptors identified at the best degradation step: (a) Zero Point Vibrational Energy, (b) HOMO-LUMO Gap, (c) Internal Energy at 0 K, and (d) Rotational Constant C. The x-axis represents the average absolute contribution of each feature to the model's output across the test set, highlighting the dominance of specific topological and electronic indices in governing these molecular properties.

When XAI does not achieve the lowest test loss, the chosen features remain chemically meaningful. For Dipole moment, the baselines such as Pearson or LASSO achieved better test errors, and the dominant SHAP features (e.g., $ATS2v$, $ATS2i$) still reflect charge distribution and size. Thus, the model identifies interpretable drivers despite of minor differences in predictive performance.

In contrast, for Rotational Constant C, XAI attains the lowest test loss (Table 1) and the descriptors (Fig. 4d) agree with the geometric and inertial drivers of rotational constants. Figure 3a shows that the loss ratio remains stable and above 1.0 for the XAI curve as features are removed, confirming that SHAP based feature selection acts as denoising rather than arbitrary removal.

3.4 Comparisons to Feature Selection Methods

Comparison with traditional baselines (Table 1, Table 2) highlights both the competitiveness and the distinctive value of the XAI approach. For Internal Energy at 0 K (U0), XAI achieved the best loss and required only 57 features compared to 406 for LASSO. This example illustrates that XAI offers a viable alternative to LASSO, with the added benefit that the retained descriptors align with chemical intuition (Fig. 4c).

Table 2. Comparison of the Number of Selected Features for the Lowest Test Loss across XAI and Baseline Methods for QM9 Properties.

Property	XAI	CFS	LASSO	Pearson	RF	Spearman
μ	1057	769	1057	855	622	855
R_e	1057	769	769	951	692	1057
ϵ_H	855	951	692	1057	214	622
ϵ_L	1057	951	1057	855	951	951
H	951	22	22	28	25	22
G	64	22	25	28	22	25
ϵ_{gap}	559	622	192	452	214	559
U_0	57	51	406	1057	855	64
U	45	22	25	22	28	28
α	951	855	265	1057	855	951
A	192	855	769	769	40	28
B	99	452	124	192	214	855
C	452	406	111	214	406	503
E_{zpve}	769	559	951	855	1057	769

Legend: RF: Random Forest. μ: Dipole moment; R_e: Electronic spatial extent; ϵ_H: Energy of HOMO; ϵ_L: Energy of LUMO; H: Enthalpy (298.15 K); G: Free energy (298.15 K); ϵ_{gap}: Gap energy; U_0: Internal energy (0 K); U: Internal energy (298.15 K); α: Isotropic polarizability; A, B, C: Rotational constants; E_{zpve}: Zero point vib. energy.

For Internal Energy at 298.15 K (U), XAI achieved the third highest performance gain with one of the smallest feature sets. The network complexity

dropped from 1,253,057 parameters (1057 features) to 346,305 parameters (45 features), a 72.4% reduction without compromising accuracy. For larger architectures, such savings would translate into substantial gains in training time and memory. Moreover, even in cases where another method reports a marginally lower test loss, XAI provides transparency over which features drive the prediction, shifting the focus from purely statistical optimization to chemically grounded, understandable models.

The descriptors selected by the different methods reveal a clear distinction between the baselines and XAI. LASSO, CSF, and Random Forest frequently rely on generic topological indices and constitutional counts (e.g., nAtom, SpAbs_A) for most of the properties, showing that it does not search for chemically meaningful features that describe the property. For the XAI selected features, SHAP gives more importance and highlights Moreau-Broto autocorrelations weighted by ionization potential and van der Waals volumes (ATS4i, ATSC3v), which accurately reflect spatial electron distribution for electronic properties. Similarly, for thermodynamic properties such as Enthalpy (H), XAI selects descriptors related to mass distribution and internal connectivity (ATS3m, ATS5i) rather than purely statistical correlations.

4 Conclusions

We have demonstrated that explainability driven feature selection based on SHAP attributions substantially reduced the original set of Mordred descriptors for QM9 molecular property prediction. This approach maintains or improves predictive performance, with 11 of 14 targets benefiting from SHAP-based reduction compared to the 1057-feature initially. Six properties achieved a reduction of over 50% while simultaneously lowering test error. Thermodynamic and rotational properties reached optimal performance with fewer than 30 descriptors, whereas electronic and vibrational targets required larger subsets. This heterogeneity reflects the underlying physics, as global descriptors are sufficient for thermodynamic and rotational behavior, and it demonstrates that energy gaps and zero-point energy depend on finer electronic and structural details.

The SHAP based approach matches traditional methods like LASSO, Random Forest importance, CFS, Pearson Correlation, and Spearman Ranking on several properties, including Rotational Constants C and B, Electronic Spatial Extent, and Internal Energy at 0 K. It remains competitive for Internal Energy at 298.15 K using significantly fewer features. The retained subsets including ATS type autocorrelations for electronic targets, heteroatom counts for thermodynamic energies, and geometry related descriptors for rotational constants align with physicochemical intuition to provide understandable models. Even when alternative methods provide marginally lower test loss, the interpretability and trust of the SHAP selected set offer a superior trade-off.

These results establish XAI as a functional tool for feature selection. Integrating iterative degradation with global SHAP importance offers compact, chemically meaningful descriptor sets that reduce computational costs without sacrificing accuracy. Future work may extend this framework to other descriptor

families, model architectures, and larger datasets where efficiency and interpretability are critical.

5 Data and Software Availability

The QM9 dataset is available at https://quantum-machine.org/datasets/. The source code implementing the XAI-driven feature selection framework and the associated validation pipelines will be made publicly available in a GitHub repository upon acceptance of this manuscript for publication.

Acknowledgments. The authors gratefully acknowledge support from FAPESP (São Paulo Research Foundation) and Shell, projects Numbers 2017/11631-2, 2022/09285 – 7, and 2025/01862 – 3, the Brazilian National Council for Scientific and Technological Development (CNPq), grant 310377/2025-0, and the strategic importance of the support given by ANP (Brazil's National Oil, Natural Gas and Biofuels Agency) through the R&D levy regulation.

References

1. Adadi, A., Berrada, M.: Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access **6**, 52138–52160 (2018). https://doi.org/10.1109/ACCESS.2018.2870052
2. Arrieta, A.B., et al.: Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Inf. Fusion **58**, 82–115 (2020). https://doi.org/10.1016/j.inffus.2019.12.012
3. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. Comput. Electr. Eng. **40**(1), 16–28 (2014). https://doi.org/10.1016/j.compeleceng.2013.11.024
4. Fryer, D., Strümke, I., Nguyen, H.: Shapley values for feature selection: the good, the bad, and the axioms. IEEE Access **9**, 144352–144360 (2021). https://doi.org/10.1109/ACCESS.2021.3119110
5. Gomes-Esteves, J., Mendes, G., Faria, P., Vale, Z.: Leveraging explainable artificial intelligence in solar photovoltaic mappings: model explanations and feature selection. Energies **18**(5), 1282 (2025). https://doi.org/10.3390/en18051282
6. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003)
7. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17, Red Hook, NY, USA, pp. 4768–4777. Curran Associates Inc. (2017)
8. Marcílio, W.E., Eler, D.M.: From explanation to feature selection: an approach based on local explanations. Comput. Phys. Commun. **256**, 107455 (2020). https://doi.org/10.1016/j.cpc.2020.107455
9. Moriwaki, H., Tian, Y.S., Kawashita, N., Takagi, T.: Mordred: a molecular descriptor calculator. J. Cheminform. **10**(4) (2018). https://doi.org/10.1186/s13321-018-0258-y

10. Munawar, M.A., et al.: Optimizing solar radiation prediction with ANN and explainable AI-based feature selection. Axioms **13**(7), 263 (2024). https://doi.org/10.3390/axioms13070263
11. Pinheiro, G.A., Mucelini, J., Soares, M.D., Prati, R.C., Da Silva, J.L.F., Quiles, M.G.: Machine learning prediction of nine molecular properties based on the smiles representation of the qm9 quantum-chemistry dataset. J. Phys. Chem. A **124**(47), 9854–9866 (2020). https://doi.org/10.1021/acs.jpca.0c05969
12. Ramakrishnan, R., Dral, P.O., Rupp, M., von Lilienfeld, O.A.: Quantum chemistry structures and properties of 134 kilo molecules. Sci. Data **1**, 140022 (2014). https://doi.org/10.1038/sdata.2014.22
13. Rozemberczki, B., et al.: The shapley value in machine learning. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22), pp. 5572–5579 (2022). https://doi.org/10.24963/ijcai.2022/782
14. Schreiber, M., Hochberg, K.: Explainable machine learning and predictive statistics for sustainable photovoltaic power prediction on areal meteorological variables. Appl. Sci. **15**(14), 8005 (2025). https://doi.org/10.3390/app15148005
15. Zacharias, J., Zahn, M., Chen, J., Hinz, O.: Designing a feature selection method based on explainable artificial intelligence. Electron. Mark. **32**(4), 2175–2204 (2022). https://doi.org/10.1007/s12525-022-00608-1
16. van Zyl, C., Ye, X., Naidoo, R.: Harnessing explainable artificial intelligence for feature selection in time series energy forecasting: a comparative analysis of grad-cam and shap. Appl. Energy **353**, 122079 (2024). https://doi.org/10.1016/j.apenergy.2023.122079

High Performance Computing and Networks

AMMOF: An Adaptive Multi-metric Objective Function for Enhanced RPL Routing in Industrial IoT Networks

Lyes Ibersiene(✉), Badis Djamaa, Mustapha Reda Senouci, Ahmed Yadine Merabet, and Oussama Zahali

Ecole Militaire Polytechnique, BP 17, 16046 Bordj El Bahri, Algiers, Algeria
lyes.ibersiene@gmail.com

Abstract. The Routing Protocol for Low-Power and Lossy Networks (RPL) is the routing protocol of choice for Low-Power and Lossy Networks (LLNs) and constitutes the standard routing solution for 6TiSCH-based Industrial Internet of Things (IIoT) networks. RPL relies on a parent selection and switch mechanism, managed by Objective Functions (OFs) that critically impacts network performance in terms of latency, reliability, and energy consumption. However, default OFs such as OF0 and MRHOF are based on single metrics, hop count and expected transmission count, respectively. In this paper, we investigate the limitations of these two default solutions and propose AMMOF, an Adaptive Multi-Metric Objective Function that jointly considers link quality, residual energy, and queue occupancy to optimise parent selection decisions. We also design a distance-aware weight tuning mechanism to further enhance the parent selection process. Through extensive Cooja simulations, we demonstrate that AMMOF significantly improves network performance in terms of packet delivery ratio, end-to-end delay, energy efficiency, and topology stability compared to OF0 and MRHOF in 6TiSCH industrial scenarios.

Keywords: RPL · 6TiSCH · Parent Switch · Objective Function · IIoT · Multi-Factor Routing · TSCH

1 Introduction

The Industrial Internet of Things (IIoT) revolutionizes modern industrial environments by allowing dense and reliable interconnections between sensors, actuators, and controllers. Several industrial applications such as factory automation, robotic assembly lines, smart grid protection, and chemical plant safety monitoring, have been made possible thanks to this technology, however they require stringent conditions on the underlying communication infrastructure, notably in terms of high reliability, bounded end-to-end latency, and energy efficiency [1].

The 6TiSCH architecture [2] has emerged as an eminent and distinguished solution for IIoT networks. It combines the deterministic medium access control

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 221–235, 2026.
https://doi.org/10.1007/978-3-032-30491-9_15

of IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) with the scalability of IPv6. Within 6TiSCH architecture, the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [3] is the standard routing protocol. RPL is responsible for building and maintaining a Destination Oriented Directed Acyclic Graph (DODAG) rooted at the edge router.

The *parent switch* is a central mechanism in RPL, whereby a node selects its preferred parent using an Objective Function (OF). The impact of this decision is critical in terms of the quality of the routing paths and, consequently, on the overall network performance. OF0 and MRHOF are the two default OFs standardized by the IETF. OF0 [4] minimizes hop count, while MRHOF [5] relies on the Expected Transmission Count (ETX). Despite these single-metric approaches are simple and effective under certain conditions, they suffer from significant limitations in complex industrial environment where multiple conflicting objectives must be considered and simultaneously optimized.

Contributions. This paper introduce the main contributions:

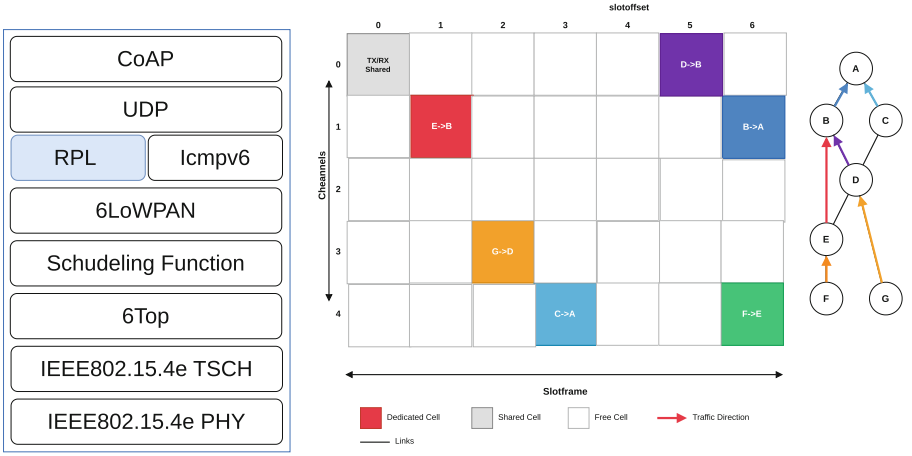
- An analysis of the default standardized OFs, OF0 and MRHOF limitations in 6TiSCH industrial environments.
- A novel multi-factor Objective Function that RSSI/LQI, residual energy, and queue occupancy are integrated into a unified weighted routing metric.
- A distance-aware weight tuning for parent selection, where the node distance to the root is used to adjust the OF weights.
- A hysteresis-based stabilization mechanism to reduce parent switch oscillations and ensuring more network stability.
- A comparative performance evaluation through simulation, demonstrating the performances of the proposed approach.

The remainder of the paper is organized as follows. Section 2 provides background on 6TiSCH and RPL. Section 3 reviews related work. Section 4 formulates the problem. Section 5 details the proposed multi-factor OF. Section 6 presents and discusses the evaluation results, and Sect. 7 concludes the paper.

2 Background

2.1 6TiSCH Architecture

6TiSCH [2] specifies a protocol stack (Fig. 1) to combine IPv6 connectivity with IEEE 802.15.4e TSCH networks. TSCH defines channel access into a cyclic *slotframe* divided into cells, each identified by a (timeslot, channel offset) pair (Fig. 1). This deterministic schedule aims to eliminate collisions and to enable energy-efficient communication through precise duty-cycling. The 6top sublayer manages cell negotiation between neighbors regarding a Scheduling Function (SF) policy, while the 6TiSCH minimal configuration provides a baseline shared schedule for bootstrapping and control traffic.



(a) The 6TiSCH network stack

(b) Example of a schedule in TSCH

Fig. 1. 6TiSCH architecture and TSCH scheduling overview

2.2 RPL Protocol

A DODAG is constructed by RPL [3], that is rooted at the border router (or LLN gateway), as shown in Fig. 2, providing upward (multipoint-to-point) and optionally downward routes. Nodes periodically exchange control packets as the DODAG Information Objects (DIOs) which is governed by the Trickle Timer algorithm, which adapts control message frequency to network stability. Each node maintains a *rank*, a scalar representing its score (for example, the distance to the root) as computed by the active OF. The preferred parent is the neighbor offering the best rank improvement.

Parent Switch Mechanism. The node occurs a parent switch when it identifies a neighbor offering a significantly better rank than its current parent. Despite this is necessary for topology adaptation, frequent switches introduce control overhead, routing instability, and transient packet losses. To limit unnecessary changes, a *hysteresis* parameter (MinHopRankIncrease) is defined by RPL, but default settings suffer in dynamic industrial environments (Fig. 3).

2.3 Default Objective Functions

OF0 [4] relies on hop count as the primary routing metric, to minimize path length to the root. This OF is designed to be metric-agnostic and acts as a fallback when no specific metrics are available. However, Of0 ignores link quality and energy considerations.

Moreover, **MRHOF** [5] aims to minimize the expected number of transmissions ETX along the path to the root. While ETX captures link reliability better

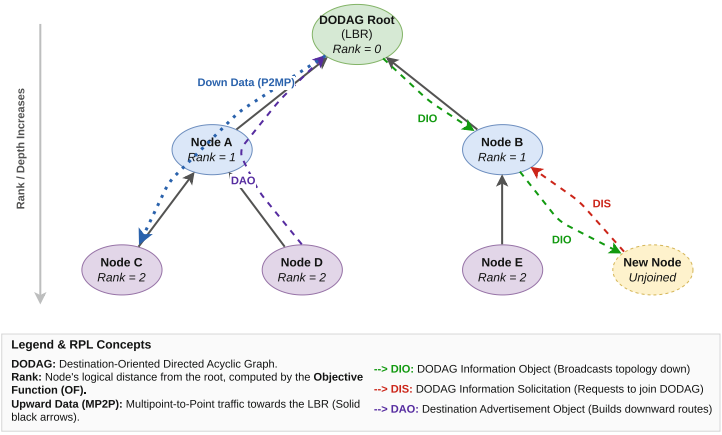


Fig. 2. DODAG Construction and Message Exchange in RPL protocol

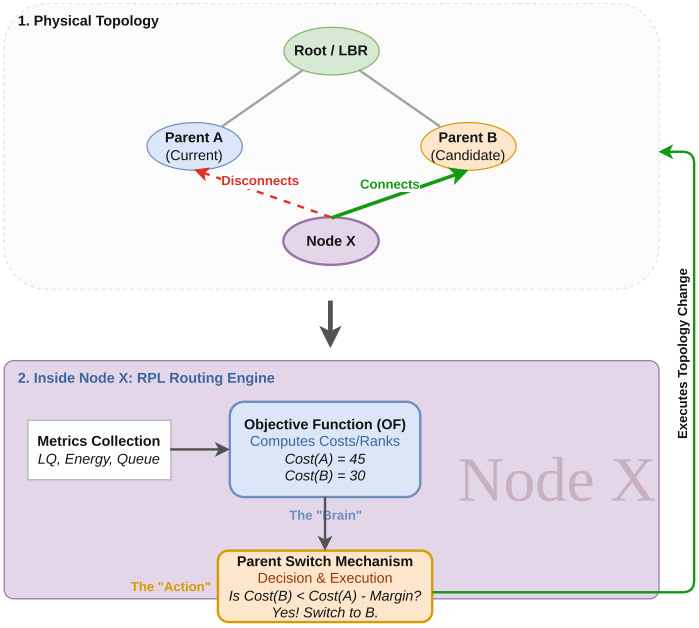


Fig. 3. Parent switch mechanism

than hop count, it remains a mono-metric approach that does not account for the parent state.

As illustrated in Fig. 4, Even if OF0 and MRHOF provide foundational routing mechanisms, their reliance on the only node parameters renders them ill-equipped to handle the multifaceted bottlenecks of dense IIoT environments. Specifically, optimizing custom made for path length or link reliability inevitably

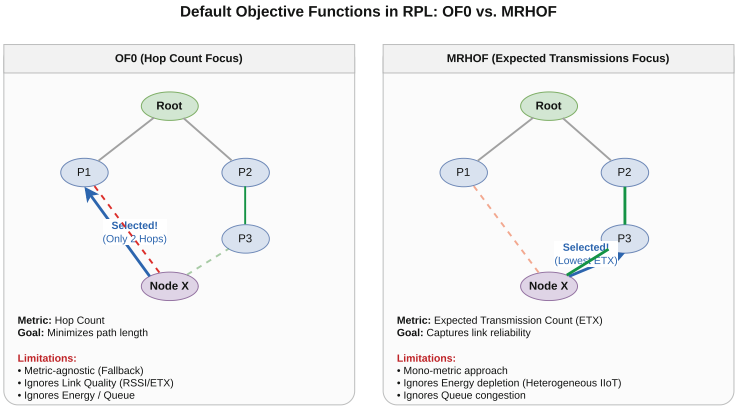


Fig. 4. Default Objective Functions in RPL: OF0 vs. MRHOF

ignores critical network conditions, leading to premature battery depletion and queue overflow, forcing nodes to frequently drop packets and trigger destabilizing parent switches.

3 Related Work

Many studies have examined RPL performance in IIoT and 6TiSCH environments. [6] evaluated RPL under various traffic loads and demonstrated that both OF0 and MRHOF suffer from instability under dense deployments. Multi-metric OF proposals have been explore in the literature: [7] proposed an energy-aware OF combining ETX and residual energy, while [8] employed fuzzy logic to combine multiple metrics into a composite routing cost.

More recently, machine learning approaches have been investigated for dynamic parent selection. [9] used reinforcement learning to adapt routing decisions to network conditions, and [10] proposed a DRL-based scheduler for 6TiSCH that jointly optimizes routing and scheduling. However, these approaches often require significant computational resources and are difficult to deploy on constrained IoT devices.

In contrast to existing work, this paper proposes a lightweight multi-factor OF that is directly compatible with standard RPL operation in 6TiSCH, without requiring modifications to the protocol stack or additional computational infrastructure.

4 Problem Formulation

To counteract the limitations observed in standard OFs, we propose, we propose **AMMOF**, a novel multi-factor objective function for RPL. AMMOF goes beyond traditional mono-metric evaluations by taking into account multi parameters like link quality (LQ), residual energy (E), and queue occupancy (Q) (see

Table 1). Furthermore, to address the heterogeneous traffic loads inherent at different DODAG depths, we introduce a distance-aware weight tuning mechanism that dynamically adjusts routing priorities based on each node's topological distance to the root. Our approach ensures a stable, energy-efficient, and resilient parent-selection process across the entire 6TiSCH network.

Table 1. Comparison of RPL Objective Functions: Default Approaches vs. Proposed AMMOF

Feature	OF0 (Default) [4]	MRHOF (Default) [5]	AMMOF (Proposed)
Primary Metric(s)	Hop Count	ETX (Expected Transmission Count)	Link Quality (LQ), Residual Energy (E), Queue (Q)
Primary Objective	Minimize path length to the root	Minimize expected transmissions	Balance reliability, energy consumption, and congestion
Link Quality Awareness	No (Metric-agnostic)	Yes	Yes (via RSSI/LQI)
Energy Awareness	No	No	Yes
Queue Awareness	No	No	Yes
Weight Tuning Policy	None (Single metric)	None (Single metric)	Dynamic (Distance-aware adaptation)
Parent Switch Stability	Low (Prone to routing loops)	Moderate (Prone to ping-pong effect)	High (Stabilized via adaptive hysteresis)
Target Environment	Basic, homogeneous deployments	Lossy networks with light traffic	Dense, heterogeneous, and heavy traffic IIoT

4.1 Network Model

We consider a 6TiSCH network composed of N nodes $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ and a set of wireless links $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The network employs a DODAG rooted at a border router v_0. A set of candidate parents $\mathcal{P}_i \subseteq \mathcal{V}$ are mainted by each node v_i, discovered through DIO messages.

4.2 Parent Switch Decision

For each node v_i, the parent switch decision is autonomously made by the Objective Function, by assigning a *rank* R_i to each node based on the rank of its preferred parent p_i^* and the cost of the link (v_i, p_i^*):

$$R_i = R_{p_i^*} + \text{rank_increase}(v_i, p_i^*) \tag{1}$$

A parent switch is triggered when a candidate parent p' satisfies this condition:

$$R_{p'} + \text{rank_increase}(v_i, p') < R_{p_i^*} + \text{rank_increase}(v_i, p_i^*) - \delta \tag{2}$$

where δ is the hysteresis threshold. The main challenge consists of defining `rank_increase` to capture multiple performance objectives simultaneously.

4.3 Performance Metrics

The performance of the parent switch mechanism is evaluated along four dimensions: **Packet Delivery Ratio (PDR)** which represent the success rate of delivered packets. **End-to-End Latency**, that average delay of transmission from source to the border router. Moreover, **Energy Consumption**, to estimate average consumption of energy per node per unit time, and finally **Topology Stability**, or the parent switch number per node during the simulation duration (Fig. 5).

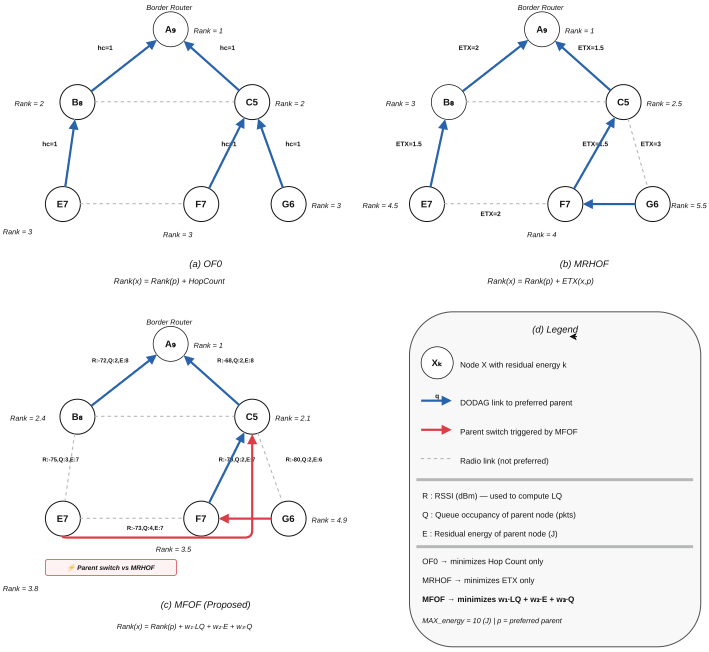


Fig. 5. Different Objective Function configurations

5 Proposed Multi-factor Objective Function

By designing the proposed OF, we took into account three principles: *comprehensiveness*, by including more metrics that reflect the diverse requirements in IIoT; *lightweight implementation*, to be deployable on constrained IoT devices; and *stability*, by integrating a lightweight hysteresis mechanism avoiding unnecessary expensive parent switches.

The complete execution flow of the proposed AMMOF is summarized in Algorithm 1. A node extracts the transmitted metrics in receiving DIO messages (Phase 1), and dynamically selects the weighting coefficients according to its topological position (Phase 2). Phase 3 consists to evaluate the composite cost for all candidate parents, and then the node applies the adaptive hysteresis threshold to make a final, stabilized routing decision (Phase 4), thereby mitigating unnecessary topology fluctuations.

5.1 Multi-factor Routing Metric

For a link (v_i, p_j), the composite link cost $C(v_i, p_j)$ is defined like a weighted combination of three normalized metrics as following:

$$C(v_i, p_j) = w_1 \cdot \hat{\text{LQ}}ij + w_2 \cdot \hat{E}j + w_3 \cdot \hat{Q}_j \tag{3}$$

where:

- $\hat{\text{LQ}}_{ij}$: normalized inverse link quality (derived from RSSI/LQI measurements).
- $\hat{E}_j$: normalized inverse residual energy of parent p_j (higher energy → lower cost).
- $\hat{Q}_j$: normalized queue occupancy of parent p_j(lower queue → lower cost).
- $w_1, w_2, w_3 \geq 0$, $\sum_{k=1}^{3} w_k = 1$: application-defined weight coefficients.

The rank increase function is then defined as follows:

$$\text{rank_increase}(v_i, p_j) = \text{MinHopRankIncrease} \times \lceil C(v_i, p_j) \times S \rceil \tag{4}$$

where S is a scaling factor that maps the composite cost to the RPL rank space.

5.2 Distance-Aware Weight Tuning for Parent Selection

In a 6TiSCH DODAG, nodes at different depths from the root are faced fundamentally to different network conditions. As nodes closer to the root as they operate under heavier traffic load due to data aggregation, making queue level and residual energy the dominant bottlenecks for parent selection. In Other hand, nodes far from the root suffer primarily from poor link quality, making RSSI the most critical metric. Thus, applying same and uniform weights w_k across all hops therefore sounds to lead to inconsistant parent selection: a fixed energy-aware policy may introduce unnecessary instability near the root, while a fixed RSSI-biased policy may drain distant nodes prematurely and reduce network lifetime.

Algorithm 1. AMMOF: Adaptive Multi-Metric Objective Function Parent Selection

Require: Set of candidate parents $\mathcal{P}$, Current parent p_{current}, Node topological distance to root d, Recent switch count $n_{\text{switch}}(t)$

Ensure: Selected preferred parent p_{best}

Phase 1: Metric Collection via DIO Messages

for each candidate parent $p \in \mathcal{P}$ **do**
  Receive DIO message from neighbor p
  Extract Link Quality LQ_p, Residual Energy E_p, and Queue Occupancy Q_p
end for

Phase 2: Distance-Aware Weight Tuning

▷ Adjust metric importance based on node depth

$(w_1, w_2, w_3) \leftarrow \text{ComputeWeights}(d)$

▷ Constraint: $w_1 + w_2 + w_3 = 1$

Phase 3: Composite Cost Calculation

for each candidate parent $p \in \mathcal{P}$ **do**
  $C(p) \leftarrow w_1 \cdot \text{LQ}_p + w_2 \cdot E_p + w_3 \cdot Q_p$
end for

$p_{\text{candidate}} \leftarrow \arg\min_{p \in \mathcal{P}} C(p)$ ▷ Find the neighbor with the lowest cost

Phase 4: Hysteresis-Based Stabilization

▷ Calculate adaptive threshold based on recent instability

$\delta(t) \leftarrow \delta_{\min} + (\delta_{\max} - \delta_{\min}) \cdot \frac{n_{\text{switch}}(t)}{n_{\text{switch,max}}}$

if $p_{\text{candidate}} \neq p_{\text{current}}$ **then**
  if $C(p_{\text{candidate}}) < C(p_{\text{current}}) - \delta(t)$ **then**
    $p_{\text{best}} \leftarrow p_{\text{candidate}}$ ▷ Trigger Parent Switch
  else
    $p_{\text{best}} \leftarrow p_{\text{current}}$ ▷ Maintain current parent to prevent oscillation
  end if
else
  $p_{\text{best}} \leftarrow p_{\text{current}}$
end if
return p_{best}

Depth-Dependent Weight Model. Let h_x represent the hop distance of node x from the root, extracted directly from its current RPL rank, and let $H_{\max}$ be the maximum network depth. The normalized depth factor could be given as (Fig. 6):

$$f(h_x) = \frac{h_x}{H_{\max}} \in [0,\, 1] \tag{5}$$

The weight of each metric $k \in \{1, 2, 3\}$ is then adapted as a linear function of $f(h_x)$:

$$w_k(h_x) = \bar{w}_k + \alpha_k \cdot f(h_x) \tag{6}$$

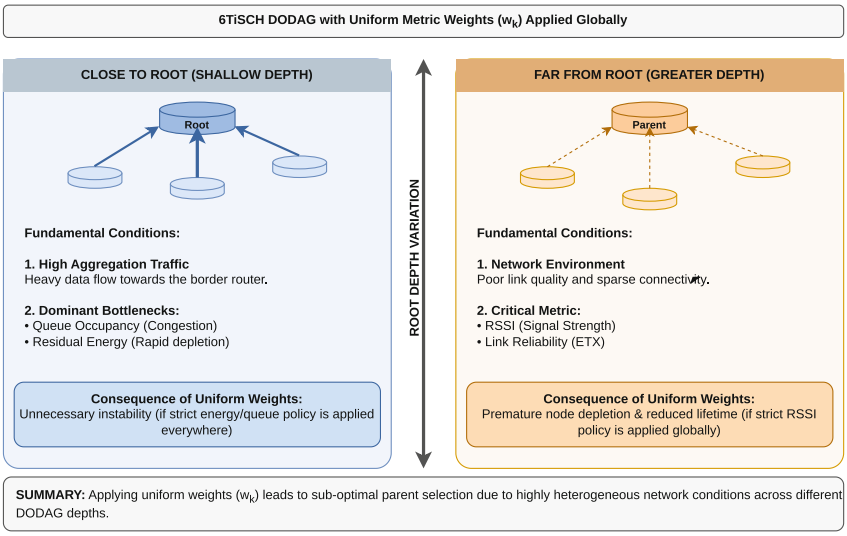


Fig. 6. Drawback of uniform weights

where $\bar{w}_k$ is the baseline weight at the root level and $\alpha_k \in \mathbb{R}$ is a signed adaptation coefficient controlling how the importance of metric k evolves with depth. The weights must satisfy the constraints:

$$\sum_{k=1}^{3} w_k(h_x) = 1, \quad w_k(h_x) \geq 0 \quad \forall\, h_x \in [0, H_{\max}] \tag{7}$$

which impose $\sum_k \bar{w}_k = 1$ and $\sum_k \alpha_k = 0$. The composite link cost is thus rewritten as:

$$C(v_i, p_j, h_{v_i}) = w_1(h_{v_i}) \cdot \widehat{\mathrm{RSSI}}_{ij}^{-1} + w_2(h_{v_i}) \cdot \hat{Q}_j + w_3(h_{v_i}) \cdot \hat{E}_j^{-1} \tag{8}$$

Adaptation Principles. The depth-dependent weight profiles are classified into three design level: **Near the root** ($f \to 0$): higher weights on queue occupancy w_2 and residual energy w_3, since these nodes relay aggregated traffic and their failure might impact the entire network and constitutes. Moreover, **Mid-depth nodes** ($f \approx 0.5$), where balanced weights across all three metrics, reflecting moderate traffic load and generally acceptable link conditions, and finally, **Leaf nodes** ($f \to 1$), where, higher weight on RSSI (w_1), since link reliability is paramount at the network periphery due to connectivity that is sparse and retransmissions are particularly energy-intensive and costly in terms latency.

Table 2 summarizes the recommended weight profiles for each depth zone.

Implementation Overhead. The hop distance h_x is directly available from the node's current rank field in DIO messages. Thanks of this approach, no

Table 2. Hop-Aware Weight Profiles per Depth Zone

Depth Zone		w_1 (RSSI)	w_2 (Queue)	w_3 (Energy)
Near root	$(h_x \leq H_{\max}/3)$	0.20	0.40	0.40
Mid-depth	$(H_{\max}/3 < h_x \leq 2H_{\max}/3)$	0.33	0.33	0.34
Leaf nodes	$(h_x > 2H_{\max}/3)$	0.60	0.20	0.20

additional control signaling is required. Weight adaptation is performed locally upon each rank update. We need only to define a Rank Max $R_{\max}$ to normalize the hop max. $R_{\max}$ is pre-configured at deployment time based on the known network topology, or estimated progressively as nodes observe the maximum rank announced in received DIO messages:

$$H_{\max} \approx \left\lfloor \frac{R_{\max}}{\texttt{MinHopRankIncrease}} \right\rfloor \tag{9}$$

where $R_{\max}$ is the highest rank value observed in the neighborhood. This estimation converges within a few Trickle Timer cycles and requires no modification to the standard RPL message format.

5.3 Hysteresis-Based Stabilization

In RPL based networks, parent switches are very expensive. To reduce parent unnecessarily switch oscillations, the hysteresis threshold δ in Eq. (2) is set adaptively based on recent switch frequency as described below:

$$\delta(t) = \delta_{\min} + (\delta_{\max} - \delta_{\min}) \cdot \frac{n_{\text{switch}}(t)}{n_{\text{switch,max}}} \tag{10}$$

where $n_{\text{switch}}(t)$ is the number of parent switches performed by the node in a recent observation window. This adaptive mechanism increases the switch threshold when frequent switches are detected, thereby penalizing oscillatory behavior and promoting topological stability.

The relevance of Eq. 10 resides in its bounded linear interpolation. Unlike a static threshold that rigidly locks a node to a degrading parent (leading to packet drops), or a zero-threshold which causes severe routing loops and control overhead, this adaptive formulation provides a proportional penalty.

6 Performance Evaluation

6.1 Simulation Setup

Simulation experiments are conducted under Cooja, the network simulator integrated with Contiki-NG [12]. We consider a 6TiSCH network with $N \in \{20, 60, 100\}$ nodes deployed in a maximum $100 \times 100\,\text{m}^2$ industrial environment. The TSCH slotframe consists of 101 timeslots, with a slot duration of 10 ms. Traffic rate used is $R = 2\,\text{pkt/s}$. Table 3 summarizes the simulation parameters.

Table 3. Simulation Parameters

Parameter	Value
Number of nodes	20, 60, 100
Area	$100 \times 100\,\mathrm{m}^2$
Slotframe size	101 slots
Slot duration	10 ms
Traffic model	Poisson
Packet size	127 bytes
Rate	2 pkt/sec
Simulation duration	3600 s
Number of runs	5
w_1, w_2, w_3	(0.2, 0.4, 0.4), (0.33, 0.33, 0.34), (0.6, 0.2, 0.2)
$\delta_{\min}$, $\delta_{\max}$	128, 512

6.2 Evaluation Metrics and Baselines

To evaluate our concepts and approach, the proposed multi-factor OF (AMMOF) is compared against:

- **OF0**: default hop-count-based objective function.
- **MRHOF**: default ETX-based objective function.

6.3 Results and Discussion

Packet Loss Ratio (Reliability). As shown in Fig. 7(a), AMMOF notably maintains the lowest Packet Loss Ratio (PLR) accross different network scenarios, reflecting a high reliability. For example, at 100 nodes, AMMOF achieves a PLR of approximately 1.8% (equivalent to a 98.2% PDR), outperforming the compared standard solutions, MRHOF (3.5%) and OF0 (4.8%). We obtained this result thanks to the consideration of link quality and queue occupancy, which helps the protocol to avoid routing data through congested or unreliable links.

End-To-End Latency. The AMMOF demontrates good performances in terms of latency. Figure 7(b) illustrates that AMMOF obtains lower average end-to-end latency with varying network topologies. For instance, at 60 nodes, the proposed AMMOF achieves a better score in terms of latency compared to standard solutions (approximately 42 ms, compared to 45 ms for MRHOF and 50 ms for OF0). At high density (100 nodes), AMMOF remains competitive when acheiving latency of roughly 66 ms. It incorporates queue occupancy into the composite metric, to make better choice in parent selection and successfully avoids forwarding packets through bottleneck nodes.

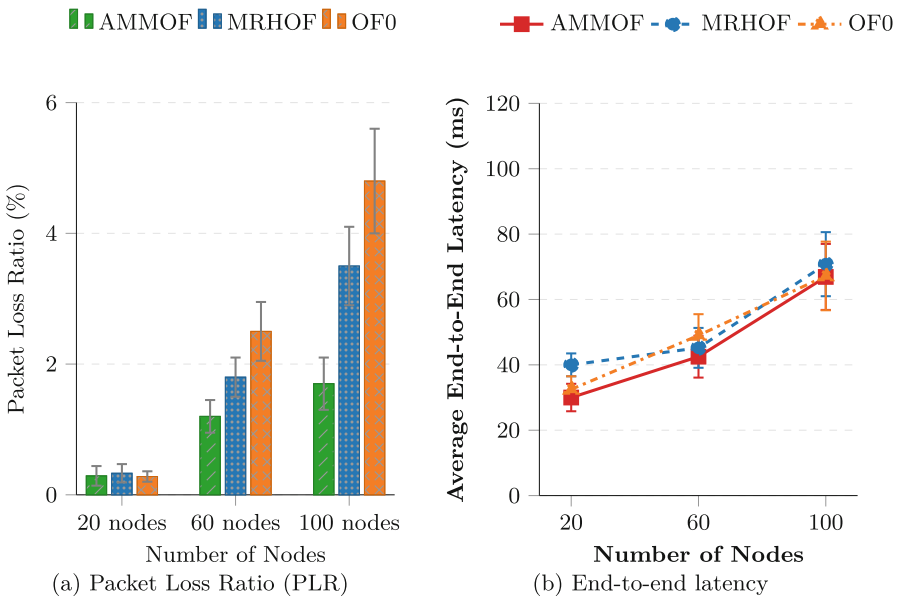
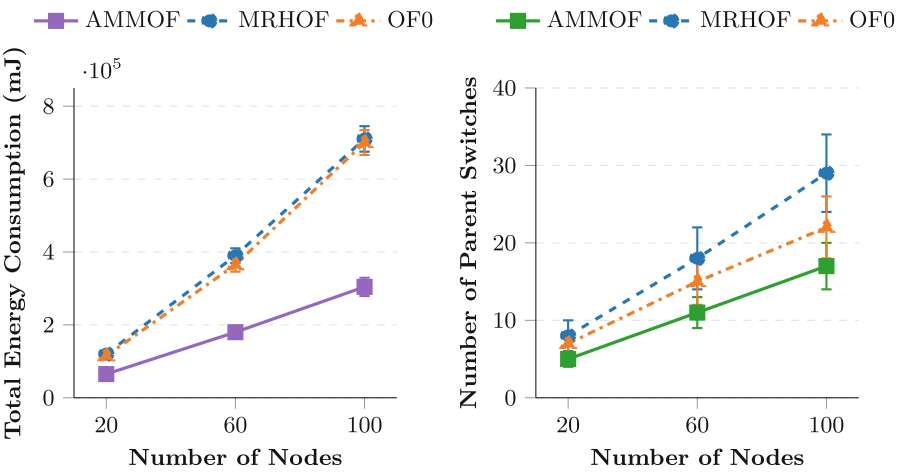


Fig. 7. Performance vs. network size (top row). Error bars represent standard deviation.

Energy Consumption. The adopted mechanisms in AMMOF of avoiding retransmissions and control overhead allow to minimize energy consumption, as shown in Fig. 8(a). At a network size of 100 nodes, AMMOF consumes roughly 3.0×10^5 mJ, whereas both MRHOF and OF0 exceed 7.1×10^5 mJ. This translates to a consistent energy savings than the default approaches. This performance is realized by prioritizing nodes holder of higher residual energy and reducing drastically control overhead. This approach allows thereby extending the overall network lifetime.

Topology Stability. The network is stable when the number of parent switches is reduced. Network stability degree is given by the number of parent switches, as depicted in Fig. 8(b). thanks to the adopted design,AMMOF consistently does not need a lot of parent switches unlike MRHOF and OF0. for instance, at 100 nodes, AMMOF incurs 5 and 12 fewer switches, compared respectively to OF0 and MRHOF. This enhanced stability is made possible with the hysteresis-based stabilization mechanism and the distance-aware weight tuning, which effectively decrease unnecessary topology fluctuation and avoid excessive ping-pong effect.

The results confirm that single-metric OFs are ill-suited for dense IIoT deployments where multiple objectives must be simultaneously satisfied. The proposed AMMOF provides a flexible and lightweight framework tuned to prioritize different application requirements.



(a) Total energy consumption (b) Parent-switch stability

Fig. 8. Performance vs. network size (bottom row). Error bars represent standard deviation.

7 Conclusion and Future Work

In this paper, we have presented a multi-factor Objective Function for RPL parent switch optimization in 6TiSCH IIoT networks. Our Approach consists of combining multiple network parameters as link quality, residual energy and queue occupancy into a unified routing metric, allowing to significantly improve network performances with the topology stability compared to the default standardized objective functions, OF0 and MRHOF.

Even if our proposed solution AMMOF could demonstrate significant enhancements in network performances such as stability, energy efficiency, and packet delivery ratio, the current formulation is based on statically defined or heuristically tuned weight coefficients (w_1, w_2, w_3). In highly dynamic 6TiSCH environments, network conditions such as traffic distribution, link interference, and node activity change continuously and unpredictably.

Determining the optimal weight configuration for all possible network states becomes fundamentally an NP-hard problem. The search space is virtually unlimited because of the continuous nature of the coefficients ($w_k \in [0, 1], \sum w_k = 1$) coupled with the combinatorial explosion of possible DODAG topologies and traffic loads. Thus, a uniform or manually configured weighting policy may be inefficient especially if network conditions are highly volatile or heterogeneous.

To address this computationally intractable problem, our future work will focus on integrating **Reinforcement Learning (RL)** into the AMMOF architecture to achieve dynamic, self-optimizing weight adaptation. We plan to for-

mulate the parent selection and metric weighting process as a Markov Decision Process (MDP).

By opting to an RL solution (such as Q-Learning for discrete weights or Deep Deterministic Policy Gradient (DDPG) for spaces of continuous action), as training progresses, nodes could learn autonomously the optimal routing policies. The RL approach consists of a cognitive and context-aware routing protocol capable of extending the lifetime of network under unpredictable IoT traffic conditions.

References

1. Wollschlaeger, M., Sauter, T., Jasperneite, J.: The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0. IEEE Ind. Electron. Mag. **11**(1), 17–27 (2017)
2. Vilajosana, X., et al.: RFC 9030: An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4 (6TiSCH). IETF (2021)
3. Winter, T., et al.: RFC 6550: RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. IETF (2012)
4. Thubert, P.: RFC 6552: Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL). IETF (2012)
5. Gnawali, O., Levis, P.: RFC 6719: The Minimum Rank with Hysteresis Objective Function (MRHOF). IETF (2012)
6. Gaddour, O., Koubaa, A.: RPL in a nutshell: a survey. Comput. Netw. **56**(14), 3163–3178 (2012)
7. Kamgueu, P.O., Nataf, E., Ndie, T.D., Festor, O.: Energy-based routing metric for RPL. In: Proc. IFIP/IEEE NOMS, pp. 1–4 (2013)
8. Lamaazi, H., Benamar, N.: A comprehensive survey on enhancements and limitations of the RPL protocol. J. Netw. Comput. Appl. **43**, 101–117 (2020)
9. Baddeley, M., et al.: DiSC: Reinforcement learning-based distributed scheduling for 6TiSCH. In: Proc. IEEE DCOSS, pp. 1–8 (2019)
10. Rodrigues, B., et al.: Deep reinforcement learning for scheduling in 6TiSCH networks. In: Proc. IEEE PIMRC, pp. 1–6 (2021)
11. Iova, O., Theoleyre, F., Noël, T.: Exploiting Multiple Parents in RPL to Improve both the Network Lifetime and its Stability. 610-616. https://doi.org/10.1109/ICC.2015.7248389
12. Oikonomou, G., et al.: Contiki-NG: The next generation of the Contiki OS (2022) . https://github.com/contiki-ng/contiki-ng
13. Aydin, B., Aydin, H., Gormus, S.: RL-OF: a reinforcement learning-inspired objective function for IETF 6TiSCH networks. In: IEEE Access **14**, 23598–23623 (2026). https://doi.org/10.1109/ACCESS.2026.3663078.
14. Ibersiene, L., Djamaa, B., Senouci , M.R., Tabouche, A.: Impact of manual parameter tuning on the performance of industrial 6TiSCH networks. In: 2025 7th International Conference on Pattern Analysis and Intelligent Systems (PAIS), Laghouat, Algeria, 2025, pp. 1–6. https://doi.org/10.1109/PAIS66004.2025.11126474(2025)
15. Zirak, M., Sedaghat, Y., Moghaddam, M.H.Y.: RPL-TSCH cross-layer design for improve quality of service in LLNs of IIoT, Ad Hoc Networks, vol. 174 (2025)

Design and Experimental Evaluation of Secure Overlay Networks for Docker Microservices Architectures

Damiano Perri(✉), Osvaldo Gervasi, and Francesco Bietolini

Department of Mathematics and Computer Science, University of Perugia, Perugia, Italy
{damiano.perri,osvaldo.gervasi}@unipg.it

Abstract. Overlay networks are today a key technology for enabling communication across distributed microservices in containerised environments. However, in standard Docker Swarm deployments, these networks offer logical isolation but lack default encryption, leaving east-west traffic vulnerable and highlighting the challenge of balancing security with efficient connectivity. This work addresses this challenge by systematically evaluating secure and insecure overlay network configurations in a controlled three-node Docker Swarm cluster. We compare four scenarios: a physical network, a standard VXLAN overlay, an IPsec-encrypted overlay, and WireGuard tunnelling for secure inter-node communication. The analysis considers latency, TCP and UDP throughput, jitter, and traffic observability via Wireshark captures. The experimental results highlight that native overlay encryption introduces a measurable overhead compared to the physical baseline, whilst WireGuard provides broader host-level protection with a different performance impact profile. In particular, significant reductions in TCP throughput and variable increases in latency are observed, depending on the solution adopted. The results highlight the trade-off between security and performance in containerised architectures and show that the WireGuard implementation yields greater benefits than other configurations, making it the most suitable choice for this project.

Keywords: Docker · Cloud Computing · Cybersecurity · Networking · Microservices

1 Introduction

Container-based microservices architectures are now the most widespread, significant and dominant [1]. The use of orchestration platforms such as Docker Swarm, in our case, allows services to be deployed across different nodes, ensuring scalability, isolation and operational flexibility; in such environments, communication between distributed microservices typically takes place via overlay

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 236–250, 2026.
https://doi.org/10.1007/978-3-032-30491-9_16

networks, which create a logical topology superimposed on the physical infrastructure. Overlay networks, which are generally based on encapsulation technologies using the VXLAN protocol, simplify the management of connectivity between distributed containers by providing isolation. However, in the standard configuration, they do not guarantee the encryption of application traffic, and this is particularly critical in the context of so-called east-west traffic—that is, internal communication between services within the same cluster—as it may carry sensitive information or application control data [2]. The evolution of Zero Trust models and the focus on the security of internal communications have highlighted how east-west traffic constitutes a significant attack surface in distributed architectures. In the absence of adequate protection mechanisms, an attacker could gain access to the physical network or a compromised node and consequently intercept, analyse or manipulate traffic between microservices, which is a critical risk in domains requiring high data integrity and confidentiality, such as secure electronic assessment systems [3]. Docker Swarm offers native IPsec-based overlay network encryption, which can be enabled using the –opt encrypted option. Alternatively, external solutions based on Virtual Private Networks (VPNs), such as WireGuard, can be used; these establish tunnels that provide end-to-end encryption between cluster nodes. Although both solutions ensure traffic confidentiality, they operate at different levels of the network stack and can affect performance in different ways. Although the extensive literature on containerised network security and VPN protocol performance is fundamental, there is a lack of systematic experimental analysis to directly compare the native encryption of Docker overlay networks with a host-level solution based on WireGuard in a real-world Docker Swarm environment. In particular, the trade-off between security and performance in terms of latency, throughput and jitter is of limited relevance in the context of east-west traffic between microservices. This work aims to explore this important concept through a comparative experimental evaluation conducted on a three-node Docker Swarm cluster, running on three different VMware virtual machines. Four scenarios related to the four configurations were analysed: the reference physical network, a standard overlay network, an IPsec-encrypted overlay network, and secure communication using WireGuard. The main contributions of this work are as follows. First, we systematically evaluate Docker overlay network performance, both with and without encryption, to determine measurable impacts. Second, we provide a direct, side-by-side comparison between Docker Swarm's native IPsec encryption and a host-level WireGuard solution, highlighting their respective performance and security implications. Third, we quantitatively analyse the effect of these security mechanisms on latency, TCP/UDP throughput, and jitter. Finally, we discuss the architectural implications of selecting specific security mechanisms, offering clear recommendations for cloud-native environments. The results provide practical guidance for system designers and administrators in selecting appropriate security strategies for distributed microservices infrastructures.

2 Related Works

The security and performance of communications in cloud-native architectures have evolved significantly in recent years, particularly with the rise of microservices and containers. The literature can be divided into four main areas: the performance of containerised and overlay networks, the security of Docker networks, performance analysis of modern VPN protocols, and the protection of east-west traffic in microservices systems. The cloud computing is today an important technology that allow users to reach remote content and services, and to use them in a flexible way, without the need to manage the underlying infrastructure [4,5]. It is also important to note that the modern Artificial Intelligence (AI) systems are increasingly being deployed in cloud environments, and a microservices architecture is often used to manage the complexity of these systems and to ensure scalability, flexibility and security [6–8].

2.1 Performance of Container and Overlay Networks

Several studies have analysed the impact of virtualisation technologies and overlay networks on network performance. Wiedner and Helm [9] evaluate container performance in low-latency virtualised environments, highlighting how virtual networking mechanisms can introduce overhead compared to direct communication over a physical network. More recent studies have specifically analysed the performance of VXLAN-based overlay networks, showing that Layer 2 encapsulation over IP infrastructure results in increased latency and reduced throughput compared to direct host-to-host communication [10,11]. Lin [12] proposes a new low-overhead overlay network architecture for containerised environments, demonstrating how optimising forwarding mechanisms can significantly reduce the performance impact. In parallel, other works have investigated the impact of the Linux kernel's virtual networking mechanisms. Abani [13] analyses the latencies introduced by network namespaces, virtual bridges and veth interfaces, showing how these components can influence the behaviour of distributed applications.

2.2 Container Network Security

The security of communications between containers has been analysed in great depth recently. Lee, for example [14], conducts an experimental analysis of attacks on Docker networks, demonstrating that traffic between containers can be intercepted and manipulated in the absence of adequate protection mechanisms. A survey on container security [15] highlights how the increase in the attack surface in cloud-native architectures is closely linked to the complexity of communications between distributed services. In the context of microservices architectures [16,17], several studies have analysed the security issues associated with communication between services. Berardi [18] presents a systematic review of security in microservices, highlighting the need to protect internal communications between application components. Similarly, Nasab [19] conducts an

empirical study on security techniques in microservice systems, seeking to understand the numerous vulnerabilities linked to the management of internal communications. This is particularly critical when microservices are integrated into broader distributed architectures, such as IoT systems monitoring public areas, where data flow securely between edge nodes and the cloud [20].

2.3 VPN Protocols and Communication Encryption

In recent years, several VPN protocols have been proposed, designed to improve performance and ease of configuration. WireGuard is now one of the most recent and promising approaches, thanks to its minimalist architecture and the use of modern cryptographic primitives [21]. Several experimental studies have compared WireGuard with traditional VPN solutions. Anyam [22] conducts a comparative performance analysis between WireGuard and OpenVPN in cloud and virtualised environments, highlighting significant differences in terms of throughput and latency. Other works analyse the comparison between WireGuard and IPsec in different network configurations. Recent studies show that the performance of the two protocols depends heavily on the deployment context and the characteristics of the underlying infrastructure [23].

2.4 Protecting East-West Traffic in Microservices

The protection of internal communications in microservices architectures is often addressed through application-level mechanisms. Chellu [24] discusses the use of mutual TLS to ensure mutual authentication and end-to-end encryption between containerised services. In parallel, several studies have analysed the performance impact of service mesh architectures used to implement such mechanisms. Barr [25] compares different service-mesh implementations, highlighting that using sidecar proxies can introduce significant latency and resource-consumption overhead. Other works investigate alternative techniques to improve the security of east-west traffic. Hussain [26] proposes an eBPF-based approach to prevent spoofing attacks in containerised clusters, whilst recent studies analyse access control and authentication models in microservices architectures [27]. A review of the literature reveals that numerous studies have separately investigated the performance of overlay networks, the security of container networks, and the performance of modern VPN protocols. However, there are few studies that directly and systematically compare the native encryption of Docker overlay networks with host-level VPN solutions such as WireGuard in the context of a Docker Swarm cluster. This work fits within this area of research by providing a comparative experimental evaluation that combines security analysis and performance measurements in the context of east-west traffic between containerised microservices.

3 System Architecture and Experimental Methodology

This section analyses the experimental architecture used in the study, the test environment, the methodology for data collection, and the results obtained

across the different network scenarios. The aim of the experiments is to assess the impact of different traffic protection solutions on east-west communication between distributed, containerised microservices. The experimental infrastructure was implemented using a Docker Swarm cluster consisting of three virtualised nodes. One node acts as the manager and is responsible for service orchestration and cluster state management, whilst the other two nodes act as workers and host the application containers. Each node runs Docker Engine and participates in the cluster network via a standard IP connection. Communication between containers distributed across different nodes is achieved via a Docker overlay network based on VXLAN, which creates an independent virtual structure superimposed on the underlying physical network. The network architecture can be viewed as a multi-layered structure, where the first layer is the underlay network – that is, the underlying IP network that connects the virtual machines hosting the cluster nodes; the overlay network is the VXLAN-based Docker virtual network that enables communication between containers distributed across different nodes; the encrypted overlay, which is a variant of the Docker overlay with native IPsec-based encryption; and finally, WireGuard, a host-level VPN tunnel that ensures fully encrypted traffic between the cluster nodes. This multi-tiered structure enables comparison of different communication security strategies across microservices. Figure 1 shows the logical architecture of the experimental system.

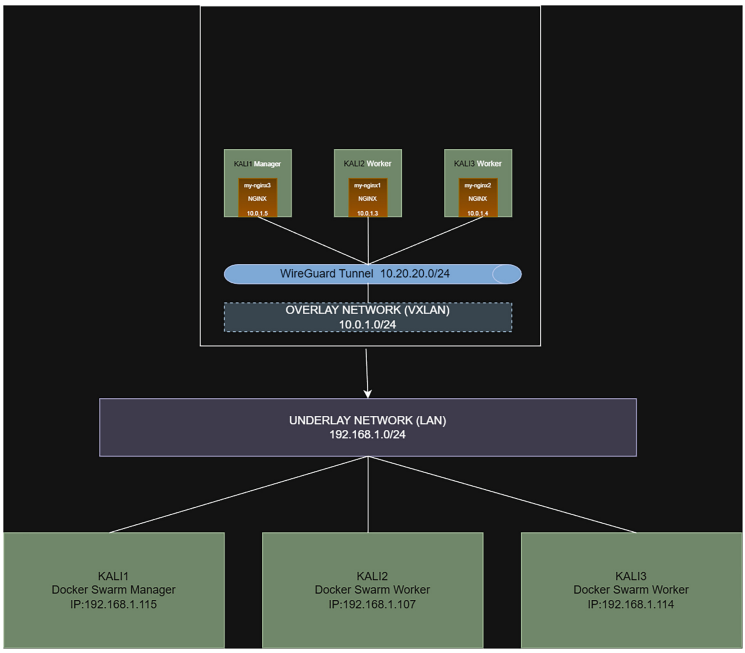


Fig. 1. System network architecture

3.1 Experimental Environment and Methodology

The experiments were conducted using three virtual machines configured as nodes in a Docker Swarm cluster. Each node runs Linux and has IP connectivity within the same virtualised local network. Application containers were deployed within the cluster and replicated across hosts to generate east-west traffic between nodes. Specifically, containers based on `nginx` images were used to simulate web services, and `alpine` containers were used to run testing and diagnostic tools. Communication between the containers was achieved via the Docker overlay network, which was dynamically created within the cluster. This approach allows containers to communicate using IP addresses from the overlay network, regardless of the physical host on which they run. The infrastructure was designed to closely resemble a realistic scenario and, above all, to be fully reproducible, utilising open-source tools and official Docker and WireGuard configurations. The aim of the experiments is to assess the impact of different encryption solutions on communication between distributed containers; for this reason, various test scenarios were defined in which the level of protection applied to communication between nodes varies. Measurement tools were used, such as tcpdump – a tool that allows packets to be captured at the interface level, useful for verifying traffic visibility – and Wireshark, one of the world's most famous and widely used tools for the graphical analysis of network activity, from flow reconstruction to payload inspection; the collected data was saved in .pcap files; Ping is a diagnostic tool based on two types of messages: the echo request and the echo reply, i.e. the request and the subsequent response to confirm connectivity; it uses the ICMP protocol and was used here to measure the Round Trip Time (RTT); curl is a tool that measures the total application response time (time_total), and finally iperf3, which measures TCP/UDP throughput, i.e. the actual data transfer rate. In addition to the tools, the study focused on analysing certain key metrics, namely RTT (i.e. average, minimum and maximum latency, with standard deviation); HTTP time (i.e. the total time taken to complete the application request); and TCP throughput (with average bitrate), and, where applicable, UDP jitter and packet loss.

3.2 Network Scenarios

The experiments were conducted by comparing four distinct network scenarios:

- Scenario A – Baseline underlay. This particular scenario represents the absolute baseline of the infrastructure, in which communication takes place directly between virtual machines without any overlay or tunnelling; it allows the performance of the physical network to be measured and provides a reference point for interpreting subsequent overheads, as this is where the overhead is lowest. In the absence of end-to-end encryption, containerised networking can expose east-west traffic to interception and analysis; it is therefore assumed that an adversary capable of capturing packets on a cluster node or a virtual network segment could compromise the system due to poor protection or misconfigurations.

- Scenario B – VXLAN Overlay. In this scenario, microservices communicate via a Docker overlay network based on VXLAN encapsulation. The overlay network enables connectivity between containers deployed on different nodes, but does not provide traffic encryption. An Alpine test container was used to enable communication between the nodes.
- Scenario C – Encrypted Overlay. The Docker overlay network is configured with native IPsec encryption. In this case, VXLAN traffic between the cluster nodes is automatically encrypted.
- Scenario D – WireGuard. In this scenario, traffic between nodes is protected by a tunnel, providing network-level encryption and integrity. Measurements were taken on hosts outside the container.
- Scenario E – Wireshark. This section analyses the observability of traffic across the three network layers used in the experimental lab: underlay, overlay and overlay with WireGuard. The aim is to demonstrate how the capture point influences traffic visibility and how different network mechanisms affect confidentiality and isolation.

Comparing these scenarios enables a systematic analysis of the trade-off between performance and security in communications between containerised microservices.

3.3 Results

We now present the results obtained from the experiments conducted across the different network scenarios, focusing on the key performance metrics of latency, throughput, and application response time, as well as traffic observability through packet captures.

Latency. The detailed results of the average latency measurements are shown in the graph below in Fig. 2 As can be seen, starting with the underlay scenario – i.e. the baseline physical network without any additional security or encryption – the average latency is optimal, at around 0.33 ms; as we move through the scenarios and add security and encryption, the latency increases. For example, in the second scenario, there is a slight increase due to VXLAN, which adds an additional header, resulting in an average latency of 0.44 ms. In the third scenario, there is a further, albeit modest, increase due to the use of IPsec, which introduces cryptographic encryption and decryption operations, with a latency of around 0.52 ms. The most significant increase in latency is seen in the WireGuard scenario, where we reach 0.78 ms, but this is still an optimal value as we remain below the 1 ms threshold; this is due to cryptographic operations but, above all, to the introduction of this VPN tunnel, which the packet must traverse, thereby lengthening the data path.

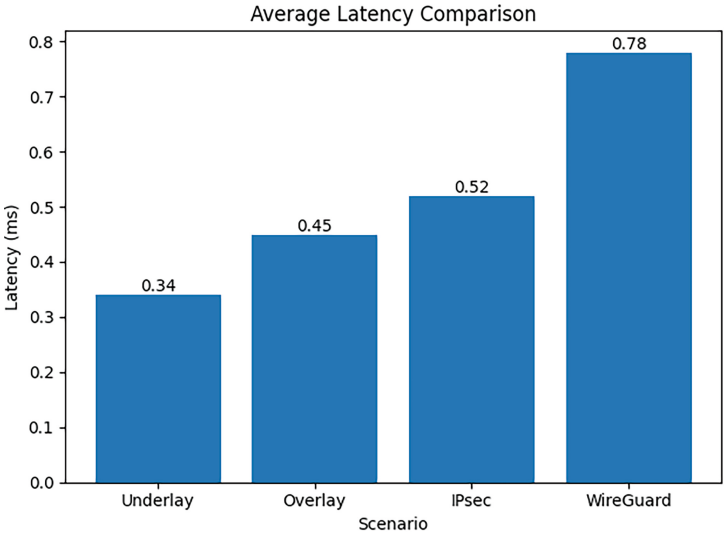


Fig. 2. Average latency results by scenario

Throughput. The detailed results of the TCP and UDP throughput measurements are shown in the graph below in Fig. 3

Starting again with the first underlay network scenario, we see a very high TCP throughput of around 3.07 Gbits/sec – which is therefore optimal – and a UDP throughput of around 222 Mbits/sec; at this point, simply introducing the overlay network with VXLAN drastically reduces the TCP value in the second scenario to 934 Mbits/sec and slightly reduces the UDP value to 198 Mbits/sec as well; this is due to the encapsulation of the protocol's additional headers; in the third scenario with IPsec, greater overhead is introduced and, consequently, both TCP throughput (639 Mbits/sec) and UDP throughput (170 Mbits/sec) are reduced. As can also be seen from the graph, WireGuard appears to be the best scenario in terms of performance, as whilst the TCP value does decrease further to 560 Mbits/sec, the UDP value increases for the first time compared to the previous scenarios to 478 Mbits/sec, confirming that WireGuard is the best solution to date in terms of the security-performance trade-off.

Curl. The detailed results of the application curl test are shown in the graph below in Fig. 4

As regards the application analysis, the response times measured using curl are in the order of a few milliseconds, indicating that the overhead introduced by the overlay is limited for simple, sequential HTTP requests. It is therefore important to clarify that this observation is valid in the specific context of the test performed, namely with synchronous, non-concurrent requests and a small

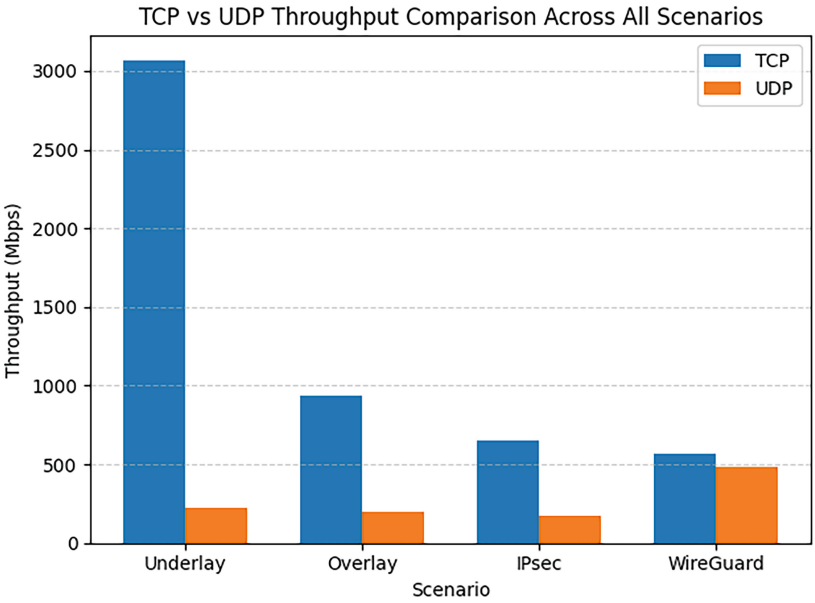


Fig. 3. TCP and UDP throughput results by scenario

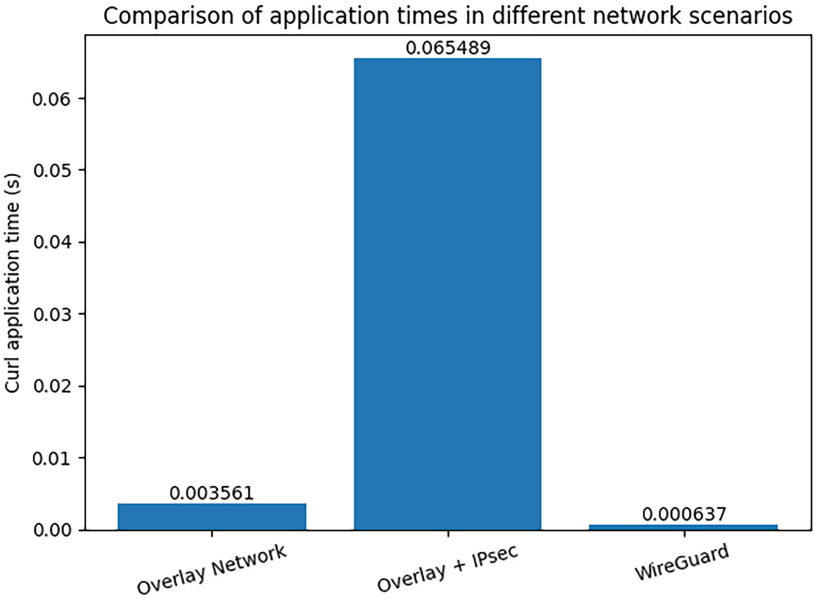


Fig. 4. Results for application time using curl

payload, which do not saturate the datapath or stress the container scheduling system. Under these conditions, the cost of VXLAN encapsulation is negligible compared to the overall application processing time. In real-world scenarios characterised by high concurrency, persistent connections, HTTP pipelines, or heavy loads, the overlay overhead could, however, become more significant. Therefore, the result obtained should be interpreted as valid for light, sequential workloads, whilst it does not claim to generalise the behaviour to intensive or highly concurrent application loads.

Wireshark. For the network analysis, we used Wireshark, which enabled us to examine the behaviour of two configurations: the first, the unencrypted overlay network shown in Fig. 5, and the second, WireGuard, shown in Fig. 6. In the first case, if we focus on the 'Protocol' section, we can see that the protocols are described and displayed in plain text, for example TCP and HTTP, and by opening the packet we can also see the corresponding TCP headers and payloads in plain text; on the right-hand side of the information, however, we can clearly see a GET request for HTTP and, a few lines below, the corresponding response, all in a transparent manner and readable by anyone; this makes it a highly vulnerable situation, prone to attacks such as sniffing. In the second case, again in the 'protocol' section, we see that only WireGuard appears in a generalised form, rather than the specific individual protocol, and on the right-hand side of the information, we no longer see the request and response; instead, all traffic is fully encrypted.

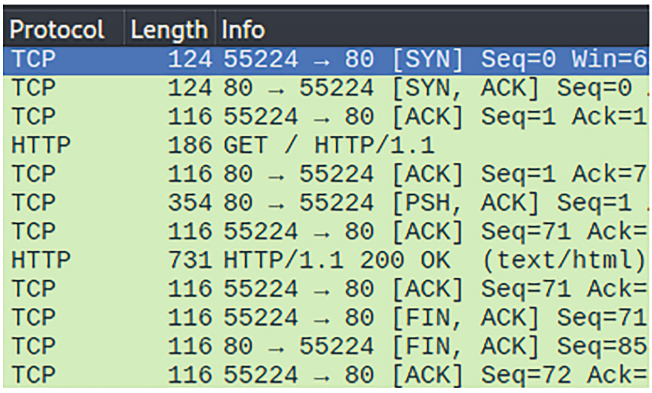


Fig. 5. Unencrypted overlay analysis

Protocol	Length	Info
WireGu…	74	Keepalive, receiver=0x3B1D2C
WireGu…	74	Keepalive, receiver=0xCA526B
WireGu…	74	Keepalive, receiver=0x1FD133
WireGu…	138	Transport Data, receiver=0xC
WireGu…	190	Handshake Initiation, sender
WireGu…	134	Handshake Response, sender=0
WireGu…	74	Keepalive, receiver=0x30C828
WireGu…	138	Transport Data, receiver=0x2
WireGu…	138	Transport Data, receiver=0x3
WireGu…	218	Transport Data, receiver=0x3
WireGu…	138	Transport Data, receiver=0xA
WireGu…	314	Transport Data, receiver=0xA

Fig. 6. WireGuard analysis

4 Discussion

The increase in latency and reduction in throughput observed in overlay and encrypted scenarios are not random phenomena, but the direct result of a combination of architectural and implementation factors that affect different network layers of the stack. The computational cost of encryption is significant. In the case of encrypted overlay networks (IPsec) and the WireGuard tunnel, each packet undergoes symmetric cryptographic operations (AES-GCM for IPsec and ChaCha-Poly1305 for WireGuard), which incur a per-packet computational cost. This cost manifests as increased processing time and a rise in the number of context switches between the kernel and user space, particularly evident in virtualised environments, where CPU resources are shared between hosts and virtual machines. The fact that TCP throughput suffers a more marked reduction than UDP is consistent with the greater number of control, retransmission, and state management operations required by reliable protocols. The reduction in effective MTU and fragmentation are two further important factors, as the use of overlays and VXLAN, as well as encrypted tunnels, introduces significant overhead due to encapsulation. Thus, approximately 50 bytes are added per packet with VXLAN, IPsec ESP adds 50–60 bytes, and WireGuard adds around 60 bytes. This effective reduction in the MTU therefore increases the likelihood of packet fragmentation or segmentation, particularly in TCP traffic, leading to more frames transmitted and placing additional load on the datapath. Does this explain why throughput does not decrease linearly but shows a measurable reduction relative to the baseline? The lengthening of the data path is also a factor; compared to the underlay, each packet traverses more components, such as the Docker virtual switch, VXLAN encapsulation, IPsec or WireGuard encryption, transmission over the physical underlay, and decapsulation and decryption upon reception. This lengthening directly contributes to the increase in end-to-end latency, but because the values remain below 1 ms, the observed overhead is primarily attributable to per-packet costs rather than congestion or packet loss.

In summary, the results show that the reduction in throughput is primarily due to computational overhead and fragmentation; the increase in latency is dominated by the lengthening of the data path and per-packet encryption; and the differences between IPsec and WireGuard can be explained by stack complexity and per-packet cost. It is important to compare the two security approaches and explain why it is insufficient to implement only the native IPsec method. The customised WireGuard approach must also be used. We can summarise this by saying that both are excellent security approaches, but we see that IPsec, in the configuration used, primarily protects the nodes that are part of the overlay network and therefore has a very limited scope and is mainly usable in this context, whilst WireGuard fully encrypts all traffic between nodes, offering greater flexibility even for different scenarios and outside the overlay context. The analysis of CPU usage, as shown in the graph in Fig. 7, based on the aggregate usage of CPU cores, highlights a different behaviour compared to, for example, an analysis of individual cores. In particular, WireGuard shows slightly higher overall CPU utilisation than IPsec. This is because WireGuard has an excellent ability to distribute the processing load across multiple processor cores, making more efficient use of multi-core architectures. Conversely, IPsec tends to concentrate a significant portion of the load on fewer cores, leading to greater saturation of individual cores but lower overall CPU utilisation across the system. Although it has this higher overhead, WireGuard remains a highly efficient solution thanks to its ability to handle parallel processing and the simplicity of its cryptographic design. When one also considers its greater flexibility in implementation and its integration with various network scenarios, a WireGuard-based configuration remains the most balanced choice in terms of the trade-off between security, performance, and scalability.

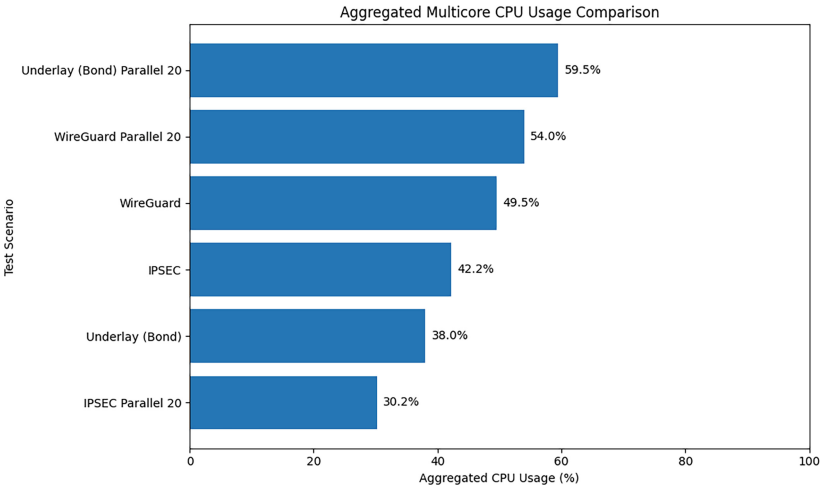


Fig. 7. CPU usage analysis

5 Conclusions

The study presented an experimental evaluation of performance and security properties across various network configurations in containerised environments based on Docker Swarm. Specifically, the study focused on analysing east-west traffic between distributed microservices, comparing four communication scenarios: a reference physical network, a VXLAN-based Docker overlay network, an IPsec-encrypted overlay, and communication secured via a WireGuard tunnel. The experimental results showed that the introduction of overlay networks and encryption mechanisms undoubtedly affects network performance. Direct communication over the physical network serves as the performance baseline, with high throughput and low latency. The use of the VXLAN overlay introduces an initial level of overhead due to encapsulation mechanisms, whilst the addition of encryption via IPsec or WireGuard tunnels entails further computational costs associated with cryptographic operations. From a security perspective, traffic analysis using packet-capture tools has shown that application traffic is easily observable in unencrypted configurations. Conversely, the use of encryption mechanisms renders the application payload unreadable in network captures, significantly reducing the attack surface associated with east-west traffic. Compared to the state of the art, this work provides a comparative experimental evaluation between the native encryption of Docker overlay networks and a host-level solution based on WireGuard in a real Docker Swarm cluster. The integrated analysis of performance and security highlights the trade-off between traffic protection and the overhead introduced by encryption mechanisms. From a practical perspective, the results suggest that the native encryption of the Docker overlay represents a simple solution that can be immediately integrated into existing infrastructures when the primary objective is to protect traffic between containers. Indeed, adopting a WireGuard tunnel between cluster nodes secures all inter-host traffic, providing a more comprehensive security model and making it particularly suitable in contexts where complete protection of internal communications is required. Overall, the study highlights that the choice of network protection mechanisms in microservice systems must be guided by a balance among security requirements, ease of management, and performance impact. Future developments could include extending the work to scenarios involving larger, geographically distributed clusters and to other widely used orchestration platforms such as Kubernetes, and incorporating metrics on CPU utilisation, energy consumption, and data-plane scalability. A further area of research concerns integration with application security mechanisms, such as service meshes, to assess the combined impact of network and end-to-end encryption in high-density microservice environments.

References

1. Perri, D., Simonetti, M., Gervasi, O.: Deploying efficiently modern applications on cloud. Electronics (Switzerland) **11**(3) (2022). https://doi.org/10.3390/electronics11030450

2. Fanfani, M., Palesi, L.A.I., Nesi, P.: Data-driven mobility demand-versus-offer matching by dynamic integration of public and sharing services. IEEE Access **14**, 23166–23193 (2026). https://doi.org/10.1109/ACCESS.2026.3661850
3. Gervasi, O., Perri, D., Simonetti, M.: Strategies and system implementations for secure electronic written exams. IEEE Access **10**, 20559–20570 (2022). https://doi.org/10.1109/ACCESS.2022.3150860. Green Open Access. All Open Access, Gold Open Access
4. Laganà, A., Gervasi, O., Tasso, S., Perri, D., Franciosa, F.: The ECTN virtual education community prosumer model for promoting and assessing chemical knowledge. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10964. LNCS (2018), pp. 533–548. https://doi.org/10.1007/978-3-319-95174-4_42. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85049988150&doi=10.1007%2f978-3-319-95174-4_42&partnerID=40&md5=23a9d4c7ef4ec100b6b06eeef01a25a6
5. Gervasi, O., Perri, D., Simonetti, M., Tasso, S.: Strategies for the digitalization of cultural heritage. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 13382. LNCS (2022). All Open Access, Hybrid Gold Open Access, pp. 486–502.https://doi.org/10.1007/978-3-031-10592-0_35. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85135788491&doi=10.1007%2f978-3-031-10592-0_35&partnerID=40&md5=51b7cb4fcebd5fbee1f71095b5b40a18
6. Fereidooni, Z., Palesi, L.A.I., Nesi, P.: Multi-agent optimizing traffic light signals using deep reinforcement learning. IEEE Access **13**, 106974–106988 (2025). https://doi.org/10.1109/ACCESS.2025.3578518
7. Dwivedi, R., et al.: Explainable AI (XAI): core ideas, techniques, and solutions. ACM Comput. Surv. **55**(9), 1–33 (2023)
8. Bilotta, S., Alessandro, L., Palesi, I., Nesi, P.: Exploiting open data for CO2 estimation via artificial intelligence and eXplainable AI. Expert Syst. Appl. **291**, 128598 (2025). issn: 0957-4174. https://doi.org/10.1016/j.eswa.2025.128598. https://www.sciencedirect.com/science/article/pii/S0957417425022171
9. Helm, M., Wiedner, F.: Performance evaluation of containers for low-latency packet processing in virtualized network environments, vol. 166. Elsevier (2024). https://doi.org/10.1016/j.peva.2024.102442
10. Gentile, A.F., et al.: Overlay and virtual private networks security performances analysis with open source infrastructure deployment. Future Internet **16**(8), 283 (2024). https://doi.org/10.3390/fi16080283
11. Ahmad, M.M., Haruna, Y., Yarima, K.I., Sani, M.A.: Analysis of docker networking and optimizing the overhead of docker overlay networks using OS kernel support. In: SCience Publishing Group (2022). https://doi.org/10.11648/j.net.20221002.11
12. Cao, P., Lin, S., Zhao, S., Han, D.: A cache-based low-overhead container overlay network. In: Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI) (2025)
13. Svensson, F., Abeni, L., Gustafsson, H., Cucinotta, T.: An investigation of linux virtual networking latencies. IEEE Trans. Network Serv. Manage. (2024)
14. Lee, J.-H., Lee, H., Kwon, S.: Experimental analysis of security attacks for docker container network in cloud computing. Electronics **12**(4), 940 (2023). https://doi.org/10.3390/electronics12040940
15. Dong, N., Jarkas, O., Ko, R., Mahmud, R.: A container security survey: exploits, attacks, and defenses. ACM Comput. Surv. (2025). https://doi.org/10.1145/3715001

16. Perri, D.: Dockerized architecture for a progressive web app: an Italian collocations dictionary. Lect. Notes Comput. Sci. **15891**, 33–43 (2026). https://doi.org/10.1007/978-3-031-97617-9_3. LNCS
17. Perri, D., Simonetti, M., Tasso, S., Ragni, F., Gervasi, O.: Implementing a scalable and elastic computing environment based on cloud containers. In: Gervasi, O., et al. ICCSA 2021. LNCS, vol. 12949. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86653-2_49
18. Mauro, J., Berardi, D., Giallorenzo, S.: Microservice security: a systematic literature review. J. Internet Serv. Appl. (2022). https://doi.org/10.7717/peerj-cs.779
19. Liang, P., Nasab, A.R., Shahin, M.: An empirical study of security practices for microservices systems. J. Syst. Softw. (2023). https://doi.org/10.1016/j.jss.2022.111563
20. Perri, D., Simonetti, M., Bordini, A., Cimarelli, S., Gervasi, O.: IoT to monitor people flow in areas of public interest. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **12958**, 658–672 (2021). https://doi.org/10.1007/978-3-030-87016-4_47. LNCS
21. Donenfeld, J.A.: WireGuard: next generation kernel network tunnel. In: Network and Distributed System Security Symposium (2017). https://api.semanticscholar.org/CorpusID:2590070
22. Larijani, H., Anyam, P., Singh, R.R.: Empirical performance analysis of WireGuard and OpenVPN in cloud and virtualised environments. Computers **14**(8), 326 (2025). https://doi.org/10.3390/computers14080326
23. Kırmızı, S., Demirdelen, T.: Performance evaluation of WireGuard and IPSec protocols in various network configurations. J. Inst. Sci. Technol. **8** (2025). https://doi.org/10.47495/okufbed.1660352
24. Chellu, R.: Secure Containerized Microservices Using PKI-Based Mutual TLS in Google Kubernetes Engine. Communications Networks and Information Security (2021)
25. Rampal, S., Barr, A.B., Tavori, J.: Performance comparison of service mesh frameworks: the mTLS test case. In: NOMS 2025-2025 IEEE Network Operations and Management Symposium (2025). https://doi.org/10.1109/NOMS57970.2025.11073712
26. Syed, H.J., Hussain, A., Aziz, A., Raza, S.: Preventing IP spoofing in kubernetes using eBPF. Computers, Mater. Continua (2025). https://doi.org/10.32604/cmc.2025.062628
27. Aldea, C.L., Bocu, R.: Authentication challenges and solutions in microservice architectures. Appl. Sci. **15**(22), 12088 (2025). https://doi.org/10.3390/app152212088

Dynamic Regression Algorithm and Multi-Objective Query Optimization in Spark SQL

Trung-Dung Le(✉)

Thuyloi University, Hanoi 11515, Vietnam
dung_lt@tlu.edu.vn
https://tlu.edu.vn/

Abstract. Cloud computing enables a pay-as-you-go model to manage heterogeneous data. This leads to complexities in Multi-Objective Query Optimization problems (MOQP) by finding a balance between conflicting objectives, such as execution time and monetary cost. Spark SQL's Catalyst optimizer uses a dynamic programming algorithm to select the best m-way join order based on Cost-Based Optimization (CBO). However, its single-objective focus is often insufficient for cloud computing, in which case the optimizer may skip good plans. Furthermore, Multi-Objective Query Optimization raises the problem of high complexity in generating the space of candidate solutions. Although the Non-dominated Sorting Genetic Algorithm based on Grid partitioning solves the high complexity problem, it needs to simplify the way to determine the Grid partitioning. In addition, estimating costs in dynamic environments is a significant challenge. To address these problems, this paper introduces a Multi-Objective Optimization extension for Spark SQL. The primary novelty of this work is the integration of a dynamic estimation model with an efficient evolutionary search for multi-objective query optimization in Spark SQL. Our contribution integrates two modules: (1) a Dynamic Regression Algorithm (DREAM) to provide accurate cost estimations with minimal historical data, and (2) Non-dominated Sorting Genetic Algorithm based on Grid partitioning and Simple front (NSGA-S) to find an approximate optimal solution with adaptive Grid partitioning. Experimental evaluations show that our approach provides superior convergence and diversity compared with other algorithms such as NSGA-II, NSGA-III, and MOEA/D. Finally, we validate the system-level feasibility through experiments on the Spark SQL framework, demonstrating high predictive precision and practical viability for cloud workloads.

Keywords: Non-dominated Sorting Genetic Algorithm · Multi-Objective Query Optimization · Linear Regression Algorithm · Spark SQL

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 251–268, 2026.
https://doi.org/10.1007/978-3-032-30491-9_17

1 Introduction

Cloud computing provides infrastructure for managing and analyzing heterogeneous data through a pay-as-you-go model [1]. This model allows users to scale the dynamic resources across multiple providers. Apache Spark and its Catalyst optimizer provide a framework for data processing, using Cost-Based Optimization (CBO) to optimize query processing [1,2], enabling the processing of data from multiple cloud providers.

However, the pay-as-you-go model also raises an important issue in terms of Multi-Objective Query Optimization (MOQO) problems, where users must balance conflicting objectives such as execution time, and monetary cost [3,4]. Moreover, dynamic workloads and wide-range communications affect the accuracy of query cost estimation [5,6]. In this context, a challenging problem is how to accurately estimate the cost of Multi-Objective Query Processing (MOQP) using historical data effectively. Furthermore, the problem of finding an approximate optimal solution of a MOQP using an efficient Multi-Objective Optimization algorithm in Spark SQL was solved in our previous study with NSGA-G [7]. However, a method for determining the grid in NSGA-G has not yet been implemented in Spark SQL.

Despite its efficiency, Spark SQL and the Catalyst optimizer primarily rely on a single-objective and dynamic programming algorithm to determine the best m-way join order [8]. This strategy significantly reduces the search space. However, in the MOQP scenario, this strategy skips good plans [9,10]. The problem of MOQP often seeks a Pareto-optimal execution plan that balances conflicting objectives. However, generating a Pareto-optimal front is difficult owing to its high complexity [11]. While NSGA-G in [6,12] solved this problem by finding an approximate optimal solution for the MOQP problem, the determination of the grid parameter n has not yet been validated in Spark SQL and Catalyst optimizer.

Furthermore, accurately estimating the costs in variable environments is a significant challenge. Cost estimation approaches can be divided into two classes: approaches without machine learning [13–15] and approaches with machine learning [16]. Among machine-learning-based approaches, traditional cost models often fail to estimate real-time cost values [17]. The Dynamic Regression Algorithm (DREAM) in [5] provided accurate estimates with low computational cost. The variable cloud environment of Spark SQL is well suited to the implementation of DREAM.

Based on our previous research, this paper introduces a Multi-Objective Query Optimization extension for Spark SQL. Our contribution integrates two components: (1) the Dynamic Regression Algorithm (DREAM), which provides accurate cost estimations with low computational cost in the variable environment [5,6], and (2) the Non-dominated Sorting Genetic Algorithm based on Grid partitioning and Simple front (NSGA-S), which uses a grid partitioning strategy to maintain population diversity. While the mathematical foundation for grid division was established in our earlier work [6], this paper provides the first validation of its automated implementation within the Spark SQL Catalyst

optimizer. Furthermore, experimental evaluations show that NSGA-S maintains population diversity, convergence, and requires less computation compared to other algorithms such as NSGA-II [18], NSGA-III [19], and MOEA/D [20].

The remainder of this paper is organized as follows. Section 2 provides a review of related work on Multi-Objective Query Optimization and cost estimation models. Section 3 shows the preliminaries and motivations for this research. A Non-dominated Sorting Genetic Algorithm based on Grid partitioning with Simple front is presented in Sect. 4. The integration of NSGA-S and DREAM as an extended library within Spark SQL is shown in Sect. 5. Finally, Sect. 6 concludes the study and discusses future research directions.

2 Related Work

Query optimization in distributed systems has been considered for both single and multiple objectives in cloud environments. This section analyzes state-of-the-art research in three domains: MOQP, cost estimation models, and evolutionary optimization algorithms.

The pay-as-you-go model raises the Multi-Objective Optimization (MOO) problem to find a good Query Execution Plan that balances conflicting objectives, such as monetary and execution time. Spark SQL and Catalyst Optimizer use Cost-Based Optimization (CBO) to optimize query processing in a single-objective problem. Cost-Based Optimization minimizes execution time based on classical access path selection algorithms [1,8]. It combines the selectivity and size of tables into a single cost metric. Song et al. (2021) [4] proposed Spark-based Cloud Data Analytics using Multi-Objective Optimization, but they did not consider the performance of the algorithm they used. Nevertheless, the solution of single-objective optimization problems cannot represent Multi-Objective Optimization problems [11]. Hence, this solution is unsuitable for the MOQP problem.

Trummer and Koch [9,10] proposed randomized and parametric algorithms for MOQP, demonstrating that the search space for join reordering expands exponentially when multiple objectives are considered. Multi-Objective Optimization often raises the problem of finding a good Query Execution Plan and the Pareto-optimal front [21]. However, generating Query Execution Plans in MOQO may be highly complex and infeasible [11]. Hence, one of the solutions in MOQO is to determine an approximately optimal solution with low complexity.

To address the high complexity and infeasibility of multi-objective spaces, Multiobjective Evolutionary Optimization (MEO) approaches often utilize Pareto dominance techniques [21]. In the class of MEO, Non-dominated Sorting Genetic Algorithms (NSGAs) [18,19] have low computational complexity with $O(MN^2)$, compared to $O(MN^3)$ computational complexity of other MEOs (where M is the number of objectives and N is the population size). Cordero et al. (2016) [22] also proposed a Dynamic multi-objective optimization with Spark, but the algorithm of NSGA-II may not be suitable for multi-objective

test problems [23]. Cheng et al. (2021) [3] solved the problem of multi-objective optimization of the configuration of Spark using a Multi-objective evolutionary algorithm based on decomposition (MOEA/D) [20]. However, NSGA-II, NSGA-III, and MOEA/D do not outperform NSGA-G [6,7,12,24]. In previous studies [6,7,12,24], NSGA-G was proposed to solve the problem in DICOM data management [24], Multi-Objective Optimization in IReS platform [6,16], cloud computing [12], and Spark SQL [7]. Although the size of the Grid partitioning of NSGA-G is considered in [6], the automated determination of the number of grid n has not yet been implemented in Spark SQL.

The accuracy of an optimizer depends on its cost estimation model. Cost estimation methods fall into two main categories: methods without machine learning [13–15] and methods with machine learning [16]. In an environment of variability, the cost functions may be quite complex. In the first class, [13] focuses on the cost model in MapReduce [25]. The query execution time was predicted using the PostgreSQL model [15]. OptEx [14] predicted job completion times in Spark using the size of the dataset, iterations, and number of nodes in cloud computing. These works focused on the execution time, which is a single objective. In the second class, many machine learning algorithms are used to estimate the execution time [26–29]. They used a complete information dataset for training and testing in the model-building process. However, the variability of the environment was not considered after the model was implemented in their systems. The cost models in [17] also fail to estimate real-time cost values. [3] proposed Machine learning-based approaches to predict execution configurations. However, they frequently demand massive training datasets and costly retraining cycles. Our previous works [5,6] were proposed to solve the problem of variability in the environment. The Dynamic Regression Algorithm (DREAM) model [5] addresses these limitations by utilizing dynamic regression to adapt to an environment of variability with minimal historical data. Spark SQL in cloud environments is well suited to the implementation of DREAM.

3 Preliminaries

This section shows the foundations and technical background required for Multi-Objective Optimization (MOO) and Dynamic Regression Algorithm extension in Spark SQL. We focus on defining the relationships among query plans, cost metrics, and the parameter space that governs the optimization process.

3.1 Pareto Plan Set

A **query** q is defined as a request for information from a database, including a set $\mathcal{Q}$ of tables. A **query execution plan** (QEP) $p \in \mathcal{P}$ is characterized by an ordered sequence of operators $o \in \mathcal{O}$ that perform data scans and joins. Any plan p can be constructed as a combination of two **sub-plans**, p_1 and p_2, through a join operator o, denoted as $p = Combine(p_1, p_2, o)$.

In a cloud environment, the cost function of the QEP depends on dynamic variables, that are unknown at the optimization time. The variables are denoted as a vector $\mathbf{x}$ within a **parameter space** $\mathcal{X}$. For the MOQO problem, the set of n cost metrics is denoted as N. QEPs are compared according to n cost metrics with respect to the parameter vector $\mathbf{x}$, defined as $\mathbf{C} = \{c^n(p, \mathbf{x}) \mid n \in N\}$. The set of cost function c is $\mathbf{C}$.

To find an optimal QEP, we would like to determine a plan that minimizes all the cost metrics across the parameter space. Given two plans $p_1, p_2 \in \mathcal{P}$, p_1 **dominates** p_2 (denoted as $p_1 \preceq p_2$) if p_1 is not worse than p_2 in any cost metric in the space of parameter $\mathcal{X}$:

$$\mathbf{C}(p_1, \mathcal{X}) \preceq \mathbf{C}(p_2, \mathcal{X}) \mid \forall n \in N, \forall \mathbf{x} \in \mathcal{X} : c^n(p_1, x) \leq c^n(p_2, x) \tag{1}$$

According to [9], the function $Dom(p_1, p_2) \subseteq \mathcal{X}$ yields the parameter space region where p_1 dominates p_2:

$$Dom(p_1, p_2) = \{x \in \mathcal{X} \mid \forall n \in N : c^n(p_1, x) \leq c^n(p_2, x)\} \tag{2}$$

If $Dom(p_1, p_2) = \mathcal{X}$, then p_1 globally dominates p_2.

p_1 **strictly dominates** p_2 if its cost is strictly lower across all metrics:

$$StriDom(p_1, p_2) = \{x \in \mathcal{X} \mid \forall n \in N : c^n(p_1, x) < c^n(p_2, x)\} \tag{3}$$

Finally, we define the **Pareto region** of a plan p as the portion of the parameter space where no alternative plan p^* exists that can strictly outperform it [21]:

$$PaReg(p) = \mathcal{X} \setminus \left(\bigcup_{p^* \in \mathcal{P}} StriDom(p^*, p) \right) \tag{4}$$

A set of plans in the Pareto regions that covers the space $\mathcal{X}$ is the optimal solution set for the MOQP problem.

3.2 Dynamic Regression Algorithm

To achieve an accurate cost prediction in a Multi-Objective Optimization problem, we use a regression-based approach. Following the statistical foundations of Multiple Linear Regression (MLR) [30], the relationship between a set of independent variables (e.g., data size and hardware configurations) and the cost metric (e.g., execution time or monetary cost) can be modeled as

$$c = \beta_0 + \sum_{j=1}^{M} \beta_j x_j + \epsilon \tag{5}$$

where β_j $(j = 0, \ldots, M)$ denotes the unknown coefficients, and x_j are the input features. The stochastic component ϵ represents the random error, typically assumed to follow a normal distribution $\mathcal{N}(0, \sigma^2)$. The **fitted equation** for estimating the cost $\hat{c}$ is expressed as

$$\hat{c} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_M x_M \tag{6}$$

The **Dynamic Regression Algorithm (DREAM)** was developed in [5, 6] by evaluating the reliability of the MLR model through the coefficient of determination, R^2. This metric signifies the proportion of the total variance in the dependent cost variable that is predictable from the independent variables, defined as:

$$R^2 = 1 - \frac{SSE}{SST} \tag{7}$$

In this equation, SSE is the sum of squared errors and SST represents the amount of total variation corresponding to the predictor variable X.

As noted in our previous works [5,6], the value of R^2 inherently increases as more historical data are integrated into the model, which may lead to overfitting or unnecessary computational complexity. In addition, in an environment of variability, updated historical data are more important than distant historical data. Consequently, DREAM incorporates a selection mechanism to identify a model that is sufficiently accurate by balancing the efficiency of determination against the number of historical records. In the context of Spark SQL, this allows the Catalyst optimizer to utilize an estimation model that is both lightweight and robust, even when the available historical query logs are sparse.

3.3 Non-dominated Sorting Genetic Algorithm

The Non-dominated Sorting Genetic Algorithm (NSGA) is a prominent class of Evolutionary Algorithms (EAs) that is widely employed to solve Multi-Objective Optimization Problems. A Multi-Objective Optimization Problem can be defined as the minimization of the objective vector function:

$$\text{minimize } F(x) = (f_1(x), f_2(x), \ldots, f_m(x)) \tag{8}$$

where $x = (x_1, \ldots, x_n)^T \in \Omega \subseteq \mathbb{R}^n$ represents an n-dimensional vector of decision variables within decision space Ω. Function F maps these variables to an m-dimensional objective space consisting of real-valued cost functions.

The NSGA procedure operates based on the principle of non-domination ranking. The process starts with the initialization of a parent population P_0 of size N. For each generation t, an offspring population O_t is generated using genetic operators, including binary tournament selection, crossover, and mutation. The combined population $C_t = P_t \cup O_t$ (size $2N$) is then partitioned into a hierarchy of non-dominated fronts $(F_1, F_2, \ldots)$. Solutions in F_1 are not dominated by any other members in C_t, whereas solutions in F_k are dominated only by those in preceding fronts $F_1, \ldots, F_{k-1}$.

The selection of the subsequent parent population P_{t+1} follows a strategy. Fronts are sequentially added to $P_{t+1}(F_1, F_2, \ldots)$ until the addition of a front F_l exceeds the population size N (i.e., $|P_{t+1}| + |F_l| > N$). In this step, a secondary

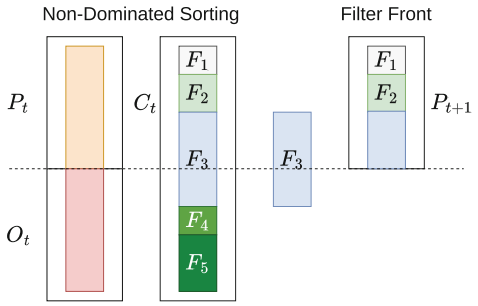


Fig. 1. NSGA-II [18], NSGA-III [19] and NSGA-G [12] procedure.

selection is required to fill the remaining solution in P_{t+1} from F_l, as illustrated in Fig. 1.

The specific approach to selection in F_l distinguishes the different versions of the algorithm. While NSGA-II implements a *crowding distance* metric to prioritize solutions in less dense regions of the solution space, maintaining a uniform distribution along the Pareto front [18], NSGA-III uses *reference points* to maintain diversity, which is particularly effective for problems with more than three objectives [19].

These methods have low computational complexity with $O(MN^2)$, compared to $O(MN^3)$ computational complexity of other MEOs (where M is the number of objectives and N is the population size). Both the crowding distance in NSGA-II and the reference points association in NSGA-III require high computation. This complexity leads to grid partitioning selection in the NSGA-S algorithm.

3.4 Motivation

Spark SQL's Catalyst [1] is highly effective at pruning search spaces through rule-based and cost-based strategies. However, they are designed for single-objective optimization, which often skips solutions that trade off between execution time and monetary costs in cloud environments. Whereas other systems, such as CloudMdsQL [31] and IReS [16], have addressed multi-engine resource scheduling, they often use scalarized cost functions or single-objective priorities. These approaches lack solutions for Multi-Objective Optimization Problems to manage the expansive search spaces of Query Execution Plans (QEPs). Although recent studies [4,22] have solved Multi-Objective Optimization Problems in Spark SQL, there remains a significant gap in providing computationally efficient methods for solving Multi-Objective Optimization Problems.

Non-dominated Sorting Genetic Algorithms (NSGA-II and NSGA-III) proposed solutions with low computational complexity, $O(MN^2)$ compared to $O(MN^3)$ of other MEOs (where M is the number of objectives and N is the population size). Nevertheless, the performance and solution quality of NSGA-II and NSGA-III did not outperform that of NSGA-G. The comparison is shown in

the benchmarks using DTLZ test problems [32]. The metrics Generational Distance (GD), Inverted Generational Distance (IGD), and Maximum Pareto Front Error [33] were used in previous works [6,7,12]. Although the size of the Grid partitioning of the NSGA-G was considered in [6], the automated determination of the number of grids n has not yet been implemented in Spark SQL.

Among machine-learning-based approaches for cost estimation, studies used the full dataset for training and testing during model building. However, an environment of variability was not considered after the model was implemented in their systems. Our previous works [5,6] were proposed to solve the problem of variability in the environment. The Dynamic Regression Algorithm (DREAM) model [5] addresses these limitations by utilizing dynamic regression to adapt to an environment of variability with minimal historical data. Spark SQL in cloud environments is well suited to the implementation of DREAM.

These limitations lead to the solution in this paper. We propose the integration of NSGA-S and the DREAM model into Spark SQL. These two components are designed to address the MOQP challenge in cloud computing by combining a dynamic estimation model and an efficient NSGA based on Grid partitioning and Simple front.

4 NSGA-S: Non-dominated Sorting Genetic Algorithm Based on Grid Partitioning and Simple Front

Multi-objective Evolutionary Optimization (MEO) has seen significant advancements with Pareto dominance-based techniques. Among these, NSGA-II and NSGA-III have lower complexity than other MEOs that use sorting-based Multi-Objective Evolutionary Optimization. While NSGA-II uses a *crowding distance* metric to prioritize solutions in less dense regions of the solution space, maintaining a uniform distribution along the Pareto front [18], NSGA-III implements a *reference points* approach to maintain diversity, which is particularly effective for problems with more than three objectives [19]. However, both solutions require high computation to remove the solutions from the last front. Although NSGA-G [6,7,12,24] was proposed and compared to NSGA-II and NSGA-III, the size of the Grid partitioning of NSGA-G was considered in [6]. Automated determination of the number of grids n has not yet been implemented in Spark SQL.

In this section, we present a Non-dominated Sorting Genetic Algorithm based on Grid partitioning and Simple front (NSGA-S) and evaluate its performance against other NSGAs with DTLZ test problems [32] with Inverted Generational Distance (IGD) and Maximum Pareto Front Error (MPFE) metrics [33].

4.1 NSGA-S

The selection of the last front of the NSGA-S is illustrated in Fig. 1. At each generation t, a parent population P_t of size N generates an offspring population O_t through binary tournament selection and genetic operators. The combined

population $C_t = P_t \cup O_t$ ($|C_t| = 2N$) is split into many fronts by non-dominated sorting to partition solutions into hierarchical fronts $(\mathcal{F}_1, \mathcal{F}_2, \dots)$ [18].

The next parent population P_{t+1} is constructed by sequentially adding the fronts from $\mathcal{F}_1$. This process continues until the front $\mathcal{F}_l$ cannot be fully added into P_{t+1}, that is, $|\mathcal{F}_l| + |P_{t+1}| \geq N$. In this step, the number of solutions in the last front that should be selected is $M = N - \sum_{i=1}^{l-1} |\mathcal{F}_i|$, members from $\mathcal{F}_l$ to complete the population.

Grid Partitioning Selection. To ensure that the selected solutions are well-distributed across the solution space without the high cost of crowding distance or reference point association, NSGA-S employs a Grid Partitioning strategy. Unlike NSGA-II, which calculates distance metrics between neighbors, or NSGA-III, which uses complex hyperplane projections, our approach groups each solution to a specific grid cell defined by Grid Min and Max points.

As illustrated in Fig. 2, for a two-objective problem with a grid unit of 0.25, a solution with objective values of $[0.44, 0.44]$ is linked to a Grid Min Point of $[0.25, 0.25]$ and a Grid Max Point of $[0.5, 0.5]$. Solutions in the same Grid Min Point and Grid Max Point are added to the same group. In each group, only the solution closest to the Grid Min Point is selected for P_{t+1}. This selection eliminates the expensive calculation of the crowding distance or the distance from all solutions to the reference point in every iteration.

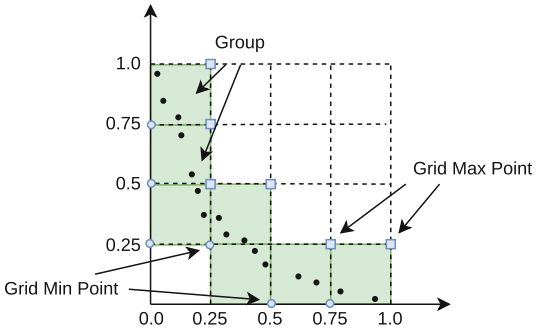


Fig. 2. Groups, Grid Min and Max Point in NSGA-S.

Simple Front. Following the selection of the size of the grid in [6], the Simple front in NSGA-S proposed a simple front group. We assume that each dimension of the solution space can be divided into n segments. In a space with N objectives, the total number of grid cells is n^N. To reduce the high comparison of each solution with the others, NSGA-S focuses on the simple front group. The number of groups in the front group consists of n^{N-1} cells. If the last front

should remove k solutions, then the number of grid n segments is

$$n = \lceil k^{\frac{1}{N-1}} \rceil \tag{9}$$

By selecting groups randomly and removing the least optimal solution within those groups, NSGA-S maintains high diversity while avoiding comparisons across all solutions in the last front $\mathcal{F}_l$.

4.2 Computational Complexity Analysis

The efficiency of the proposed NSGA-S algorithm is analyzed in two phases of each generation: non-dominated sorting and selection in the last front.

1) Non-dominated Sorting: Similar to NSGA-II [18] and NSGA-III [19], a fast non-dominated sorting approach was employed. The complexity of this phase is $O(MN^2)$, where M is the number of objectives and N is the population size. This stage is shared among most NSGA-based variants.

2) Selection: The advantage of NSGA-S is that it compares the solutions in a small group instead of all the solutions in the last front.

- **Grouping:** For each solution, adding to a specific grid cell requires normalization and coordinate calculation across M objectives, resulting in $O(NM)$ complexity.
- **Grid pruning:** Unlike NSGA-II, which requires sorting the population for each objective to calculate crowding distances ($O(MN \log N)$), or NSGA-III, which involves computing distances to H reference points ($O(N^2M)$ or $O(NHM)$), NSGA-S selects locally within the grid cells. Because the total number of individuals across all cells is N, the overall pruning process is performed in linear time, $O(N)$.

Consequently, the total selection complexity of the NSGA-S is $O(NM)$, which is a significant reduction compared to its predecessors. Table 1 summarizes the complexity of the selection phase for the compared algorithms.

Table 1. Selection Phase Complexity Comparison.

Algorithm	Diversity Mechanism	Selection Complexity
NSGA-II [18]	Crowding Distance	$O(MN \log N)$
NSGA-III [19]	Reference Points	$O(N^2M)$
NSGA-S	**Grid Partitioning**	$O(NM)$

This linear scalability in the selection phase is particularly beneficial for the Spark SQL Catalyst optimizer, where decision-making must be rapid to minimize query planning latency in high-dimensional objective spaces.

4.3 Experimental Evaluation

To ensure a fair comparative analysis, we conducted experiments comparing NSGA-S[1] with three state-of-the-art algorithms: NSGA-II, NSGA-III, and MOEA/D. All experiments were executed within MOEA Framework[2] using DTLZ test problems [32]. We evaluated these algorithms across varying objective dimensions, specifically $m \in \{4, 6, 8, 10, 12\}$.

To maintain consistency, all algorithms used the same genetic parameters, including the crossover probability, and mutation probability. Each algorithm was configured with a population size of $N = 500$ and a maximum of $M = 5000$ evaluations. We performed 50 independent runs for each problem instance to account for the stochastic nature of evolution.

The performance of the algorithms was quantified using three standard metrics: Generational Distance (GD) [33], Inverted Generational Distance (IGD) [34], and Maximum Pareto Front Error (MPFE) [35]. In all three metrics, lower values indicate better performance. Tables 2, 4, and 6 summarize the solution quality results. The data show that NSGA-S outperformed the competing algorithms in most experimental scenarios. Specifically, the Grid partitioning with Simple front selection in NSGA-S effectively maintains a better compromise between convergence and diversity as the number of objectives increases.

In addition, we analyzed computational efficiency by calculating the average execution time in seconds. As shown in Table 3, 5, and 7, NSGA-S demonstrates a significant advantage in terms of execution time. While NSGA-II, NSGA-III, and MOEA/D require high computational costs due to crowding distance, reference point association, and decomposition, respectively, the grid partitioning in NSGA-S allows for lower execution time, particularly in multi-objective experiments.

In conclusion, the experimental results demonstrate that NSGA-S provides a balance between solution quality and computational efficiency. It not only achieves high convergence and diversity but also incurs significantly lower computational cost, making it highly suitable for Multi-Objective Optimization environments of variability.

5 Integration of DREAM and NSGA-S in Spark SQL

Spark SQL provides advanced relational data processing. The Catalyst optimizer leverages both rule-based (RBO) and cost-based optimization (CBO) to enhance query performance. In the Catalyst implementation, join reordering is performed using a Dynamic Programming (DP) approach [8].

During the m-way join reordering process, the traditional CBO maintains only the single best plan for each subset of relations to minimize the search space. For instance, given a set of tables $\{A, B, C\}$, the optimizer evaluates permutations such as $(A \bowtie B) \bowtie C$, $(A \bowtie C) \bowtie B$, and $(B \bowtie C) \bowtie A$, but

[1] https://github.com/dungltr/MOEA.
[2] https://moeaframework.org/.

Table 2. Generational Distance

	m	NSGA-II	MOEA/D	NSGA-III	NSGA-S
DTLZ1	4	3.308e+01	1.727e+00	9.772e+00	1.061e+00
DTLZ3	4	4.823e+01	1.506e+01	2.729e+01	1.186e+01
DTLZ1	6	9.391e+01	1.891e+00	6.259e+00	7.091e-01
DTLZ3	6	1.252e+02	1.679e+01	2.450e+01	7.039e+00
DTLZ1	8	1.122e+02	1.642e+00	1.031e+01	9.279e-01
DTLZ3	8	1.627e+02	1.427e+01	3.023e+01	6.399e+00
DTLZ1	10	3.206e+01	2.339e-01	3.491e+00	2.469e-01
DTLZ3	10	6.113e-01	6.756e-03	1.020e-01	1.171e-03
DTLZ1	12	2.840e+01	2.579e-01	2.021e+00	2.968e-01
DTLZ3	12	5.152e-01	6.672e-03	4.432e-02	1.131e-03

Table 3. Average compute time in Generational Distance experiment (seconds)

	m	NSGA-II	MOEA/D	NSGA-III	NSGA-S
DTLZ1	4	3.512e+01	1.753e+02	1.284e+02	3.918e+01
DTLZ3	4	3.538e+01	1.802e+02	1.315e+02	4.348e+01
DTLZ1	6	4.514e+01	1.915e+02	1.359e+02	4.554e+01
DTLZ3	6	4.788e+01	1.934e+02	1.421e+02	4.862e+01
DTLZ1	8	5.370e+01	2.253e+02	1.996e+02	4.976e+01
DTLZ3	8	5.612e+01	2.235e+02	2.042e+02	5.292e+01
DTLZ1	10	6.024e+01	2.424e+02	3.396e+02	5.258e+01
DTLZ3	10	6.422e+01	2.431e+02	3.660e+02	5.784e+01
DTLZ1	12	6.710e+01	2.730e+02	1.657e+02	5.702e+01
DTLZ3	12	7.208e+01	2.765e+02	1.783e+02	6.250e+01

Table 4. Inverted Generational Distance

	m	NSGA-II	MOEA/D	NSGA-III	NSGA-S
DTLZ1	4	4.268e+01	2.448e+00	1.077e+01	4.038e+00
DTLZ3	4	1.134e+02	4.123e+01	8.400e+01	4.806e+01
DTLZ1	6	2.796e+02	3.001e+00	9.513e+00	3.458e+00
DTLZ3	6	5.527e+02	3.977e+01	8.305e+01	3.570e+01
DTLZ1	8	3.947e+02	2.354e+00	1.337e+01	3.831e+00
DTLZ3	8	6.735e+02	4.178e+01	1.046e+02	3.100e+01
DTLZ1	10	1.229e+01	3.464e-01	1.135e+00	8.677e-01
DTLZ3	10	2.568e+00	1.741e-01	4.336e-01	1.959e-01
DTLZ1	12	8.299e+00	4.644e-01	3.848e-01	8.174e-01
DTLZ3	12	2.897e+00	2.214e-01	2.214e-01	2.365e-01

Table 5. Average compute time in Inverted Generational Distance experiment (seconds)

	m	NSGA-II	MOEA/D	NSGA-III	NSGA-S
DTLZ1	4	3.244e+01	1.746e+02	1.240e+02	3.704e+01
DTLZ3	4	3.530e+01	1.736e+02	1.302e+02	4.074e+01
DTLZ1	6	4.430e+01	1.875e+02	1.337e+02	4.460e+01
DTLZ3	6	4.696e+01	1.909e+02	1.395e+02	4.796e+01
DTLZ1	8	5.312e+01	2.212e+02	1.947e+02	4.874e+01
DTLZ3	8	5.604e+01	2.240e+02	2.061e+02	5.298e+01
DTLZ1	10	6.016e+01	2.422e+02	3.455e+02	5.292e+01
DTLZ3	10	6.442e+01	2.434e+02	3.643e+02	5.814e+01
DTLZ1	12	6.704e+01	2.722e+02	1.652e+02	5.690e+01
DTLZ3	12	7.164e+01	2.772e+02	1.773e+02	6.258e+01

Table 6. Maximum Pareto Front Error

	m	NSGA-II	MOEA/D	NSGA-III	NSGA-S
DTLZ1	4	8.439e+02	2.764e+01	5.250e+02	2.309e+01
DTLZ3	4	1.064e+03	1.276e+02	8.852e+02	1.830e+02
DTLZ1	6	1.560e+03	6.986e+01	3.675e+02	1.489e+01
DTLZ3	6	1.838e+03	1.947e+02	7.156e+02	1.099e+02
DTLZ1	8	1.993e+03	4.372e+01	5.577e+02	1.651e+01
DTLZ3	8	2.167e+03	2.058e+02	8.883e+02	8.685e+01
DTLZ1	10	8.640e+02	7.067e+00	2.238e+02	3.935e+00
DTLZ3	10	1.517e+01	1.979e-01	6.927e+00	5.841e-02
DTLZ1	12	9.586e+02	6.155e+00	1.142e+02	4.428e+00
DTLZ3	12	1.137e+01	1.501e-01	1.582e+00	3.541e-02

Table 7. Average compute time in Maximum Pareto Front Error experiment (seconds)

	m	NSGA-II	MOEA/D	NSGA-III	NSGA-S
DTLZ1	4	3.224e+01	1.739e+02	1.237e+02	3.718e+01
DTLZ3	4	3.530e+01	1.742e+02	1.301e+02	4.086e+01
DTLZ1	6	4.414e+01	1.884e+02	1.332e+02	4.432e+01
DTLZ3	6	4.686e+01	1.890e+02	1.393e+02	4.810e+01
DTLZ1	8	5.318e+01	2.219e+02	1.944e+02	4.894e+01
DTLZ3	8	5.588e+01	2.234e+02	2.032e+02	5.270e+01
DTLZ1	10	6.062e+01	2.408e+02	3.384e+02	5.248e+01
DTLZ3	10	6.460e+01	2.460e+02	3.664e+02	5.794e+01
DTLZ1	12	6.676e+01	2.722e+02	1.656e+02	5.692e+01
DTLZ3	12	7.184e+01	2.773e+02	1.778e+02	6.246e+01

ultimately retains only the one with the lowest estimated cost. While efficient for single-objective optimization, this pruning strategy is inherently limited for Multi-Objective optimization, as it potentially skips plans that balance conflicting objectives. Currently, the Catalyst estimates the cost of a join operator using a scalarized formula:

$$\text{Cost} = w \cdot \text{cardinality} + (1.0 - w) \cdot \text{size} \tag{10}$$

where w is a predefined weight. Our strategy extends this framework by integrating the **DREAM** and **NSGA-S** algorithms to support Multi-Objective Query Processing across cloud environments.

5.1 Multi-objective Query Processing

The sequence of the proposed query processing method is illustrated in Fig. 3. The procedure begins when a user submits a query along with its specific preference model (e.g., a Weighted Sum Model). We integrated our extension into Spark SQL to generate a set of candidate logical plans. Unlike the standard optimizer, this step maintains multiple objectives throughout the cost-based optimization phase.

Subsequently, a Pareto-optimal set of logical plans is generated, and the most suitable plan is selected based on the Weighted Sum model. The selected plan is then passed back to the standard Spark SQL engine to generate physical planning and final execution.

The detailed architecture for generating multi-objective logical plans is shown in Fig. 4. This integration was implemented within the logical optimization stage of Catalyst.

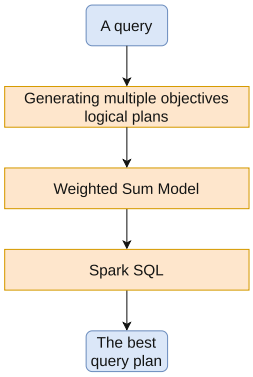


Fig. 3. Steps of query processing.

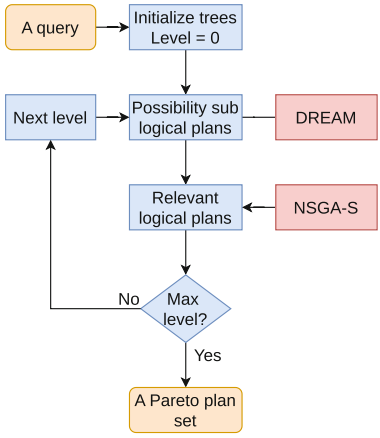


Fig. 4. Steps of generating logical plans.

5.2 Experimental Implementation and Evaluation

The Multi-Objective Optimization extension was implemented in Java, extending the core Spark SQL and Catalyst libraries. In the standard Catalyst optimizer, join reordering is handled using a dynamic programming (DP) algorithm [1,8] that identifies the optimal m-way join order. However, this baseline approach remains the single best plan for any given set of m relations based on the scalarized cost function defined in Eq. 10. Although this effectively prunes the search space, it may skip good candidate plans that balance the conflicting objectives.

To address this, our experimental setup modifies the join reordering phase to retain a set of intermediate logical subplans. We optimized these plans with multi-metric cost estimates generated by the **DREAM** model. Specifically, DREAM uses historical data statistics to estimate regression coefficients $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_L$. These coefficients are then applied to the current logical plan features to predict the estimated cost $\hat{c}$.

The efficiency of our approach was evaluated using the TPC-DS benchmark with a 1GB dataset, focusing on complex queries involving 5-table joins. The experimental results demonstrated a high degree of estimation accuracy. The average observed execution time was **39.298 s**, which closely matches the average estimated time of **39.177 s**. In particular, the number of times for error = abs (estimated running time - actual running time) $< 20\%$ reached **263 out of 285** instances (approximately 92.3%). This confirms that the integration of DREAM and NSGA-S not only maintains a balanced set of plans but also achieves a high level of predictive precision for Spark SQL.

The superior performance of NSGA-S in terms of convergence and diversity can be explained by its Grid partitioning, which avoids the computation of the crowding distance in NSGA-II or reference point association in NSGA-III. This allows the Spark Catalyst optimizer to explore the Pareto front within a smaller time budget.

From a system-level perspective, the integration of NSGA-S introduces minimal overhead to the Catalyst optimization phase. While traditional NSGA variants suffer from quadratic complexity $O(MN^2)$ in population selection, the linear scalability of NSGA-S $O(NM)$ ensures that the Pareto-optimal search does not become a bottleneck during query planning. Furthermore, the 92.3% estimation accuracy achieved on TPC-DS benchmarks suggests that the dynamic adaptation of DREAM effectively predicts the cost of logical plans, making Pareto-based query planning practically viable for cloud workloads.

6 Conclusion and Future Work

This paper presented a Multi-Objective Query Optimization extension for Spark SQL. In an environment of variability, the DREAM model is implemented for the estimation of cost functions using the **NSGA-S** algorithm for efficient Pareto-optimal search space pruning.

Experiments on DTLZ test problems, Generational Distance (GD), Inverted Generational Distance (IGD), and Maximum Pareto Front Error (MPFE) metrics show that **NSGA-S** significantly outperforms NSGA-II, NSGA-III, and MOEA/D, particularly in terms of convergence, diversity, and execution times in Multi-Objective Optimization problems. Furthermore, the integration of NSGA-S and DREAM into the Spark SQL Catalyst optimizer enables the processing of queries across cloud environments. Our preliminary experimental results highlight the capability of the framework to maintain a high predictive accuracy (over 92%) in Spark SQL.

Future research will focus on validating this proposal through extensive testing with commercial cloud providers and large-scale data management systems.

References

1. Armbrust, M.: Spark SQL: Relational Data Processing in Spark. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (SIGMOD), pp. 1383–1394. ACM (2015). https://doi.org/10.1145/2723372.2742797
2. Databricks. Cost-based optimizer in Apache Spark SQL (2024). https://docs.databricks.com/spark/latest/spark-sql/cbo.html
3. Cheng, G., Ying, S., Wang, B.: Tuning configuration of apache spark on public clouds by combining multi-objective optimization and performance prediction model. In: J. Syst. Softw. 180.C (2021). https://doi.org/10.1016/j.jss.2021.111028
4. Song, F.: Spark-based cloud data analytics using multi-objective optimization. In: IEEE 37th International Conference on Data Engineering (ICDE), pp. 396–407. , Chania, Greece (2021). https://doi.org/10.1109/ICDE51399.2021.00041
5. Le, T.-D., Kantere, V., d'Orazio, L.: Dynamic estimation for medical data management in a cloud federation. In: Proceedings of the Workshops of the EDBT/ICDT 2019 Joint Conference (DARLI-AP), vol. 2322. CEUR Workshop Proceedings
6. Le, T.-D., Kantere, V., d'Orazio, L.: Dynamic estimation and grid partitioning approach for multi-objective optimization problems in medical cloud federations. In: Transactions on Large-Scale Data- and Knowledge-Centered Systems XLVI. Lecture Notes in Computer Science 12410 (2020), pp. 27–57. https://doi.org/10.1007/978-3-662-62386-2_2
7. Le, T.-D.: Dynamic multiple linear regression and multiple objectives query optimization in spark SQL. In: Computational Science and Its Applications – ICCSA 2025, vol. 15648. Lecture Notes in Computer Science. Cham: Springer (2025)
8. Selinger, P.G.: Access path selection in a relational database management system. In: Proceedings of the 1979 ACM SIGMOD International Conference on Management of Data (SIGMOD'79), pp. 23–34. ACM (1979). https://doi.org/10.1145/582095.582099
9. Trummer, I., Koch, C.: Multi-objective parametric query optimization. Proc. VLDB Endowment **8**(3), 221–232 (2014). https://doi.org/10.14778/2735508.2735512
10. Trummer, I., Koch, C.: A fast randomized algorithm for multi-objective query optimization. In: Proceedings of the 2016 International Conference on Management of Data (SIGMOD'16), pp. 1737–1752. ACM (2016). https://doi.org/10.1145/2882903.2915206

11. Zitzler, E., et al.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans. Evol. Comput. **7**(2), 117–132 (2003). https://doi.org/10.1109/TEVC.2003.810758
12. Le, T.-D., Kantere, V., d'Orazio, D.: An efficient multi-objective genetic algorithm for cloud computing: NSGA-G. In: IEEE International Workshop on Benchmarking, Performance Tuning and Optimization for Big Data Applications (BPOD). Seattle, WA, USA, pp. 3001–3010 (2018). https://doi.org/10.1109/BigData.2018.8622151
13. Nykiel, T., et al.: MRShare: sharing across multiple queries in MapReduce. Proc. VLDB Endow. **3**(1–2), pp. 494–505 (2010). issn: 2150-8097. https://doi.org/10.14778/1920841.1920906
14. Sidhanta, S., Golab, W., Mukhopadhyay, S.: OptEx: a deadline-aware cost optimization model for spark. In: 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pp. 193–202. (2026). https://doi.org/10.1109/CCGrid.2016.10
15. Wu, W.: Predicting query execution time: are optimizer cost models really unusable? In: IEEE 29th International Conference on Data Engineering (ICDE), pp. 1081–1092. (2013). https://doi.org/10.1109/ICDE.2013.6544899
16. Doka, K.: IReS: intelligent, Multi-Engine Resource Scheduler for Big Data Analytics Workflows. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (SIGMOD '15), pp. 1451–1456. ACM (2015). https://doi.org/10.1145/2723372.2735377
17. Misegiannis, M.G., Kantere, V., d'Orazio, L.: Multi-objective query optimization in Spark SQL. In: Proceedings of the 26th International Database Engineering & Applications Symposium (IDEAS). 2022, pp. 70–74. https://doi.org/10.1145/3548785.3548800
18. Deb, K.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002). https://doi.org/10.1109/4235.996017
19. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach. IEEE Trans. Evol. Comput. **18**(4), 577–601 (2014). https://doi.org/10.1109/TEVC.2013.2281535
20. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007). https://doi.org/10.1109/TEVC.2007.892759
21. Hulgeri, A., Sudarshan, S.: Parametric Query Optimization for Linear and Piecewise Linear Cost Functions. In: Proceedings of the 28th International Conference on Very Large Data Bases (VLDB'02). VLDB Endowment, pp. 167–178. (2002). https://doi.org/10.5555/1287369.1287385
22. Cordero, J., et al.: Dynamic multi-objective optimization with jMetal and spark: a case study. In: International Workshop on Machine Learning, Optimization and Big Data (MOD 2016). Vol. 10122. Lecture Notes in Computer Science. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-51469-7_9
23. Ishibuchi, H.: Performance comparison of NSGA-II and NSGA-III on various many-objective test problems. In: IEEE Congress on Evolutionary Computation (CEC), pp. 3045–3052. IEEE (2016). https://doi.org/10.1109/CEC.2016.7744174
24. Le, T.-D., Kantere, V.: Optimizing DICOM data management with NSGA-G. In: Proceedings of the Workshops of the EDBT/ICDT 2019 Joint Conference, vol. 2324. () CEUR Workshop Proceedings
25. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Commun. ACM **51**(1), 107–113 (2008). issn: 0001-0782. https://doi.org/10.1145/1327452.1327492

26. Akdere, M.: Learning-based query performance modeling and prediction. In: IEEE 28th International Conference on Data Engineering, pp. 390–401. (2012). https://doi.org/10.1109/ICDE.2012.64
27. Ganapathi, A.: Predicting multiple metrics for queries: better decisions enabled by machine learning. In: IEEE 25th International Conference on Data Engineering, pp. 592–603. (2009). https://doi.org/10.1109/ICDE.2009.130
28. Tozer, S., Brecht, T., Aboulnaga, A.: Q-Cop: avoiding bad query mixes to minimize client timeouts under heavy loads. In: IEEE 26th International Conference on Data Engineering (ICDE 2010), pp. 397–408. (2010). https://doi.org/10.1109/ICDE.2010.5447850
29. Xiong, P., et al.: ActiveSLA: a profit-oriented admission control framework for database-as-a-service providers. In: Proceedings of the 2nd ACM Symposium on Cloud Computing. SOCC '11. Cascais, Portugal: Association for Computing Machinery, 2011. isbn: 9781450309769. https://doi.org/10.1145/2038916.2038931
30. Soong, T.T.: Fundamentals of Probability and Statistics for Engineers, Wiley (2004)
31. Kolev, B.: The CloudMdsQL Multistore System. In: Proceedings of the 2016 International Conference on Management of Data (SIGMOD'16), pp. 2113–2116. ACM (2016). https://doi.org/10.1145/2882903.2899400
32. Deb, K.: Scalable multi-objective optimization test problems. In: Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02), vol. 1, pp. 825–830. IEEE (2002). https://doi.org/10.1109/CEC.2002.1007032
33. Veldhuizen, D.A.V.: Multiobjective evolutionary algorithms: classifications, analyses, and new innovations, Air Force Institute of Technology (1999). Ph.D. thesis
34. Coello, C.A.C., Cortés, N.C.: Solving Multiobjective Optimization Problems Using an Artificial Immune System. Genet. Program Evolvable Mach. **6**, 163–190 (2005). https://doi.org/10.1007/s10710-005-6164-x
35. Veldhuizen, D.A.V., Lamont, G.B.: Evolutionary Computation and Convergence to a Pareto Front. In: Late Breaking Papers at the Genetic Programming 1998 Conference. 1998, pp. 221–228

Analyzing Optimization Trade-Off Between Energy and Model Convergence in Federated Learning

Nilo Marchiori Louzada(✉), Renato Elias Nunes de Moraes, Rodolfo S. Villaça, Vinícius F. S. Mota, and Maria Claudia Silva Boeres

Universidade Federal do Espírito Santo, Vitória, ES, Brazil
nilo.louzada@edu.ufes.br, {renato.moraes,rodolfo.villaca,vinicius.f.mota, maria.boeres}@ufes.br

Abstract. Federated Learning (FL) has established itself as a promising solution for training artificial intelligence models on edge devices while preserving data privacy. However, the hardware heterogeneity and energy constraints of IoT devices pose severe challenges to the sustainability of this paradigm. This work addresses energy efficiency optimization in FL networks by formulating the problem through the joint control of training time and device CPU frequency. Unlike purely theoretical approaches, the proposed methodology employs Lagrangian Relaxation and Karush-Kuhn-Tucker (KKT) conditions to determine the optimal frequency, validating results through realistic network emulation using a container-based tool. Experiments conducted with the MNIST dataset in mid-iid and non-iid data distribution scenarios demonstrate that strict time limit control, although reducing consumption per round, significantly impairs the global model's convergence. In contrast, the proposed optimization via frequency adjustment proved capable of reducing total energy consumption without degrading final accuracy. Additionally, combining frequency optimization with client selection strategies yields further reductions in accumulated energy consumption.

Keywords: Federated Learning · Lagrangian Relaxation · KKT · Convex Optimization

1 Introduction

The accelerated proliferation of smart edge devices, driven by the Internet of Things (IoT), has generated an unprecedented volume of data that challenges centralized processing architectures [8]. Although training Artificial Intelligence (AI) models on these data is highly valuable, transferring it to the cloud imposes critical latency, bandwidth, and privacy barriers [14]. Federated Learning (FL) has emerged as the solution to this scenario, enabling the collaborative training of global models while keeping the data residing on local devices [15,20].

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 269–286, 2026.
https://doi.org/10.1007/978-3-032-30491-9_18

However, the practical implementation of FL in wireless networks faces the inherent heterogeneity of the edge environment. Devices vary in processing capacity and, crucially, in energy constraints [19,21]. Energy efficiency is therefore a primary requirement for the sustainability of FL. Although reducing CPU frequency is a common strategy to save energy, it directly impacts computing time and, consequently, the total latency of training rounds.

The optimization problem can be defined as, given a federated learning network, determining the optimal values for CPU frequency and training operation time, maximizing accuracy while minimizing energy consumption, considering the heterogeneity of the devices and their energy constraints. In this context, the objective of this work is to analyze the real impacts on the global model convergence when applying energy optimization based on CPU frequency control.

The literature has established rigorous theoretical foundations for this challenge. The authors in [19] formulated the trade-off between learning time and energy consumption through the FEDL (Federated Learning over wireless networks problem), whose solution requires the joint control of computation (via CPU frequency) and communication latencies. Extending these concepts, [11] introduced the DCM (Dataset and Computation Management) framework, which considers the dataset fraction as a control variable.

Additional complexity is given by the network topology and the nature of the data. Indeed, [14] and [20] demonstrated that edge hierarchy and adaptive participant selection are vital for minimizing costs. In network scenarios formed by low-energy devices, [2] highlighted the need for routing topology awareness to avoid the depletion of critical nodes. Moreover, the non-iid distribution of data imposes extra challenges to convergence, requiring adaptive learning rate strategies [8].

Despite theoretical advancements, the literature still presents a methodological gap in the use of CPU frequency as a decision variable in optimization models, which frequently rely on purely numerical solutions [11,14,19,21]. This approach provides limited insights into network behavior in a realistic emulation environment. To narrow the gap between theoretical optimization and practical assessment, this work provides the following contributions:

- An experimental characterization of the convergence impact of CPU frequency based energy optimization in FL, validated through realistic network emulation;
- An empirical demonstration that frequency control reduces total energy consumption by up to 30% without accuracy degradation under both mid-iid and non-iid data distributions.

The remainder of this article is organized as follows. Section 3 introduces the mathematical formalism for abstracting federated learning networks. Next, Sect. 4 discusses the optimization problem associated with processing time and energy consumption of client devices. The employed network emulation methodology, as well as the presentation and analysis of the obtained results, are

described in Sect. 5. Finally, Sect. 6 concludes this paper and provides suggestions for future work.

2 Related Work

Resource efficiency and client selection in Federated Learning (FL) have attracted significant attention from the scientific community, which can be categorized into strategies focused on resource allocation, multi-criteria selection, and system-level validations.

Resource Allocation and Energy Efficiency: Several studies address the optimization of system parameters to enable FL. Tran *et al.* [19] modeled FL over wireless networks as a convex optimization problem, proposing an algorithm that minimizes both training time and device energy consumption. Similarly, Zhou *et al.* [21] investigated multi-objective optimization in bandwidth-limited systems. Despite their strong mathematical foundations, such works often validate their approaches primarily through numerical simulations.

Client Selection in Heterogeneous Networks: Mitigating the effect of *stragglers* is commonly addressed through intelligent client selection. Nishio and Yonetani [16] introduced the FedCS protocol, which proactively selects clients based on their resource capacities to maximize the number of participants within a specific time window. More recent approaches, such as those by Tahir and Ali [18] and Ali *et al.* [1], expand this idea by applying heuristic and multi-criteria selection (*e.g.*, energy, data quality, and bandwidth).

The Gap in Realistic Emulation: Recent review papers, such as those by Badra *et al.* [10] and Daly *et al.* [6], emphasize that one of the greatest challenges in modern FL is the transition from mathematical theory to production environments. Emulation tools have begun to bridge this gap. Bastos *et al.* [3] and the developers of MininetFed [17] provided container-based platforms for FL evaluation.

Our work differs from the existing literature by closing the loop between rigorous mathematical optimization and realistic systemic emulation. Table 1 summarizes the main contributions of this paper in contrast to the current literature.

3 The Federated Learning System Formulation

This section details the modeling of the FL system. The focus of Sect. 3.1 establishes the mathematical notation and describes the distributed learning network architecture analyzed. Section 3.2 introduces the mathematical framework that approximates the physical energy consumption and estimates the devices' processing times.

Table 1. Expanded qualitative comparison between this work and the Federated Learning literature.

Work	Optimization Focus	Solution Method	Opt Model	Validation Strategy	Emulation Tool
Ali *et al.* [1]	Client selection	Multi-criteria Greedy		Network Emulation	EdgeCloudSim
Bastos *et al.* [3]	Energy in IoT (RPL)	Energy-Aware Selection		Network Emulation	MininetFed
Kim *et al.* [11]	Energy and Convergence Time	Convex Optimization	x	Numerical	FL-Pytorch
Luo *et al.* [14]	Cost and Latency	Convex Opt. + Heuristic	x	Numerical	Not mentioned
Nishio *et al.* [16]	Device selection	Heuristic	x	Numerical	Not mentioned
Tahir *et al.* [18]	Device selection	Greedy Heuristic	x	Numerical	Not mentioned
Tran *et al.* [19]	Energy and Time	Analytical Relaxation	x	Numerical	Not mentioned
Wei *et al.* [20]	Energy and Selection	Selection and Topology	x	Numerical	Not mentioned
Zhou *et al.* [21]	Energy and Selection	Multi-objective Opt.	x	Numerical	Not mentioned
This Work	**Energy and Time**	**Convex Optimization**	**x**	**Network Emulation**	**MininetFed**

3.1 Distributed Neural Network Training

Let R be a tree-structured network consisting of a set of $\mathbf{N}$ clients $CD_1, \ldots, CD_{\mathbf{N}}$, and a Centralized Aggregation Server (AS). The CDs train local neural networks, while AS performs the global aggregation of weights or gradients sent by the CDs. In this topology, each $CD_n, n = 1, \ldots, N$ is connected to the AS by single-path routes comprising one or more hops. Furthermore, communication between CDs and the AS nodes of R is performed via a wireless network. An example of this topology with N = 6 is depicted in Fig. 1.

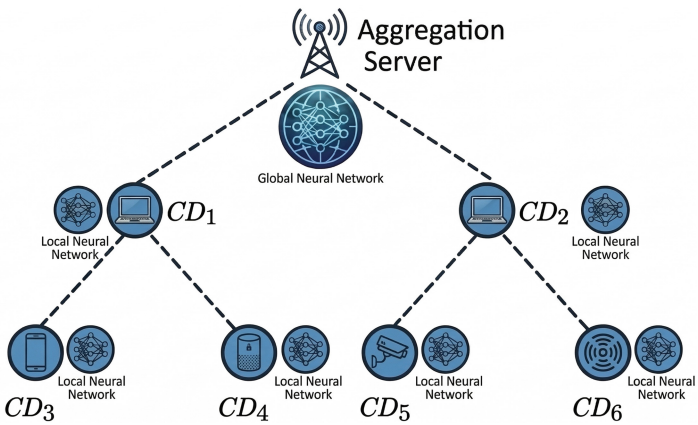


Fig. 1. Tree-structured topology of a FL.

In this topology, each client device $CD_n, n = 1, \ldots, N$ is characterized by its own set of input and output pairs $\mathbf{D_n}$ of size d_n, where each sample $\{x_n^i, y_n^i\}$, $x_n^i, y_n^i \in \mathbb{R}$, comprises a vector of s features and its expected output label. The parameter S_n specifies the size in bits of the CD_n dataset. Hardware constraints for each device are characterized by a CPU frequency $f_n \in [f_n^m in, f_n^m ax]$, a

chipset capacitance coefficient α_n, and a computational factor c_n (CPU cycles required to execute the model training task per sample). The parameters notation necessary for the problem formulation are summarized in Table 2.

The aforementioned scenario characterizes the learning problem as a typical classification problem. Therefore, each client device CD_n has the task of solving Problem 1, as defined in Eq. 1, considering the neural network weights w_n that minimize a local loss function $L(w)$ formulated as a sample loss function $l(x^i, y^i, w)$:

$$Prob.1: \quad w_n = \arg\min_{w_n} \; L(w_n), \text{ where } L(w_n) = \frac{1}{d_n} \sum_{i=1}^{d_n} l(x^i, y^i, w_n). \quad (1)$$

For instance, $l(x^i, y^i, w) = 1/2((x^i)^T w - y)^2$ [11]. Several methods can be employed to solve Problem 1; among them, SGD (stochastic gradient descent) is widely adopted [9]. AS, in turn, has the task of aggregating the distinct local weights w_n into a single object w. Furthermore, AS is responsible for determining the CPU frequency for each CD during the training phase. To achieve this, the AS leverages the constants characterizing each device, while maintaining no access to their local datasets.

Table 2. Notation Table - Federated Learning Model Variables

Symbol	Description
AS	Aggregation Server
CD	Client Device
N	Total number of client devices (CDs), $N \in \mathbb{N}$
n	Client device index, $n \in \{1, 2, \ldots, N\}$
s	Number of attributes (features), $s \in \mathbb{N}$
d_n	Number of samples of the n-th device, $d_n \in \mathbb{N}$
x_n^i	i-th vector of s input attributes of CD_n, $i = (1, 2, ..., d_n)$
y_n^i	i-th expected output label for x_n^i, $y_n^i \in \mathbb{R}$
$\mathbf{D_n}$	Set of attributes and labels of a CD_n: $\{x_n^i, y_n^i\}_{i=1}^{d_n}$
S_n	Size in bits of $\mathbf{D_n}$, $S_n \in \mathbb{R}$
f_n^{min}	Minimum CPU frequency of CD n, $f_n^{min} \in \mathbb{R}^+$
f_n^{max}	Maximum CPU frequency of CD n, $f_n^{max} \in \mathbb{R}^+$
α_n	Chipset capacitance of CD n, $\alpha_n \in \mathbb{R}^+$
c_n	CPU cycles per bit of training sample, $c_n \in \mathbb{R}^+$
w_n	Local model weights of CD_n, $w_n \in \mathbb{R}^K$, $K \in \mathbb{N}^+$
w	Global model weights (aggregated), $w \in \mathbb{R}^K$, $K \in \mathbb{N}^+$
$L(w_n)$	Local loss function of CD_n, $L(w_n) \in \mathbb{R}$
$l(x^i, y^i, w_n)$	Sample loss function, $l(\cdot) \in \mathbb{R}$
t	Training round index, $t \in \mathbb{N}$
w_n^t	Local weights of a CD_n in round t, $w_n^t \in \mathbb{R}^K$, $K \in \mathbb{N}^+$
w^t	Global model weights (aggregated), in round t. $w \in \mathbb{R}^K$, $K \in \mathbb{N}^+$
d_n^t	Number of samples of CD_n in round t, $d_n^t \in \mathbb{N}$
f_n^t	Allocated frequency for CD_n in round t, $f_n^t \in \mathbb{R}^+$
E_n^{cmp}	Computational energy consumption of CD_n, $E_n^{cmp} \in \mathbb{R}^+$
T_n^{cmp}	Computational time of CD_n, $T_n^{cmp} \in \mathbb{R}^+$

The way the AS aggregates the different weights can vary, as different methods are proposed in the literature. Perhaps the most classic algorithm, also adopted in this work, is *FedAvg* [15], where the local weights $w_n, n = 1, \ldots, \mathbf{N}$ are aggregated into w as defined in Eq. 2.

$$w = \frac{1}{d} \sum_{n=1}^{N} d_n \cdot w_n. \tag{2}$$

The FL scheme performed on the network is summarized in Algorithm 1, which introduces the index t associated with each training round (see Table 2).

Algorithm 1: Distributed Federated Learning Scheme

Input: T,$\{S_n^0, c_n, \alpha_n, f_n^{min}, f_n^{max}\}$
Output: Final global model w^T

1 The AS initializes and distributes w^0 to all $CD_n, n = 1, \ldots, N$;
2 Each CD_n sends parameters $\{c_n, \alpha_n, f_n^{min}, f_n^{max}\}$ to the AS;
3 **for** $t = 0$ **to** $T - 1$ **do**
  `// Coordination and Optimization Phase`
4 Each CD_n sends $\{S_n^t\}$ to the AS;
5 The server defines the devices that will participate in the training round;
6 The server executes the optimization model;
7 The server informs each CD_n of its operating frequency f_n^t;
  `// Local Training Phase at the CDs`
8 **for** *each client device* CD_n *in parallel* **do**
9 Each CD_n obtains w_n^{t+1} by solving Problem 1 for $L(w_n^t)$ via SGD;
10 Sends w_n^{t+1} to the AS;
11 **end for**
  `// Aggregation Phase at the Server`
12 The AS aggregates the received weights using Eq. 2:;
13 $w^{t+1} \leftarrow \frac{1}{d} \sum_{n=1}^{N} d_n \cdot w_n^{t+1}$;
14 The AS distributes the new global model w^{t+1} to the devices;
15 **end for**

- **Inputs and Outputs:** The algorithm receives essential configuration parameters, such as the total number of rounds T, and the physical parameters and hardware constraints for each CD_n ($f_{min}, f_{max}, c_n, \alpha_n, S_n$). The output is the optimized global model w^T.
- **Lines 1–2:**: The AS initializes the system by broadcasting the global model weights w^0 (an array initialized with arbitrary values without loss of generality) to all CD_n. In response, each client transmits its fixed hardware parameters to the server.
- **Line 3:** The main federated training loop begins, iterating from round $t = 0$ to $t = T - 1$.

- **Line 4:** Each device informs the server of the amount of data (S_n^t) it holds for that specific round. This step is necessary because the data volume at a client may vary as new data is acquired.
- **Line 5:** The server defines which clients will participate in the current training round. This is because certain optimization strategies select only a subset of clients to save energy.
- **Lines 6–7:** The server acts as a coordinator, executing the mathematical optimization model to determine the optimal CPU frequency for the selected clients. Subsequently, the server informs each CD_n of its specific operating frequency f_n^t.
- **Lines 8–11 (Local Training Phase):** A parallel loop begins where each selected client adjusts its CPU and computes the new model weights (w_n^{t+1}) by solving its local optimization problem (Prob. 1) using the SGD method. After the computation, the updated weights are sent back to the AS server.
- **Lines 12–14 (Aggregation Phase):** Upon receiving the local updates, the server aggregates them using Eq. 2 to form the new global model w^{t+1}. This updated model is then distributed to the devices, preparing the network for the next round.

3.2 CPU Processing Time and Energy Consumption of Client Devices

Based on the distributed training scheme presented in Sect. 3.1, this section establishes the mathematical formulation to quantify the CPU processing time and energy consumption for each CD. This formulation follows the several works in the literature, such as [11,19], and [20], the energy consumption E_n^{cmp} and the training time per round T_n^{cmp} for CD_n are presented in Eqs. 3 and 4:

$$E_n^{cmp} = \frac{\alpha_n}{2} c_n S_n f_n^2, \tag{3}$$

$$T_n^{cmp} = \frac{c_n S_n}{f_n}. \tag{4}$$

The Eqs. 3 and 4 are derived by adapting theoretical consumption and processing models. The variables used were previously defined in Sect. 3.1 and described in Table 2. Following [5,7], these equations were simplified for an ideal scenario, were the energy consumption and computational effort of the system itself are neglected, as well as energy losses through heating and leakage currents.

4 Optimization Problem Associated with Processing Time and Energy Consumption of Client Devices

Concerning the mathematical tools introduced in the previous sections, Sect. 4.1 presents the optimization problem. It incorporates an objective function designed

to balance energy consumption and processing time, which is subject to the problem constraints. Section 4.2 describes the methodology used to achieve the optimal solution, while Sect. 4.3 outlines the characteristics of the network instance constructed, which follow the structure of the example presented in Fig. 1.

4.1 The Mathematical Formulation

In this section, we present the formulation of the optimization problem of interest, described in Prob. 2, comprising Eqs. 5a to 5c:

$$Prob.2: \quad \min_{f_n, T^{cmp}} \quad \kappa T^{cmp} + \sum_{n \in \mathbf{N}} E_n^{cmp}, \tag{5a}$$

$$\text{s.t:} \quad T_n^{cmp} \leq T^{cmp} \quad \forall n \in \mathbf{N}, \tag{5b}$$

$$f_n^{min} \leq f_n \leq f_n^{max}, \quad \forall n \in \mathbf{N} \tag{5c}$$

Prob. 2 is a multi-objective optimization problem that seeks to balance two competing resources with its objective function (5a): the CPU processing time T^{cmp} and the energy consumption E_n^{cmp}. Since Prob. 2 is a bi-objective optimization problem, its objective function is compound by two conflicting criteria. The priority assigned to each cost (either energy consumption or round latency) is a subjective choice. The trade-off is managed by the parameter κ. When coupled with T^{cmp}, a is sufficiently high, κ renders the second term of the objective function negligible compared to the first. Conversely, as κ approaches 0, the first term becomes irrelevant. The parameter κ is also used to normalize these values, once they may have different orders of magnitude.

Constraint 5b imposes that the decision variable T^{cmp} is large enough so that all clients finish the task before the round ends, while constraint 5c ensures that f_n respects the computational power limits of each client device CD_n.

This formulation is a simplified version of the one proposed by [19]. In its original form, the computational resource optimization is a subproblem addressed jointly with the communication resource subproblem. We aim to investigate the feasibility and consequences of optimizing computational resources exclusively; consequently, the communication subproblem is omitted.

4.2 Obtaining the Optimal Solution

According to [19], the convex optimization problem presented in Sect. 4.1 can be solved optimally by employing a Lagrangian Relaxation [4], jointly with the Karush-Kuhn-Tucker (KKT) conditions [13]. The authors propose partitioning the set of $\mathbf{N}$ client devices into three subsets, N_1, N_2, N_3 ($N_i \cap N_j = \emptyset$ and $N_i \cup N_j = \mathbf{N}$, $i \neq j$) and prove mathematically that the optimal solution for the problem formulated in 4.1 is given by Eqs. 6 and 7a to 7d.

$$f_n^* = \begin{cases} f_n^{max}, & \forall n \in \mathcal{N}_1, \\ f_n^{min}, & \forall n \in \mathcal{N}_2, \\ \frac{c_n S_n}{T_{cmp}^*}, & \forall n \in \mathcal{N}_3, \end{cases} \tag{6}$$

$$T^*_{cmp} = \max\{T_{\mathcal{N}_1}, T_{\mathcal{N}_2}, T_{\mathcal{N}_3}\}, \tag{7a}$$

$$T_{N_1} = \max_{n \in N_1} \frac{c_n S_n}{f_n^{max}}, \tag{7b}$$

$$T_{N_2} = \max_{n \in N_2} \frac{c_n S_n}{f_n^{min}}, \tag{7c}$$

$$T_{N_3} = \left(\frac{\sum_{n \in \mathcal{N}_\ni} \alpha_n (c_n S_n)^3}{\kappa} \right)^{1/3}. \tag{7d}$$

To determine this partitioning, the procedure proposed in [19] is reproduced in Algorithm 2.

Algorithm 2: Determining partitions $\mathcal{N}_1$, $\mathcal{N}_2$ and $\mathcal{N}_3$ [19]

Input: $\mathcal{N}_1, \mathcal{N}_2 = \emptyset$, $\mathcal{N}_3 = \{CD_n, n = 1, 2, \ldots, d_n\}$
Output: $\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3$

1 Sort $\mathcal{N}_3$ by the value of $\frac{c_n S_n}{f_n^{min}}$ in ascending order;
2 $T_{\mathcal{N}_3} \leftarrow$ Value obtained from Equation (7d);
3 **for** $i \in \mathcal{N}_3$ **do**
4 **if** $\max_{n \in \mathcal{N}} \frac{c_n S_n}{f_n^{min}} \geq T_{\mathcal{N}_3}$ ***and*** $\mathcal{N}_1 == \emptyset$ **then**
5 $\mathcal{N}_1 = \mathcal{N}_1 \cup \{n\}$
6 $\mathcal{N}_3 = \mathcal{N}_3 \setminus \mathcal{N}_1$
7 Update $T_{\mathcal{N}_3}$
8 **end if**
9 **if** $\frac{c_i D_i}{f_i^{min}} \leq T_{\mathcal{N}_3}$ **then**
10 $\mathcal{N}_2 = \mathcal{N}_2 \cup \{i\}$
11 $\mathcal{N}_3 = \mathcal{N}_3 \setminus \{i\}$
12 Update $T_{\mathcal{N}_3}$
13 **end if**
14 **end for**
15 **return** $\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3$

- **Inputs and Outputs:** The algorithm begins by assuming that all clients will have their frequencies actively scaled, placing them all in set $\mathcal{N}_3$, while $\mathcal{N}_1$ and $\mathcal{N}_2$ start empty. The algorithm returns these three sets correctly populated.
- **Line 1:** The devices in set $\mathcal{N}_3$ are sorted in ascending order. The metric used for sorting is the time each device would take to process its data if it operated at its minimum possible CPU frequency ($\frac{c_n S_n}{f_n^{min}}$).
- **Line 2:** The target processing time for set $\mathcal{N}_3$ ($T_{\mathcal{N}_3}$) is initially calculated based on Eq. 7d.
- **Line 3:** A loop begins, iterating through each device i currently within set $\mathcal{N}_3$.

- **Lines 4–8 (Bottleneck Handling):** The algorithm evaluates whether a "bottleneck" device exists in the system. It checks if the maximum limiting processing time is greater than or equal to the current target time $T_{\mathcal{N}_3}$. If such a bottleneck is identified and set $\mathcal{N}_1$ is empty, this device is removed from $\mathcal{N}_3$ and included in $\mathcal{N}_1$. Since the group dynamics have changed, the target time $T_{\mathcal{N}_3}$ is immediately recalculated.
- **Lines 9–13 (Fast Client Handling):** The algorithm then seeks to identify highly efficient clients. It tests whether the iterated device i, even when operating at its minimum CPU frequency, would still complete its task in less time than the target $T_{\mathcal{N}_3}$. If this condition holds, the device is added to set $\mathcal{N}_2$ and removed from $\mathcal{N}_3$. Again, this removal requires $T_{\mathcal{N}_3}$ to be updated.
- **Line 15:** Once the loop concludes and all reallocations to handle bottlenecks and fast clients are performed, the algorithm returns the final partitions $\mathcal{N}_1$, $\mathcal{N}_2$, and $\mathcal{N}_3$ to the server.

4.3 Network Instance Description

The network instance used in the experiments follows a topology similar to the one shown in Fig. 1. It consists of $\mathbf{N} = 6$ client devices, in addition to the aggregation server (AS), organized in a binary tree topology with the AS as the root. Although the optimization methodology is scalable, the number of devices was limited due to computational resource constraints on the emulation host, without compromising the quality and reliability of the experimental results.

The configuration of each device and its parameters are presented in Table 3. The values for c_n were obtained in a preliminary experimental stage. The training process was conducted across a range of f_n values to produce an empirical characterization of T_n^{cmp}. Consequently, c_n was adjusted to align the empirical data with the theoretical predictions established in Eq. 4.

Table 3. Configuration of the client devices in the learning network

CD_n	S_n (MB)	c_n	f_n^{max} (GHz)	f_n^{min} (GHz)
CD_1	32.5	9	2.3	1.3
CD_2	54.1	5	2.8	1.3
CD_3	81.2	4	2.7	1.3
CD_4	43.3	8	2.1	1.3
CD_5	65.0	4	2.5	1.3
CD_6	21.6	11	2.1	1.3

On the other hand, since it depends on hardware characteristics, α_n was set to the same value for all devices as $\alpha_n = 2 \cdot 10^{-28}$, associated with the machine that emulated the network. This assumption does not affect the generality of the

results, as device heterogeneity is still guaranteed by the values of f_n^{min}, f_n^{max}, and c_n, given that the constant α_n is always multiplied by one of these factors. Other authors, such as [19] and [11], maintain this same assumption.

Each client device received samples from the MNIST dataset [12], a widely used benchmark in the literature. MNIST was selected deliberately: its well-understood convergence behavior allows the effects of frequency-based optimization to be isolated and attributed without confounding factors introduced by dataset complexity. This controlled setting facilitates reproducibility and direct comparison with prior emulation-based studies. Evaluating the approach on more complex datasets remains as future work.

A qualitative analysis of the optimal solution for Problem 2 can be conducted from Fig. 2, obtained by applying Algorithm 2 to the client configuration presented in Table 3. The graph shows the change in the optimal frequency value and also the processing time according to the increase in the value of κ. We will divide the solution analysis into different intervals of kappa:

- $\log(\kappa) \leq -0.35$: saving energy becomes so important that all devices execute at minimum frequency, taking the maximum possible time to finish the task; in this situation $\mathcal{N}_2 = \mathcal{N}$.
- $\log(\kappa) \cong 0.75$: T^{cmp} and f_n are not defined by boundary conditions, but rather by a value determined by Eqs. 7d and 6. In this situation, $\mathcal{N}_3 = \mathcal{N}$.
- $\log(\kappa) \geq 0.85$: In this region, a low T^{cmp} is chosen because this term has more weight in Eq. (5a). Here, the bottleneck device (CD_4) operates at maximum frequency, while the frequency of the other devices is adjusted to intermediate values ($\mathcal{N}_1 \neq \emptyset$).

The plots reinforce the role of κ in resource prioritization, as evidenced by the comparison between the two graphs in Fig. 2. When κ is adjusted to low values, the objective function prioritizes energy saving over latency, resulting in lower operating frequencies and higher processing times. Conversely, as κ increases, the priority shifts toward minimizing latency, leading to higher frequencies and increased energy consumption.

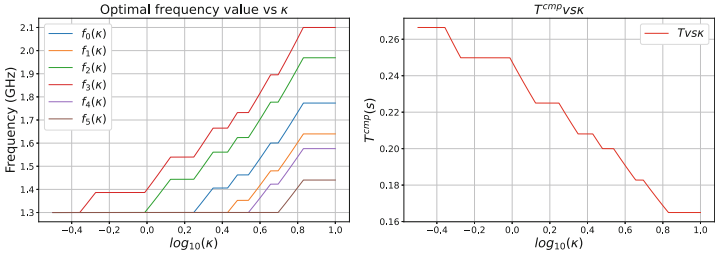


Fig. 2. Solution to Problem 2 for different values of κ.

5 Experiments and Results

Simulations were performed in virtual networks to validate the dependencies of energy and processing time discussed in the theoretical model. In particular, to evaluate whether saving resources generates impacts on the learning model's convergence and, if so, to discuss in which scenario this trade-off of efficiencies is advantageous.

To create the virtual network laboratories, MininetFed [17] was used. This tool allows the emulation of federated learning based on container networks managed by Docker. The emulation environment imposes inherent resource constraints, as each virtual device runs as an isolated Docker container sharing CPU and memory with the host machine. Scaling beyond six devices would compromise the reliability of timing and energy measurements - a known challenge in container-based FL emulation. Importantly, the optimization methodology itself is analytically scalable to arbitrary $\mathbf{N}$, as the analytical solution in Sect. 4.2 is determined by Algorithm 2 with a $\mathcal{O}(n \log n)$ complexity; the experimental scope is a constraint of the validation environment, not of the proposed approach. The differential of MininetFed is the ability to emulate networks sensitive to energy consumption, which is crucial for the objective of this work, as well as the possibility to define heterogeneous devices regarding computational power, amount of data, and other variables to which Problem 2 is sensitive.

The general network configuration (topology, configuration, and communication protocols) was inspired by [2], while changes to the server and clients were necessary to accommodate the proposed problem. Section 5.1 describes one of the modifications made to the library so that the energy consumption measurement was effective regarding CPU frequency changes. Next, Sect. 5.2 presents the training methodologies, directly linked to the specifications of Algorithm 1. Finally, the results are presented and discussed in Sect. 5.3.

5.1 Energy Consumption and Power

Although the energy consumption throughout a training round is estimated by Eq. 3 (Sect. 3.2), it is worth remembering that these are approximations. The real time of the training process can oscillate, just as the computer's clock does not remain constant [21].

Therefore, throughout the experimentation, energy consumption was measured through a function that calculates the power (J/s) consumed by the CPU, defined in Eq. 8 [7]:

$$P(f) = CV^2 f, \tag{8}$$

where C is a constant adjusted experimentally by varying f and measuring the actual CPU consumption of the experiment host. Thus, the instantaneous power $P(f, t)$ is measured by reading the frequency at predefined time intervals Δt, thus arriving at values close to reality. The energy spent in this interval is then given by $\Delta E(t) = P(f, t) \cdot \Delta t$. This is an adaptation of [7], considering only the CPU workload consumption, since this is the investigation target of this study.

Implementing the power function of Eq. (8) was necessary because the default function used by [3] only considers the CPU usage percentage and did not prove sensitive to CPU frequency changes.

5.2 Experiments

The network devices were configured following the instance from Table 3. Different training strategies were implemented, combining different forms of data distribution, client selection at each round, and resource consumption, as described below:

Regarding the Training Methodology

- **Standard:** in this configuration, no optimization is done, all devices participate in all rounds, with frequency control assigned to the operating system.
- **LeastEnergy:** Client selection proposed by [2], where only the set $\mathcal{M} \subseteq \mathcal{N}$ of devices that consumed less than the average consumption of the previous round participates in the current round (Acting on Line 5 of Algorithm 1). Frequency is also controlled by the kernel.
- **Opt and Opt&fixed T^{cmp}:** In both, the frequency control is defined by the solution to Problem 2, with namespace governor. However, "Opt&fixed T^{cmp}" is the only one where the training round stopping criterion is not a maximum number of epochs (defined as 10), but rather T^{cmp*}, the result of Equation (7a).
- **Opt&LeastEnergy:** The $\mathcal{M}$ devices selected by LeastEnergy are the input for Problem 2.

For the methodologies *Opt*, *Opt&T^{cmp} fixed*, and *Opt&LeastEnergy*, the parameter κ was set to $10^{3/4}$. This value was considered suitable for the experiments because, as shown in Fig. 2, it represents a solution regime where none of the devices operate at their maximum or minimum frequency limits.

Regarding Data Distribution

- **mid-iid:** The devices receive random samples from the MNIST dataset; however, all of them have examples from the 10 possible labels of this dataset, but in unbalanced quantities among them.
- **non-iid:** The data is unbalanced, and there may be devices that "do not know" certain image categories. In the experiments, each client received images with 4 categories individually chosen at random among the 10 possible ones. (Note: the non-iid graphs only include results for three methodology scenarios to facilitate visualization).

5.3 Results

Figure 3 illustrates that the Standard training mode is the least efficient in terms of both consumption per round and time. The *opt-fixed* T^{cmp} methodology presents the lowest absolute energy consumption at the end of the process. However, fixing the time per round directly impairs the model's quality, as evidenced in Fig. 4. This fixed time limit leads to a smaller total number of epochs, which justifies the shorter training time shown in Fig. 3c. Consequently, to achieve the same target accuracy, the total energy consumed would be higher.

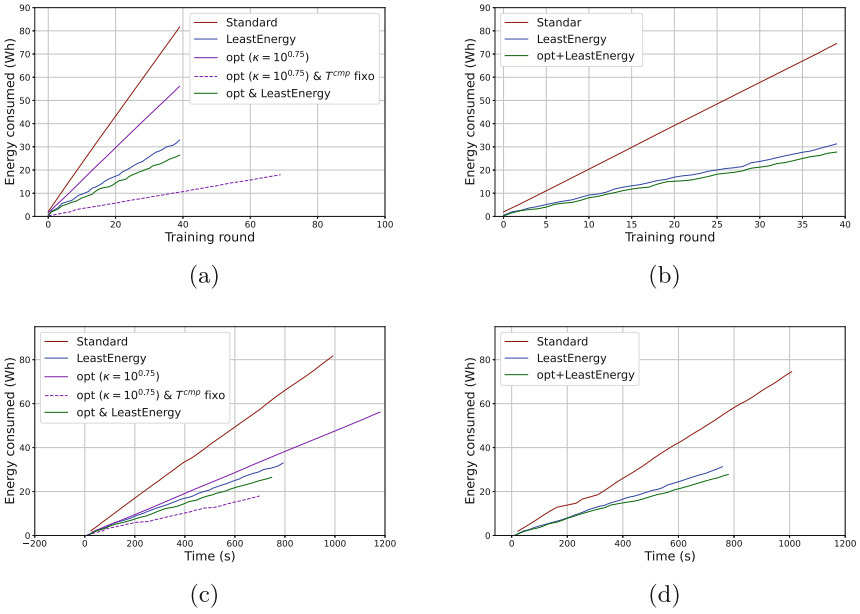
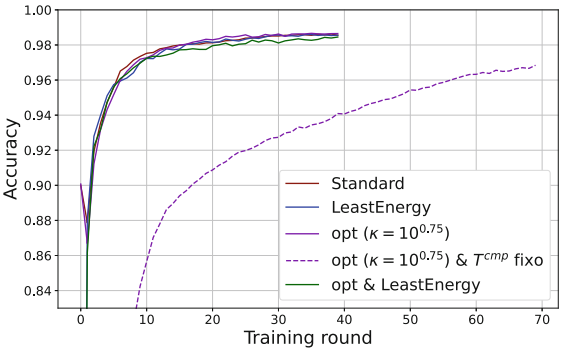


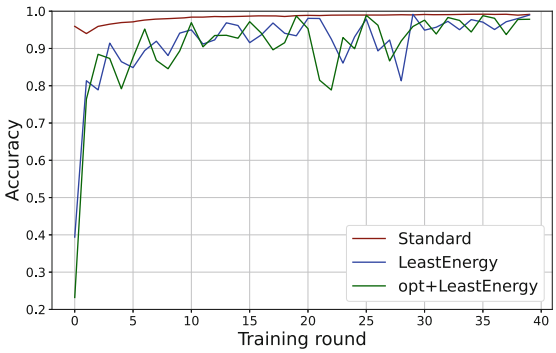
Fig. 3. Energy consumption behavior for different scenarios: (a) Energy vs Round (mid-iid), (b) Energy vs Round (non-iid), (c) Energy vs Time (mid-iid), (d) Energy vs Time (non-iid).

Conversely, comparing the pairs "Standard vs. Opt" and "LeastEnergy vs. Opt&Least Energy" demonstrates that energy optimization reduces total consumption. Furthermore, as shown in Fig. 4, it does not harm the model's performance. This result is expected, given that the number of epochs remained constant across these cases. The graphs in Figs. 3c and 3d also show that the training process time, for the same number of rounds, is not drastically altered by the optimization model under analysis, except for the fixed-time scenario.

Analyzing Fig. 5 in conjunction with the others brings an important reflection: reaching a target accuracy demands the same or less energy when employing the optimized methodology instead of the standard client selection approach. Specifically, the optimization approach achieves a reduction of 30% and 20%



(a)



(b)

Fig. 4. Accuracy evolution behavior for different scenarios: (a) Mean accuracy for mid-iid data, (b) Mean accuracy for non-iid data.

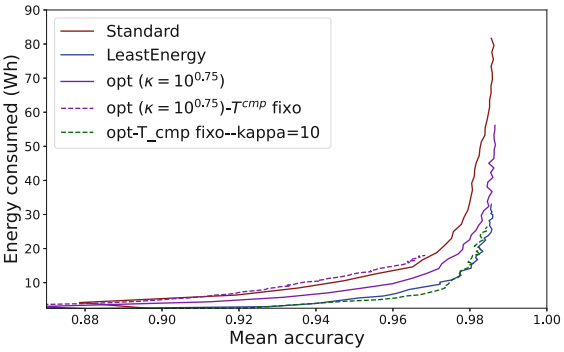


Fig. 5. Relationship between consumption and accuracy for mid-iid data.

in absolute accumulated energy consumption when compared to the standard selection of all clients and the LeastEnergy strategy, respectively.

6 Conclusion

The investigation conducted in this work allowed for the evaluation of the impact of resource optimization strategies in Federated Learning networks. The results demonstrate that, although strict control of processing time favors a reduction in energy consumption per round, this approach imposes a critical trade-off, as fixing a time limit reduces the total number of epochs, significantly impairing the quality and convergence of the final model.

On the other hand, the proposed frequency optimization algorithm, based on Lagrangian Relaxation and KKT Conditions, proved to be effective. It was evidenced that, for a previously defined client selection, controlling the computational effort via frequency adjustment is capable of optimizing energy consumption without leading to losses in model performance or accuracy. This effectiveness was experimentally validated for both partially unbalanced data distributions (mid-iid) and scenarios where certain clients lack specific categories (non-iid). Furthermore, the combination of the LeastEnergy selection methodology with the optimization algorithm enhanced the accumulated energy savings compared to the standard training mode.

As perspectives for future work, we suggest evolving the optimization model to include the number of local epochs as a decision variable, in addition to evaluating communication-related costs. Furthermore, we propose integrating client selection criteria directly into the frequency optimization model to enhance the trade-off between execution time and energy efficiency in low-power networks.

Another suggestion for future optimization involves incorporating suitable methods to measure the computational effort of hardware accelerators, such as GPUs, NPUs (Neural Processing Units), and TPUs (Tensor Processing Units). These architectures are becoming increasingly common in machine learning and neural network applications, even in edge computing scenarios. Additionally, scaling the experimental evaluation to larger device counts, potentially through distributed emulation hosts or hardware testbeds, remains as future work.

Acknowledgments. The authors acknowledge partial funding from the Coordination for the Improvement of Higher Education Personnel (CAPES) - Funding Code 001; the National Council for Scientific and Technological Development (CNPq) - Funding Codes 442344/2023-6, 444602/2024-0, and 407466/2025-8; the Espírito Santo Research and Innovation Foundation (FAPES) - Grants No. 2026-6NX7B, 2026-B74MN, 2024-9G1WP, 2023-RWXSZ; and Fapesp/CGI.br (2020/05182-3).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

Declaration of Generative AI and AI-Assisted Technologies in the Writing Process. During the preparation of this work the authors used Gemini/Google in

order to improve language and readability. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

1. Ali, N., Aloi, G., Gravina, R., Savaglio, C., Sodhro, A.H., Fortino, G.: FedHeur: multi-heuristic client selection for task offloading in federated learning. In: 2025 21st International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT), pp. 816–821 (2025). https://doi.org/10.1109/DCOSS-IoT65416.2025.00124
2. Bastos, J., Batista, J., Fontes, R., Cerqueira, E., Villaça, R., Mota, V.F.S.: A lightweight emulation framework for energy-aware federated learning. In: Proceedings of the ACM SIGCOMM 2025 Posters and Demos, ACM SIGCOMM Posters and Demos '25, pp. 130–131. Association for Computing Machinery, New York, NY, USA (2025). https://doi.org/10.1145/3744969.3748395
3. Bastos, J.J.S., Batista, J.P.C., dos Reis Fontes, R., Cerqueira, E., Villaça, R.S., Mota, V.F.S.: Otimizando energia no aprendizado federado em redes de baixa potência e com alta taxa de perda de pacotes. In: Anais do XLIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos. pp. 43–56. SBC, Porto Alegre, RS, Brasil (2025). https://doi.org/10.5753/sbrc.2025.5786, https://sol.sbc.org.br/index.php/sbrc/article/view/35121
4. Boyd, S.P., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
5. Burd, T.D., Brodersen, R.W.: Processor design for portable systems. J. VLSI Sig. Process. Syst. **13**(2–3), 203–221 (1996)
6. Daly, K., Eichner, H., Kairouz, P., McMahan, H.B., Ramage, D., Xu, Z.: Federated learning in practice: reflections and projections. In: 2024 IEEE 6th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA), pp. 148–156 (2024). https://doi.org/10.1109/TPS-ISA62245.2024.00026
7. De Vogeleer, K., Memmi, G., Jouvelot, P., Coelho, F.: The energy/frequency convexity rule: modeling and experimental validation on mobile devices. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Waśniewski, J. (eds.) PPAM 2013. LNCS, vol. 8384, pp. 793–803. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55224-3_74
8. Efthymiadis, F., Karras, A., Karras, C., Sioutas, S.: Advanced optimization techniques for federated learning on non-IID data. Fut. Internet **16**(10) (2024). https://doi.org/10.3390/fi16100370, https://www.mdpi.com/1999-5903/16/10/370
9. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016). http://www.deeplearningbook.org
10. Guendouzi, B.S., Ouchani, S., Assaad, E.L., H., EL Zaher, M.: A systematic review of federated learning: challenges, aggregation methods, and development tools. J. Netw. Comput. Appl. **220**, 103714 (2023). https://doi.org/10.1016/j.jnca.2023.103714, https://www.sciencedirect.com/science/article/pii/S1084804523001339
11. Kim, J., Kim, D., Lee, J., Hwang, J.: A novel joint dataset and computation management scheme for energy-efficient federated learning in mobile edge computing. IEEE Wirel. Commun. Lett. **11**(5), 898–902 (2022). https://doi.org/10.1109/LWC.2022.3147236

12. LeCun, Y., Cortes, C., Burges, C.: MNIST handwritten digit database. ATT Labs [Online]. http://yann.lecun.com/exdb/mnist**2** (2010)
13. Li, M.: Generalized Lagrange multiplier method and KKT conditions with an application to distributed optimization. IEEE Trans. Circ. Syst. II Exp. Briefs **66**(2), 252–256 (2019). https://doi.org/10.1109/TCSII.2018.2842085
14. Luo, S., Chen, X., Wu, Q., Zhou, Z., Yu, S.: HFEL: joint edge association and resource allocation for cost-efficient hierarchical federated edge learning. IEEE Trans. Wirel. Commun. **19**(10), 6535–6548 (2020). https://doi.org/10.1109/TWC.2020.3003744
15. McMahan, B., Moore, E., Ramage, D., Hampson, S., Aguera y Arcas, B.: Communication-efficient learning of deep networks from decentralized data. In: Singh, A., Zhu, J. (eds.) Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 54, pp. 1273–1282. PMLR (2017). https://proceedings.mlr.press/v54/mcmahan17a.html
16. Nishio, T., Yonetani, R.: Client selection for federated learning with heterogeneous resources in mobile edge. In: 2019 IEEE International Conference on Communications (ICC), ICC 2019, pp. 1–7 (2019). https://doi.org/10.1109/ICC.2019.8761315
17. Sarmento, E.M., Bastos, J.J., Villaca, R.S., Comarela, G., Mota, V.F.: MininetFed: a tool for assessing client selection, aggregation, and security in federated learning. In: 2024 IEEE 10th World Forum on Internet of Things (WF-IoT), pp. 1–6. IEEE (2024)
18. Tahir, M., Ali, M.I.: Multi-criterion client selection for efficient federated learning. Proc. AAAI Symp. Ser. **3**(1), 318–322 (2024). https://doi.org/10.1609/aaaiss.v3i1.31227, https://ojs.aaai.org/index.php/AAAI-SS/article/view/31227
19. Tran, N.H., Bao, W., Zomaya, A., Nguyen, M.N.H., Hong, C.S.: Federated learning over wireless networks: optimization model design and analysis. In: IEEE Conference on Computer Communications, IEEE INFOCOM 2019, pp. 1387–1395 (2019). https://doi.org/10.1109/INFOCOM.2019.8737464
20. Wei, X., Ye, K., Shi, X., Xu, C.Z., Wang, Y.: Joint participant and learning topology selection for federated learning in edge clouds. IEEE Trans. Parallel Distrib. Syst. **35**(8), 1456–1468 (2024). https://doi.org/10.1109/TPDS.2024.3413751
21. Zhou, Y., Liu, X., Lei, L.: Multi-objective optimization for bandwidth-limited federated learning in wireless edge systems. IEEE Open J. Commun. Soc. **4**, 954–966 (2023). https://doi.org/10.1109/OJCOMS.2023.3266389

2D CNN-Based Link Prediction in Opportunistic Networks

Vinícius da Silva Thiago[1(✉)], Ronaldo Moreira Salles[1,2], Julio Cesar Duarte[1], and Gabriela Moutinho de Souza Dias[1]

[1] Instituto Militar de Engenharia, Rio de Janeiro, RJ 22290-270, Brazil
{vinicius.thiago,salles,duarte,gabriela}@ime.eb.br
[2] CIICESI, ESTG, Instituto Politécnico do Porto, Porto, Portugal
http://www.ime.eb.mil.br

Abstract. Opportunistic Networks are mobile wireless networks without fixed infrastructure, presenting unique challenges for link prediction tasks in computer networks. Although several approaches have been proposed to address this problem, the application of such techniques in opportunistic networks remains a promising strategy for improving packet delivery. This paper presents an analysis of a two-dimensional convolutional neural network architecture for link prediction in opportunistic networks. The experiments were conducted using three real-world datasets: ITC, Infocom06, and MIT Reality Mining. The model achieved competitive performance across all datasets, with AUC values of 0.9929, 0.9833, and 0.9812, and PRAUC values of 0.8777, 0.8321, and 0.8921, respectively. Additionally, experiments were conducted to evaluate link prediction across multiple future intervals, aiming to assess the feasibility of longer-term predictions to support packet forwarding in opportunistic networks. The results indicate that long-term prediction has the potential to enhance routing protocols by improving forwarding decisions.

Keywords: Opportunistic Networks · Link Prediction · Machine Learning · Deep Learning · CNN

1 Introduction

Opportunistic networks are a subclass of Delay Tolerant Networks (DTNs) in which continuous end-to-end connectivity cannot be guaranteed and delayed packet delivery is acceptable. In these networks, communication occurs in the absence of fixed infrastructure and relies on mobile nodes equipped with short-range wireless interfaces, such as Bluetooth. In such environments, a direct path between a source and a destination is often unavailable. Consequently, nodes depend on opportunistic encounters with other nodes to establish temporary communication links and enable packet forwarding. This characteristic gives rise to the term "opportunistic", as communication takes place by exploiting transmission opportunities whenever they arise. To achieve message dissemination

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 287–302, 2026.
https://doi.org/10.1007/978-3-032-30491-9_19

under these conditions, nodes adopt the store-carry-forward paradigm, in which packets are temporarily stored, physically carried during node mobility, and forwarded only when a suitable contact opportunity becomes available. This operational model makes opportunistic networks particularly well suited for environments characterized by intermittent, sparse, or entirely absent connectivity [10].

Nevertheless, nodes in opportunistic networks typically operate under severe resource constraints, particularly with respect to energy supply and storage capacity. As a result, these resources must be carefully managed to prevent premature node depletion and the consequent drop of buffered packets. In this context, a challenge in forwarding protocols is selecting the next-hop node responsible for carrying a packet closer to its destination. Therefore, designing forwarding strategies that maintain an adequate delivery ratio while avoiding excessive resource consumption remains a widely studied problem in opportunistic networking.

The literature provides numerous examples of situations in which users can benefit from opportunistic communication, including disaster situations [1, 2,9,10,16], battlefields [1,2,5,10,28], overloaded cellular networks [33,34] and industrial environments [17]. In these contexts, the store-carry-forward routing paradigm is implemented through intermediate nodes. This mechanism is supported by an additional network layer called the bundle, or packet layer, positioned between the transport and application layers. This layer enables nodes to store complete packets or their fragments, facilitating data delivery despite intermittent connectivity [18].

In addition, link prediction is a traditional problem in computer networks that aims to forecast future connections between nodes. Based on the spatial and temporal characteristics observed in the network, predicting such connections is particularly challenging in opportunistic environments due to node mobility and the possibility of abrupt disconnections, for example, caused by battery depletion. Despite these challenges, accurate link prediction can significantly assist routing algorithms in selecting the most appropriate next-hop node. Improved prediction quality leads to more efficient forwarding strategies, enabling higher packet delivery rates to the destination while reducing overall network resource consumption [14].

One approach to link prediction in networks is the use of machine learning techniques. This approach allows artificial intelligence models to be trained on historical network data and, after learning underlying patterns of network behavior, to predict the existence of future links between nodes.

The primary contributions of this paper are twofold. First, a two-dimensional convolutional neural network is employed for link prediction on real-world opportunistic network datasets, demonstrating its ability to capture spatial-temporal interaction patterns. Second, an analysis of link prediction over multiple future time intervals is provided, showing that the proposed approach maintains strong predictive performance even for longer prediction horizons.

The remainder of this article is organized as follows. Section 2 formalizes the link prediction problem in opportunistic networks and describes the adopted methodology. Section 3 reviews related work on link prediction in opportunistic networks using machine learning techniques. Section 4 presents the experimental setup, reports the obtained results, and provides an analysis with existing approaches. Section 5 discusses the results, and Sect. 6 concludes the article.

2 Link Prediction in Opportunistic Networks

Machine learning techniques have been widely applied in dynamic networks, where long-term link prediction is particularly challenging [6]. The use of deep learning for link prediction offers several advantages, as these models are capable of automatically capturing intrinsic patterns in data, eliminating the need for manual feature engineering and reducing the analytical effort required [3].

According to [27], to formulate the link prediction problem as a machine learning task, an opportunistic network can be represented as a sequence of adjacency matrices $A^{1:T} = \{A^1, A^2, \ldots, A^i, \ldots, A^T\}$, where each matrix A^i represents a matrix whose indices correspond to the network nodes at a time interval i ($i \in \{1, 2, \ldots, T\}$). In this representation, if a connection exists between two nodes a and b, then $A^i(a, b) = 1$; otherwise, $A^i(a, b) = 0$. Following the formulation in [25], the link prediction problem is defined as follows: given a sequence of adjacency matrices, $A^{t:t+\tau-1} = \{A^t, A^{t+1}, \ldots, A^{t+\tau-1}\}$, the objective is to predict the subsequent γ matrices $A^{t+\tau:t+\tau+\gamma-1} = \{A^{t+\tau}, A^{t+\tau+1}, \ldots, A^{t+\tau+\gamma-1}\}$, where t denotes the initial time interval, τ denotes the length of the input sequence, and γ denotes the length of the output sequence. The overall structure is illustrated in Fig. 1.

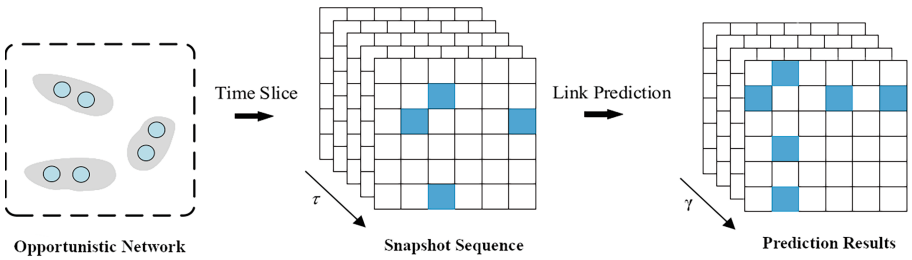


Fig. 1. Adjacency matrices' prediction. Adapted from [25].

Thus, the link prediction task can be formulated as a binary classification problem applied to the entries of the adjacency matrix, where the value 1 indicates the presence of a link and the value 0 indicates its absence.

2.1 Evaluation Metrics

This section introduces two evaluation metrics used to assess the prediction performance in this study. According to [25], the area under the receiver operating characteristic curve (AUC) is widely used for evaluating classification models. The metric is defined as

$$AUC = \frac{n' + 0.5n''}{n}, \tag{1}$$

where n is the number of comparisons in a sample, n' is the number of comparisons in which the predicted value for an existing link is greater than that for a non-existing link, and n'' is the number of comparisons in which both values are equal. The main advantage of this metric is that it provides a performance measure independent of the classification threshold, as well as being less sensitive to misclassification costs and class imbalance.

The area under the precision-recall curve (PRAUC) is commonly used to evaluate performance in sparse or highly imbalanced settings. According to [25], it can be computed as

$$PRAUC = \frac{1}{N} \sum_{u=1}^{N} \sum_{v=1}^{N} \frac{|A(u,v) = 1 \cap P(u,v) = 1|}{|A(u,v) = 1|}, \tag{2}$$

where $A(u,v) = 1$ indicates the existence of a link between nodes u and v, and $P(u,v) = 1$ indicates that a link between these nodes is predicted to exist. In highly imbalanced datasets, PRAUC is preferred because it directly evaluates the classifier's performance on the positive class, which is typically rare. By jointly emphasizing precision and recall, it provides a more informative assessment of the model's ability to identify positive instances. Both AUC and PRAUC range from 0 to 1, where 1 corresponds to the best possible performance.

3 Related Work

In recent years, link prediction in opportunistic networks has been extensively studied using machine learning techniques, involving a variety of datasets, models, and evaluation metrics.

Early approaches primarily relied on Recurrent Neural Networks (RNNs) to model temporal dependencies in node interactions. For example, [4] proposed a link prediction RNN-based method, in which the interaction history between node pairs is divided into network snapshots and represented as sequences of vectors. These sequences are used as input to a model called RNN-LP, which captures the hidden features of the vector sequence, enabling the correlation between nodes to be learned and the connection-formation patterns (which change frequently in opportunistic networks) to be extracted. To address well-known limitations of standard RNNs, such as vanishing gradients and difficulty in modeling long-term dependencies, the authors employed Long Short-Term Memory (LSTM) units. The model was evaluated using only the ITC dataset [21], achieving an AUC of 0.9780.

[15] proposed a link prediction model called BRRN-LP, which represents connection information as correlation vectors, where the correlation reflects the frequency and duration of node interactions. The model is based on a Bayesian Recurrent Neural Network (BRNN), specifically a Bayesian LSTM, which incorporates uncertainty into the network parameters while using gating mechanisms to preserve relevant temporal information. By estimating the posterior distribution of the network weights, the approach aims to improve robustness and reduce model complexity. The model was evaluated using only the MIT Reality Mining dataset [7], considering AUC and precision metrics, and achieved an AUC of 0.985.

[31] proposed a link prediction approach based on Deep Belief Networks (DBNs). In this method, traditional similarity metrics, including Katz Index, Resource Allocation, Adamic-Adar, and Common Neighbor, are used to construct input feature vectors for the DBN, focusing on non-repeated links. The DBN is formed by stacking multiple Restricted Boltzmann Machines (RBMs), trained in a greedy layer-wise manner, enabling the extraction of hierarchical representations from the data. The network is divided into temporal snapshots (10-min and 30-min intervals), from which adjacency matrices are generated. The model was evaluated using only the ITC dataset, achieving an AUC of 0.98.

[13] proposed a method that combines a Markov transition probability matrix with a similarity index matrix called improved random walk with restart (IRWR). The opportunistic network is divided into snapshots, and a similarity matrix is computed for each snapshot. The proposed model is called IRWR-DBN, since it uses a DBN that extracts the temporal characteristics of the network. The model was evaluated on the ITC and MIT Reality datasets using AUC and precision metrics. It achieved an AUC of 0.9591 on the ITC dataset and 0.9789 on the MIT Reality dataset.

More recent studies have shifted toward end-to-end deep learning models that jointly learn spatial and temporal representations. In [23], a correlation factor based on the number and duration of connections between nodes is used to assign weights to the links in the adjacency matrix of the opportunistic network. These matrices, computed over different time intervals, are then combined to form a three-dimensional structure. The proposed architecture consists of five convolutional layers, two fully connected layers, and an output layer. For evaluation, only the ITC dataset was used, and performance was assessed using accuracy and precision metrics.

In [25], the authors proposed a link prediction model based on the Transformer framework, composed of four main components: a topology feature extractor, a temporal feature extractor, a feature fusion module, and a vector mapping model. An attention mechanism is employed to extract topological features by aggregating information from neighboring nodes, enabling the model to learn long-term temporal dependencies and capture global network evolution patterns. The topological and temporal features are then fused, and the resulting feature vectors are mapped to predict the future network topology. The model was evaluated on the ITC, MIT, and Infocom06 [21] datasets using AUC and

PRAUC metrics. For the ITC dataset, the model achieved an AUC of 0.9708 and a PRAUC of 0.8915; for the Infocom06 dataset, it obtained an AUC of 0.9580 and a PRAUC of 0.7240; and for the MIT dataset, the results were an AUC of 0.9738 and a PRAUC of 0.8717.

In [22], the authors proposed a link prediction model called Continuous-graph Attentional Neural Networks for temporal link prediction (LP-CGA). The model is built upon an autoencoder architecture and integrates graph neural networks with attention-based ordinary differential equations. Experiments were conducted using the ITC and Infocom06 datasets, where opportunistic networks were represented as adjacency matrices. Model performance was evaluated using AUC and PRAUC metrics. For the ITC dataset, the model achieved an AUC of 0.9576 and a PRAUC of 0.8907, while for the Infocom06 dataset, it obtained an AUC of 0.9493 and a PRAUC of 0.7366.

In [12], a link prediction method for dynamic networks based on a Continuous Gated Recurrent Graph Convolutional Network (CGR-GCN) was proposed. The approach utilizes common neighbor indices and link attributes to capture network information. The authors incorporate Neural Ordinary Differential Equations (NODEs) into a Temporal Graph Convolutional Network (T-GCN), which combines graph neural networks with RNNs, and introduce a module designed to extract both short- and long-term spatio-temporal features. The model learns the network evolution locally through a graph convolutional network that employs gated recurrent units and NODEs. It then applies temporal convolutions with linear gates to extract global spatiotemporal features. Finally, link prediction is performed in the dynamic network using a fully connected dense neural network. The approach is based on the idea that traditional RNNs handle network snapshots in a discrete manner, limiting their ability to learn continuous dynamics. The authors argue that many existing methods rely solely on adjacency matrices, neglecting important information about link dynamics within snapshots. The model was evaluated on the ITC, Hyccups, and MIT Reality datasets using AUC, PRAUC, and F1-score metrics and compared with other methods. For the ITC dataset, the model achieved an AUC of 0.9681 and a PRAUC of 0.8653, while for the MIT dataset, it achieved an AUC of 0.9434 and a PRAUC of 0.8570.

Hybrid architectures that combine attention mechanisms, temporal convolutions, and graph-based representations have also been proposed. [24] proposed a link prediction method based on spatiotemporal attention mechanisms and temporal convolutional networks, called the Spatial-Temporal Attention and Temporal Convolutional Network (STA-TCN). The approach partitions opportunistic networks into discrete snapshots and uses a state matrix derived from the adjacency matrix, link occurrence frequency, and link duration. Attention and convolutional mechanisms are integrated into an encoder–decoder structure, followed by a fully connected dense neural network. To further enhance prediction performance, the authors transform autocorrelated errors into uncorrelated ones. The proposed method was evaluated using the ITC, MIT, and Infocom06 datasets. The metrics evaluated were AUC, precision, and F1-score. The model

achieved an AUC of 0.9644 on the ITC dataset, 0.9105 on the Infocom06 dataset, and 0.9789 on the MIT dataset.

Among the existing approaches, [27] compared several machine learning methods for link prediction in opportunistic networks, including artificial neural networks, convolutional neural networks (CNNs), RNNs, and attention-based models. The experiments were performed using the ITC and the Infocom06 datasets and evaluated performance in terms of accuracy, precision, recall, F1-score, AUC, and PRAUC. The authors concluded that CNNs employing two-dimensional convolutions represent a promising approach for link prediction in opportunistic networks, as they provide a good balance between predictive performance and architectural simplicity when compared to more complex models such as transformers and continuous-time graph neural networks (GNN). For the ITC dataset, the model achieved an AUC of 0.9840 and a PRAUC of 0.7893, while for Infocom06, it obtained an AUC of 0.9831 and a PRAUC of 0.8321.

[32] proposed a dynamic link prediction framework, named CLTGAT, which integrates contrastive learning (CL) with temporal graph attention network (TGAT). Contrastive learning is an unsupervised approach for graph representation learning. Unlike approaches based on sequences of adjacency matrices, CLTGAT employs duration-driven measures to select positive samples. The model captures network dynamics by modeling two types of temporal events: connections and disconnections. The framework was evaluated using AUC and PRAUC metrics on the Infocom05, Hyccups, and Infocom06 datasets. On the Infocom06 dataset, it achieved an AUC of 0.965 and a PRAUC of 0.962.

In addition to prediction-focused models, some studies incorporate link prediction into routing strategies. For example, [26] proposed EQ-STAR, an energy-efficient routing approach that combines encounter probability estimation with energy-aware optimization to enhance network performance. The method integrates GNNs and reinforcement learning (RL) to identify high-quality reachable paths between nodes, intending to maximize encounter probabilities while minimizing and balancing energy consumption. To estimate encounter probabilities, the approach employs the DySAT-pro model, which captures periodic patterns in the network. Subsequently, an energy-efficient quality Markov decision process (EQ-MDP) is used to jointly optimize energy consumption and encounter probabilities, enabling the selection of the most suitable paths between source and destination nodes. The next-hop relay node is then selected based on the reachable path obtained from the EQ-MDP. The proposed DySAT-pro model was evaluated on the Infocom05 and Infocom06 datasets using AUC, PRAUC, and F1-score metrics. On the Infocom06 dataset, it achieved an AUC of 0.8611 and a PRAUC of 0.8507.

[30] proposed a routing protocol for maritime opportunistic networks that employs encounter probabilities, mobility information, node resource availability, and link state prediction to guide routing decisions. The protocol first estimates the future connectivity status between a node and its neighbors using a link prediction method. When the predicted connectivity equals 1, the remaining features are used to prioritize packet forwarding. The proposed prediction

scheme, named CNN-EAA, combines CNN with an attention mechanism called Effective Additive Attention (EAA), comprising a convolutional encoder and a SwiftFormer-based encoder. The CNN is then trained using a synthetically generated maritime communication dataset. Although the overall routing protocol was evaluated, the study does not report results based on prediction-specific performance metrics.

[11] proposed a link prediction method based on Graph-BERT, in which the original network graph is decomposed into a set of local subgraphs. Each subgraph is constructed around a primary node and a fixed number of the most relevant neighboring nodes, selected based on a similarity criterion. This selection is performed using a top-K strategy, analogous to the KNN algorithm. The resulting subgraphs are used as input during both the training and testing phases. A transformer-based architecture is then used to encode and decode node features. Based on the learned representations, the model estimates the probability of link existence between pairs of nodes. To evaluate the effectiveness of the proposed approach, the authors conducted experiments on a synthetic dataset, using accuracy as the evaluation metric.

Table 1 shows a comparison between the related works on link prediction in opportunistic networks. It can be observed that a wide range of machine learning techniques has been employed in related work for the task of link prediction in opportunistic networks. The AUC metric is the most commonly used to evaluate predictions, while PRAUC is also widely adopted. The ITC, Infocom06, and MIT Reality datasets are extensively used for link prediction in opportunistic networks.

Table 1. Summary of Related Work on Link Prediction in Opportunistic Networks

Ref.	Model	Metrics	Dataset
[4]	LSTM	AUC	ITC
[15]	BRNN	AUC, Precision	MIT Reality
[31]	DBN	AUC	ITC
[13]	DBN	AUC, Precision	ITC, MIT Reality
[23]	Conv3D	Accuracy, Precision	ITC
[25]	Transformer	AUC, PRAUC	ITC, MIT Reality, Infocom06
[22]	GNN+Attention	AUC, PRAUC	ITC, Infocom06
[12]	GCN	AUC, PRAUC, F1	ITC, Hyccups, MIT Reality
[24]	CNN+Attention	AUC, Precision, F1	ITC, MIT Reality, Infocom06
[27]	2D CNN	Accuracy, Precision, Recall, F1, AUC, PRAUC	ITC, Infocom06
[32]	CL+TGAT	AUC, PRAUC	Infocom05, Hyccups, Infocom06
[26]	GNN+RL	AUC, PRAUC, F1-score	Infocom05, Infocom06
[30]	CNN+Attention	-	-
[11]	Transformer	Accuracy	-
This work	2D CNN	AUC, PRAUC	ITC, MIT Reality, Infocom06

Overall, the literature shows a clear progression from sequential and feature-engineered methods to more advanced deep learning architectures capable of modeling spatiotemporal dynamics. Nevertheless, two main limitations can be

identified. First, many studies evaluate their models on a limited set of datasets, which limits a comprehensive assessment of model generalization. Second, the existing works primarily focus on short-term link prediction, with limited or no exploration of multi-step forecasting and long-term temporal dynamics.

To address these gaps, this work extends the evaluation of the 2D CNN model proposed in [27] by considering more real-world datasets, including the MIT Reality Mining dataset, which was not previously explored. In addition, both single-step and multi-step link prediction are investigated, enabling an analysis of performance degradation over longer prediction horizons. This broader evaluation provides a more complete understanding of the model's robustness and its ability to capture long-term temporal dependencies in opportunistic networks.

4 Link Prediction Methodology and Experiments

The use of appropriate machine learning techniques can yield superior results in link prediction. Accurate link prediction can be used to select the next forwarding node more effectively, enabling informed decisions that enhance delivery rates while reducing delay and overhead in opportunistic networks.

This study adopts the two-dimensional convolutional neural network model proposed by [27], reproducing the original evaluation on the ITC and Infocom06 datasets. In addition, the analysis is extended to the MIT Reality Mining dataset, which is introduced in this study as a novel testbed for link prediction in opportunistic networks. Furthermore, this work investigates link prediction across multiple future matrices, a scenario not addressed in the previously discussed studies.

4.1 Datasets

Both the ITC [21] and Infocom06 [21] datasets contain encounter (connection) logs collected from small Bluetooth-enabled devices known as "iMotes". These datasets were preprocessed into a standardized format in which each record contains the identifiers of the two nodes and the start and end times of their contact.

The ITC dataset recorded, over twelve days, the encounters among fifty iMotes on the campus of the University of Cambridge. The deployment included both mobile devices carried by students and fixed devices installed in locations such as bars, markets, shopping centers, and the reception area of the computing laboratory. The resulting dataset, in the standardized format described above, contains 10,776 contact records.

The Infocom06 dataset recorded, over a period of four days, the movement of 98 portable devices deployed during the IEEE Infocom 2006 Conference in Barcelona. The dataset includes seventy-eight mobile iMotes, with an estimated communication range of approximately thirty meters, and twenty stationary iMotes, with a range of approximately one hundred meters. In the standardized format described previously, the dataset contains 227,657 contact records.

The MIT Reality Mining [7] dataset contains data collected at the Massachusetts Institute of Technology using ninety-eight Nokia 6600 smartphones distributed to students from the MIT Media Laboratory and the MIT Sloan School of Management. Data were collected between July 27, 2004, and May 5, 2005, and the provided file corresponds to a database dump containing anonymized data. The database comprises multiple tables, each corresponding to a specific type of collected information. By querying the database, a text file was generated in the same format as the other two datasets. The resulting file contains 145,172 encounter records in the previously described format. In this work, a ten-day subset of the dataset was extracted to ensure a temporal length comparable to that of the other two datasets.

4.2 Single Matrix Prediction

For the conducted experiments, the parameter settings reported by [25] were adopted. Specifically, the ITC and MIT Reality datasets were partitioned into 300-s intervals to construct the adjacency matrices, whereas the Infocom06 dataset was partitioned into 270-s intervals. All experiments were conducted using an input sequence of 10 adjacency matrices to predict a single future matrix as output. Using the predefined notation, given the input adjacency matrices, $A^{t:t+9} = \{A^t, A^{t+1}, \ldots, A^{t+9}\}$, the model predicts the subsequent matrix A^{t+10}.

The Infocom06 and the MIT datasets were partitioned according to time into training, validation, and test sets using an 80%–10%–10% split, respectively. Similarly, the ITC dataset was divided using a 70%–15%–15% split for training, validation, and testing, respectively.

As described by [27], the employed 2D CNN architecture consists of three intermediate Conv2d layers, each followed by a dropout layer to mitigate overfitting by randomly deactivating a fraction of the neurons. Circular padding was adopted, a technique originally designed for omnidirectional images, which reduces the information loss that may occur with conventional padding strategies such as "zero" or "same". Using circular padding on the horizontal axis, for example, the left border of the image is filled with the rightmost region of the image, and vice versa [29]. In this way, this technique uses information from opposite edges and ensures that the padding kernels have the same behavior inside and at the edges of the images [20]. In the convolutional layers, a kernel size of 3 and a stride size of 1 were used. The model was trained using the binary cross-entropy loss function (BCEWithLogitsLoss) and optimized with the Adam optimizer, with a learning rate of 0.001.

In this study, the model was trained using an early stopping strategy with a patience of 10 epochs. Training was terminated when the validation loss did not improve for 10 consecutive epochs, thereby helping to mitigate overfitting. The construction of adjacency matrices was refined at the boundaries of the training, validation, and test intervals to provide a more accurate representation of the network dynamics. Additionally, the network was represented using only the upper triangular portion of the matrices, under the assumption that encounters are symmetric; that is, when node A encounters node B, node B is also considered

to encounter node A. The experimental results obtained using the 2D CNN model are presented in Table 2.

Table 2. Results of Link Prediction Using the 2D CNN Model

	ITC	Infocom06	MIT
AUC	0.9929	0.9833	0.9812
PRAUC	0.8777	0.8321	0.8921

Results reported in related studies using the ITC, Infocom06, and MIT datasets, as previously described, are presented in Table 3 alongside the results obtained in this study. Cells left empty indicate metrics that were not reported in the corresponding works. For the ITC dataset, the 2D CNN model achieved AUC values higher than those reported by [4,12,13,22,24,25,27,31]. In addition, it achieved PRAUC values exceeding those reported in [12,27]. For the Infocom06 dataset, the model obtained AUC values higher than all compared studies [22,24–27,32], as well as PRAUC values higher than those reported in [22,24,25,27]. Finally, Table 3 shows that, for the MIT Reality Mining dataset, the model achieved higher AUC and PRAUC values compared to related studies [12,13,24,25].

The results presented in Table 3 indicate that the 2D CNN model consistently outperforms those reported in several related studies, demonstrating strong predictive performance across all considered datasets, particularly on the MIT Reality dataset. In cases where other approaches surpass the proposed model, the performance differences are generally marginal. A notable exception is the PRAUC reported by [32], which is considerably higher than that achieved by the 2D CNN on the Infocom06 dataset.

However, it is important to note that the approach in [32] relies on contrastive learning combined with a temporal graph attention network, a substantially more complex and computationally demanding architecture, especially during training. This increased cost originates from the integration of temporal attention mechanisms and contrastive objectives, which require additional sampling, multiple forward passes, and pairwise comparisons between embeddings.

Overall, these findings indicate that two-dimensional convolutional neural networks constitute a robust and promising approach for link prediction in opportunistic networks, offering competitive predictive performance while maintaining significantly lower computational complexity.

4.3 Multiple Matrices' Predictions

Link prediction in opportunistic networks seeks to take advantage of predicted connectivity information to support packet forwarding decisions within routing protocols. Given that each adjacency matrix represents a relatively short time interval (270 s or 300 s), a single predicted matrix may not provide sufficient

Table 3. Link Prediction Results

Ref.	ITC		Infocom06		MIT	
	AUC	PRAUC	AUC	PRAUC	AUC	PRAUC
[4]	0.9780	-	-	-	-	-
[15]	-	-	-	-	0.9850	-
[31]	0.9800	-	-	-	-	-
[13]	0.9591	-	-	-	0.9789	-
[25]	0.9708	0.8915	0.9580	0.7240	0.9738	0.8717
[22]	0.9576	0.8907	0.9493	0.7366	-	-
[12]	0.9681	0.8653	-	-	0.9434	0.8570
[24]	0.9644	-	0.9105	-	0.9789	-
[27]	0.9840	0.7893	0.9831	0.7590	-	-
[32]	-	-	0.9650	0.9620	-	-
[26]	-	-	0.8611	0.8507	-	-
This work	**0.9929**	0.8777	**0.9833**	0.8321	0.9812	**0.8921**

information to effectively support forwarding decisions. To enhance the utility of predictions, it is therefore desirable to forecast links over extended time horizons, thereby offering a more comprehensive view of network evolution. In this context, predicting multiple consecutive adjacency matrices becomes essential for capturing network dynamics over longer periods.

To evaluate the model's ability to predict multiple future matrices, experiments were conducted using a subset of the MIT Reality Mining dataset. This dataset was selected due to its substantially longer duration compared to the ITC and Infocom06 datasets, making it particularly suitable for assessing forwarding strategies in opportunistic networks. In these experiments, the model was provided with an input sequence of ten adjacency matrices and tasked with predicting ten consecutive future matrices, enabling the evaluation of its performance over extended time intervals. Using the adopted notation, given the input sequence, $A^{t:t+9} = \{A^t, A^{t+1}, \ldots, A^{t+9}\}$, the model predicts the subsequent sequence $A^{t+10:t+19} = \{A^{t+10}, A^{t+11}, \ldots, A^{t+19}\}$.

The mean AUC and PRAUC values obtained during the test phase with the 2D CNN model are presented in Table 4. Both metrics exhibit a gradual decline as predictions extend further into the future, reflecting the increasing difficulty of forecasting links over longer time horizons. The decline in PRAUC is more pronounced than that observed for AUC, which can be attributed to the severe class imbalance present in the dataset. While the baseline for AUC is 0.5, corresponding to the expected performance of a random classifier [8], the baseline for PRAUC is determined by the prevalence of the positive class [19], which in this case is approximately 0.0014. Consequently, PRAUC provides a more informative assessment of the model's ability to correctly predict rare positive links in highly imbalanced datasets.

Table 4. MIT Dataset Link Prediction for Multiple Future Matrices

Prediction Window	Time (s)	AUC	PRAUC
1	0—300	0.9781	0.8822
2	300—600	0.9542	0.7682
3	600—900	0.9418	0.6702
4	900—1200	0.9211	0.5946
5	1200—1500	0.9064	0.5351
6	1500—1800	0.8895	0.4861
7	1800—2100	0.8752	0.4362
8	2100—2400	0.8564	0.3937
9	2400—2700	0.8537	0.3545
10	2700—3000	0.8503	0.3251

5 Discussion

This work addresses the problem of link prediction in opportunistic networks, focusing on the development of techniques to predict future links to improve packet forwarding while preserving network resources. The adopted approach represents network dynamics by partitioning contact records into discrete time intervals and modeling them as adjacency matrices.

This study has two main objectives. The first is to evaluate the 2D CNN model proposed by [27] on three real-world opportunistic network datasets by predicting the next adjacency matrix from a sequence of previous matrices. The second is to assess the model's performance in a longer-term prediction scenario by forecasting the next ten consecutive future matrices from an input sequence of ten matrices using the MIT Reality Mining dataset.

The results indicate that the 2D CNN model achieved superior performance on the Infocom06 and MIT Reality Mining datasets, with AUC values of 0.9823 and 0.9812, and PRAUC values of 0.8321 and 0.8921, respectively. Furthermore, multi-step prediction results show a gradual decline in performance as the prediction horizon increases. Nevertheless, even for the tenth predicted matrix, results remain significantly above the baseline values of the evaluation metrics. Specifically, the AUC of 0.8503 is 170% higher than the baseline of 0.5, and the PRAUC of 0.3251 is 23,221% higher than the baseline of 0.0014.

These findings suggest that long-term link prediction using the 2D CNN model can provide meaningful insights into future connectivity patterns in opportunistic networks. This capability may support more effective forwarding decisions in routing protocols, potentially improving network efficiency.

6 Conclusions

This paper evaluated a 2D CNN-based approach for link prediction in opportunistic networks using real-world datasets. The results demonstrate that the

model effectively predicts future network connections, achieving strong performance across different scenarios and maintaining reasonable accuracy even for longer-term predictions. Overall, the findings indicate that deep learning-based approaches, particularly those employing temporal patterns in the sequence of adjacency matrices, represent a promising direction for improving link prediction and supporting routing strategies in opportunistic networks.

As directions for future work, several research avenues can be explored to further improve the proposed approach. First, conducting sensitivity analyses of the 2D CNN architecture with respect to key parameters is recommended. These parameters include the temporal window size used in generating adjacency matrices from the dataset records, as well as the length of the input matrix sequences used for predictions. Such analyses can provide deeper insight into how these factors influence model performance and help guide the selection of more effective configurations, ultimately improving the overall approach.

Additionally, it is important to investigate how adapting the current model to more advanced deep learning architectures, such as graph neural networks and temporal graph networks, would affect link prediction performance and the associated training cost. Another promising direction is the exploration of online or continual learning strategies, enabling the model to dynamically adapt to evolving network conditions over time.

Furthermore, it is important to investigate the use of multi-step future adjacency matrix predictions in message forwarding protocols. This approach aims to reduce the number of message copies by enabling more informed selection of the next forwarding node. The proposed protocol could be evaluated in simulation environments using the MIT Reality Mining dataset, allowing its effectiveness to be assessed under realistic network conditions.

Acknowledgment. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Finance Code 001.

References

1. Altaweel, A., Aslam, S., Kamel, I.: Security attacks in opportunistic mobile networks: a systematic literature review. J. Netw. Comput. Appl. **221**, 103782 (2024). https://doi.org/10.1016/j.jnca.2023.103782
2. Altaweel, A., Stoleru, R., Gu, G., Maity, A.K.: CollusiveHijack: a new route hijacking attack and countermeasures in opportunistic networks. In: 2019 IEEE Conference on Communications and Network Security (CNS), pp. 73–81. IEEE (2019). https://doi.org/10.1109/CNS.2019.8802699
3. Arrar, D., Kamel, N., Lakhfif, A.: A comprehensive survey of link prediction methods. J. Supercomput. **80**(3), 3902–3942 (2024). https://doi.org/10.1007/s11227-023-05591-8
4. Cai, X., Shu, J., Al-Kali, M.: Link prediction approach for opportunistic networks based on recurrent neural network. IEEE Access **7**, 2017–2025 (2018). https://doi.org/10.1109/ACCESS.2018.2886360

5. Chen, D., Borrego, C., Navarro-Arribas, G.: A privacy-preserving routing protocol using mix networks in opportunistic networks. Electronics **9**(11), 1754 (2020). https://doi.org/10.3390/electronics9111754
6. Chen, J., Wang, X., Xu, X.: GC-LSTM: graph convolution embedded LSTM for dynamic network link prediction. Appl. Intell., 1–16 (2021). https://doi.org/10.1007/s10489-021-02518-9
7. Eagle, N., Pentland, A.S.: CRAWDAD mit/reality (2022). https://doi.org/10.15783/C71S31
8. Fawcett, T.: An introduction to roc analysis. Pattern Recogn. Lett. **27**(8), 861–874 (2006). https://doi.org/10.1016/j.patrec.2005.10.010
9. Fu, X., Yao, H., Postolache, O., Yang, Y.: Message forwarding for WSN-assisted opportunistic network in disaster scenarios. J. Netw. Comput. Appl. **137**, 11–24 (2019). https://doi.org/10.1016/j.jnca.2019.04.005
10. Garg, P., Dixit, A., Sethi, P.: ML-fresh: novel routing protocol in opportunistic networks using machine learning. Comput. Syst. Sci. Eng. **40** (2022). https://doi.org/10.32604/CSSE.2022.019557
11. Gong, Y., Liu, R.: Opportunity network link prediction based on graph-BERT. In: Third International Conference on Informatics, Networking, and Computing, ICINC 2024, vol. 13637, pp. 194–203. SPIE (2025). https://doi.org/10.1117/12.3056957
12. Liao, Y., Shu, J., Liu, L.: Dynamic networks link prediction based on continuous gated recurrent graph convolution. Int. J. Mach. Learn. Cybern., 1–17 (2024). https://doi.org/10.1007/s13042-024-02413-y
13. Liao, Z., Liu, L., Chen, Y.: A novel link prediction method for opportunistic networks based on random walk and a deep belief network. IEEE Access **8**, 16236–16247 (2020). https://doi.org/10.1109/ACCESS.2020.2967407
14. Liu, K., Chen, Z., Wu, J., Xiao, Y., Zhang, H.: Predict and forward: an efficient routing-delivery scheme based on node profile in opportunistic networks. Fut. Internet **10**(8), 74 (2018). https://doi.org/10.3390/fi10080074
15. Ma, Y., Shu, J.: Opportunistic networks link prediction method based on Bayesian recurrent neural network. IEEE Access **7**, 185786–185795 (2019). https://doi.org/10.1109/ACCESS.2019.2961243
16. Mukhopadhyay, A., Nishchitha, N., Chaya, M.: Utilization of opportunistic networks for transmitting data of medical importance. In: 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 176–181. IEEE (2020). https://doi.org/10.1109/ICIRCA48905.2020.9182828
17. Rui, L., Zhu, Y., Gao, Z., Qiu, X.: CLPM: a cooperative link prediction model for industrial internet of things using partitioned stacked denoising autoencoder. IEEE Trans. Industr. Inf. **17**(5), 3620–3629 (2020). https://doi.org/10.1109/TII.2020.2999318
18. Sachdeva, R., Dev, A.: Review of opportunistic network: Assessing past, present, and future. Int. J. Commun Syst **34**(11), e4860 (2021). https://doi.org/10.1002/dac.4860
19. Saito, T., Rehmsmeier, M.: The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLoS ONE **10**(3), e0118432 (2015). https://doi.org/10.1371/journal.pone.0118432
20. Schubert, S., Neubert, P., Pöschmann, J., Protzel, P.: Circular convolutional neural networks for panoramic images and laser data. In: 2019 IEEE Intelligent Vehicles Symposium (IV), pp. 653–660. IEEE (2019). https://doi.org/10.1109/IVS.2019.8813862

21. Scott, J., Gass, R., Crowcroft, J., Hui, P., Diot, C., Chaintreau, A.: CRAWDAD cambridge/haggle (v. 2009-05-29) (2022). https://doi.org/10.15783/C70011
22. Shi, J., Shu, J.: Continuous-graph attentional neural networks for temporal link prediction. In: 2022 IEEE 22nd International Conference on Communication Technology (ICCT), pp. 1932–1936. IEEE (2022). https://doi.org/10.1109/ICCT56141.2022.10072410
23. Shu, J., Li, J., Zhang, X.: Link prediction based on 3d convolutional neural network. In: 2022 IEEE/CIC International Conference on Communications in China (ICCC), pp. 156–161. IEEE (2022). https://doi.org/10.1109/ICCC55456.2022.9880708
24. Shu, J., Liao, Y., Li, J.: Spatial-temporal attention TCN-based link prediction for opportunistic network. Electronics **13**(5), 957 (2024). https://doi.org/10.3390/electronics13050957
25. Shu, J., Shi, J., Liao, L.: Link prediction model for opportunistic networks based on feature fusion. IEEE Access **10**, 80900–80909 (2022). https://doi.org/10.1109/ACCESS.2022.3192608
26. Sun, Q., Seah, W.K., Xu, G.: EQ-STAR: energy-efficient high-quality routing based on spatio-temporal attention and reinforcement learning. In: 2025 IEEE/ACM 33rd International Symposium on Quality of Service (IWQoS), pp. 1–6. IEEE (2025). https://doi.org/10.1109/IWQOS65803.2025.11143523
27. Thiago, V.S., Salles, R.M., Duarte, J.C., Dias, G.M.S.: Prediçao de enlaces em redes oportunistas utilizando redes neurais profundas. In: Workshop de Testes e Tolerância a Falhas (WTF), pp. 29–42. SBC (2024). https://doi.org/10.5753/wtf.2024.2834
28. Wang, H., Song, L., Zhang, G., Chen, H.: Timetable-aware opportunistic DTN routing for vehicular communications in battlefield environments. Futur. Gener. Comput. Syst. **83**, 95–103 (2018). https://doi.org/10.1016/j.future.2018.01.013
29. Wang, T.H., Huang, H.J., Lin, J.T., Hu, C.W., Zeng, K.H., Sun, M.: Omnidirectional CNN for visual place recognition and navigation. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 2341–2348. IEEE (2018). https://doi.org/10.1109/ICRA.2018.8463173
30. Xie, H., Jiang, S.: Maritime opportunistic network routing strategies for assessing link connectivity based on deep learning. Electronics **14**(6), 1187 (2025). https://doi.org/10.3390/electronics14061187Atten
31. Yemeni, Z., Shu, J., Zhang, X., Liu, L.: A DBN approach to predict the link in opportunistic networks. In: Patnaik, S., Jain, V. (eds.) Recent Developments in Intelligent Computing, Communication and Devices. AISC, vol. 752, pp. 575–587. Springer, Singapore (2019). https://doi.org/10.1007/978-981-10-8944-2_67
32. Zhang, F., Luo, C., Seah, W.K., Xu, G., Zhao, K.: Temporal graph attention and contrastive learning model for link prediction in dynamic networks. Comput. Netw., 111596 (2025). https://doi.org/10.1016/j.comnet.2025.111596
33. Zhang, Y., Li, J., Li, Y., Xu, D., Ahmed, M., Li, Y.: Cellular traffic offloading via link prediction in opportunistic networks. IEEE Access **7**, 39244–39252 (2019). https://doi.org/10.1109/ACCESS.2019.2891642
34. Zhou, H., Wu, T., Zhang, H., Wu, J.: Incentive-driven deep reinforcement learning for content caching and D2D offloading. IEEE J. Sel. Areas Commun. **39**(8), 2445–2460 (2021). https://doi.org/10.1109/JSAC.2021.3087232

Evaluating Accuracy–Performance Trade-Offs in Deep Learning Frameworks for Diabetic Retinopathy Detection

Bruno Morales(✉), Cristiano Künas, Rodrigo Machado, Pedro Rigon, and Philippe O. A. Navaux

Institute of Informatics, Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil
{bmmorales,cakunas,rcmachado,phcrigon,navaux}@inf.ufrgs.br
https://www.inf.ufrgs.br/site/

Abstract. Deep learning has become one of the most important tools in medical image processing, requiring models that meet response-time constraints, accuracy requirements, and memory limitations imposed by hardware limitations. Given our current scenario, this demand is even more important in hospitals and institutions with limited computing resources, where access to high-performance GPUs is not always available for daily use. Our results show that optimization can significantly improve computational efficiency, with throughput gains of up to 288.8% and reductions in training time of up to 70.3%. However, the effect on Area Under the Curve (AUC) depends on the architecture used. Our research supports the selection and optimization of healthcare workloads running on GPUs, aiming to deliver more efficient, accessible, and clinically appropriate solutions.

Keywords: Deep Learning · High Performance Computing · Healthcare

1 Introduction

Artificial intelligence (AI) and deep learning (DL) have advanced very rapidly in a short period of time, enabling increasingly robust and optimized solutions for complex pattern recognition tasks across multiple domains [1]. Especially in healthcare, these advancements have been very significant, as deep learning models based on image classifications continue to improve the accuracy, speed, and scalability of diagnostic support systems. In ophthalmology, automated fundus image analysis has become particularly relevant, with deep learning now playing a central role in diabetic retinopathy (DR) screening pipelines [2].

The significance of such improvements extends beyond technology to healthcare delivery. A shortage of specialists, long waiting lists, and overcrowding are common in public healthcare organizations. With all these constraints in

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 303–320, 2026.
https://doi.org/10.1007/978-3-032-30491-9_20

mind, an automated tool that screens patients to determine whether they need a specialist referral could prove highly effective in improving early diagnosis and reducing the number of patients referred to specialist care.

However, developing deployable solutions based on deep learning algorithms entails more than simply building a powerful prediction model. Modern neural networks require substantial amounts of both RAM (host memory) and VRAM (GPU memory). Both components are expensive and rare. Therefore, optimizing computational power usage is crucial when developing deployable neural network-based prediction models that can be trained on a single GPU.

Figure 1 illustrates representative fundus images without diabetic retinopathy and with diabetic retinopathy, highlighting the visual contrast addressed by the binary classification task considered in this study.

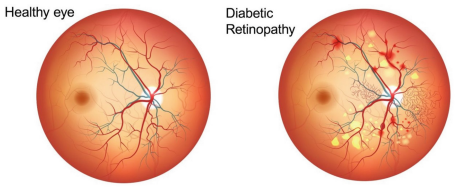


Fig. 1. Illustrative representation of diabetic retinopathy.

Motivated by this current scenario in the evolution of technology, we investigated the automated classification of diabetic retinopathy from fundus images, focusing on the practical trade-offs between predictive performance and training efficiency. We analyze six variants of the same image classification pipeline based on the pre trained InceptionV3 convolutional neural network (CNN) [3]. InceptionV3 combines multi-scale feature extraction with computational efficiency, making it a strong backbone for transfer learning, which has become a standard strategy in DR grading tasks [4]. The model is initialized with weights pre-trained on large-scale datasets such as ImageNet [5], allowing it to adapt effectively to downstream medical imaging tasks.

The evaluated variants include three baseline implementations developed in TensorFlow, PyTorch, and MONAI, as well as three optimized counterparts designed to improve performance under the selected evaluation criteria. Cross-framework comparisons of this kind have already attracted attention in medical imaging research [6]. In the optimized configurations, we explore reduced-precision execution through FP16 mixed precision training and fine-tuning, in which early network layers remain fixed while higher level layers are updated. These strategies are intended to lower computational cost, and improve convergence stability when adapting a pre trained model to a smaller dataset.

The trade-offs will be evaluated by comparing the six pipeline variations based on AUC scores, training time, and throughput. This is done for both ARM and x86 platforms, allowing us to study how efficiency and accuracy are

affected by platform architecture, deep learning framework, and floating-point precision in the context of real-world AI use in DR classification. We want to demonstrate how choices made during the design of a training workflow can affect both AI efficiency and usability intended to support specialists in diabetic retinopathy screening.

The remainder of this paper is organized as follows. Section 2 presents the background on PyTorch, TensorFlow/Keras, and MONAI. Section 3 reviews related work on diabetic retinopathy detection and on efficiency-oriented optimization strategies for deep learning pipelines. Section 4 describes the dataset, hardware platforms, model architecture, training procedure, and optimization techniques adopted in the experiments. Section 5 presents and discusses the results in terms of AUC, throughput, training time and memory. Finally, Sect. 6 concludes the paper and outlines directions for future work.

2 Background

We will analyze frameworks such as PyTorch, TensorFlow/Keras, and MONAI, as they are relevant to different aspects of our research. PyTorch is an advanced deep learning platform that supports dynamic computational graphs and emphasizes Python. In turn, TensorFlow/Keras will be an excellent benchmark due to its mature ecosystem of solutions with scalable graph-based execution. Furthermore, MONAI has a particular focus on medical imaging solutions. Thus, using these frameworks, it will be possible to evaluate both general and healthcare-specific deep learning solutions to obtain a comparison across multiple isolated scenarios.

2.1 Pytorch

PyTorch by Meta AI (2016) facilitates neural network programming using idiomatic Python and efficient use of hardware accelerators [7]. The dynamic execution paradigm of PyTorch enables quick debugging and inspection of tensors, their shapes, values, and gradients. In distributed training, PyTorch leverages distributed tooling such as DistributedDataParallel (DDP) for gradient synchronization [8]. The interplay between autograd and continual optimizations, in terms of runtime and accelerator support, makes PyTorch effective for distributed and interactive training.

2.2 TensorFlow and Keras

TensorFlow (Google Brain, 2015) [9] supports scalable execution via computation graphs and enables efficient distributed training [10]. TensorFlow is available in both graph mode and eager execution mode. Further, TensorFlow can compile and optimize intensive computations using XLA [11].

2.3 MONAI

Medical Open Network for AI (MONAI) is an open-source framework on top of PyTorch for developing applications in medical imaging [12]. It provides a wide range of domain-specific tools, such as transforms, architectures, and losses, specifically tuned for volumetric data and inference.

3 Related Work

Research on diabetic retinopathy (DR) has changed considerably over the last few years. Earlier studies often relied on handcrafted lesion descriptors and multi-stage pipelines. In contrast, more recent work has increasingly adopted end-to-end deep learning models for classification and severity grading from fundus images. Reviews of the field consistently document this shift, particularly in studies trained and evaluated on public datasets such as EyePACS, APTOS, Messidor, IDRiD, and FGADR [13,14]. Beyond reporting accuracy on a single dataset, some studies have also highlighted transferability across settings. In this context, Zhou *et al.* [15] introduced FGADR as a benchmark for lesion segmentation, grading, and transfer learning, thereby broadening the discussion on model generalization.

Among the strategies used in DR screening, transfer learning remains one of the most practical. This is primarily due to retinal datasets, which are generally small, class unbalanced, and collected under heterogeneous imaging conditions. Prior work has shown that pretrained convolutional backbones can achieve competitive results for DR diagnosis and grading [16]. Related transfer-learning strategies have also been successfully applied in smartphone based retinal imaging settings [17]. More recent studies have explored alternatives that go beyond conventional CNN pretraining. Alam *et al.* [18] reported gains from contrastive pretraining, particularly in terms of representation quality and cross-institution transfer. Goh *et al.* [19] showed that Vision Transformers are strong candidates for referable DR detection. There is also evidence from prospective screening settings that model performance should be evaluated alongside robustness and deployment conditions, rather than solely through aggregate test metrics [20].

Another relevant body of work concerns efficiency during training and inference. In this literature, the central question is not only whether a model is accurate, but also how efficiently it can be trained and executed. Mixed precision computation is one of the best known examples, and prior studies have shown that it can improve the runtime efficiency of deep neural networks [21]. However, these benefits are not always automatic. He *et al.* [22] showed that precision choices should account for casting overhead, since lower precision does not necessarily yield better end-to-end performance. Similar observations appear in work on tensor layout, where choices such as NCHW and NHWC can substantially affect throughput [23]. Systems-oriented studies have likewise shown that the input pipeline itself can become a bottleneck, limiting GPU utilization when data loading and preprocessing are not carefully optimized [24,25].

Although both lines of research are well established, they have been less frequently examined together in the DR literature. Ophthalmic studies typically emphasize predictive metrics such as AUC, sensitivity, specificity, and interpretability, whereas systems papers usually evaluate performance on more generic vision workloads. Consequently, evidence remains limited on how framework-level and optimization choices behave under the same retinal classification task. This issue is also relevant for healthcare-oriented ecosystems such as MONAI, a PyTorch-based framework designed for medical imaging [12]. However, comparable evidence across TensorFlow, native PyTorch, and MONAI-based pipelines under equivalent experimental conditions remains scarce. Motivated by this gap, our study compares accuracy, throughput, training time, for the same DR classification pipeline implemented in TensorFlow, PyTorch, and MONAI, under both baseline and optimized settings.

4 Methodology

This section describes the methodology adopted in our experiments, including the dataset, architecture, training procedure, and optimization strategies. TensorFlow was used as the baseline framework, since it corresponds to the original implementation. To make the experiments comparable across frameworks, the TFRecord (TfRec) files were converted to IDX format. In addition, all images were prepared at a resolution of 299×299 pixels, which is the expected input size for InceptionV3.

4.1 Dataset

For the experiments reported in this article, we used 15,816 fundus images from a Brazilian diabetic retinopathy dataset, with labels binarized into a two-class setting (0/1). The dataset is imbalanced, with 11,625 samples in class 0 and 4,191 in class 1. Approximately 70/30 train-test split was adopted, yielding 11,000 training images and 4,816 test images. Details on data sources, acquisition, inclusion criteria, and labeling protocol are provided in [26].

4.2 Hardware Platforms

The machines used in the experiment were 1xNVIDIA L40S with 18,176 CUDA cores and 48 GB of global memory, and 1xNVIDIA GeForce RTX 4090 with 16,384 CUDA cores and 24 GB of global memory available in the LPPD infrastructure[1] at INF/UFRGS. Hydra node at the Lyon site of Grid'5000[2] [27], equipped with an 1xNVIDIA GH200 with 16,896 CUDA cores and 96 GB of global memory.

[1] http://gppd-hpc.inf.ufrgs.br.
[2] https://www.grid5000.fr.

4.3 Training and Optimization

Table 1 summarizes the software stack and container versions used for the TensorFlow, PyTorch, and MONAI experiments.

Table 1. Version comparison across TensorFlow, PyTorch, and MONAI.

Component	TensorFlow	PyTorch	MONAI
TensorFlow	2.17.0	—	—
PyTorch	—	2.5.0	2.5.0
MONAI	—	2.5.0	1.3.2
CUDA	12.6.3	12.6.3	12.6.3
cuDNN	9.6.0.74	9.6.0.74	9.6.0.74
DALI	1.44	1.44	1.44
Python	3.12	3.12	3.12

To reduce variations in the training process, each configuration was executed 10 times. We refer to the unoptimized runs as Base and to the optimized runs as Opt. For each framework and GPU, the corresponding Base and Opt runs used the same random seeds, so paired tests were applied to the run-wise AUC differences. Each run used 200 epochs and a batch size of 96 [29], chosen as a compromise between stochastic gradient stability and computational efficiency. Figure 2 illustrates the experimental workflow chosen to compare the six training pipelines. All implementations are built on the same InceptionV3 architecture, which provides a common architectural foundation for the analysis. From there, the workflow is divided into reference and optimized versions. The reference branch corresponds to the original training configuration in each framework without any optimization, while the optimized branch includes the strategies implemented to improve performance. Despite these differences, all variants use the same training settings: 10 runs, 200 epochs, and a batch size of 96. This organization helps clarify which elements remain fixed between experiments and which are modified in the optimized configurations.

We report the mean and standard deviation of all metrics across these repetitions. Each variant uses the InceptionV3 architecture as its convolutional backbone, initialized with pre-trained weights on ImageNet. We separated our six approaches into clean, unoptimized versions (BASE) and optimized versions (OPT). The base versions follow a straightforward training procedure using each framework's standard data pipeline and, in most cases, the default precision and scheduling settings. The three optimized variants TF-Opt, PT-Opt, and MN-Opt—incorporate a shared set of performance and convergence techniques, while also making architecture-specific choices such as tensor memory layout and compiler backend.

During training, we also investigated how the memory organization behavior of the input tensors affects model execution. We evaluated two tensor layouts

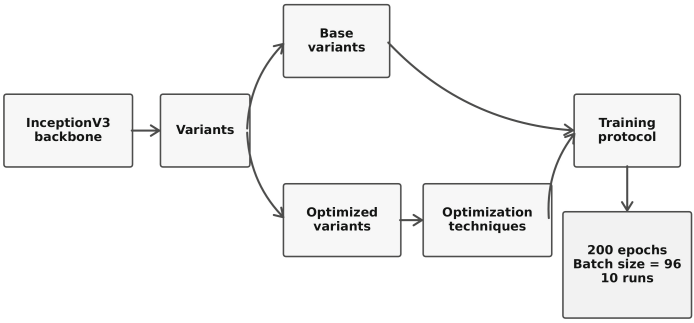


Fig. 2. Workflow of the steps.

supported by the frameworks in isolation, since the order of dimensions influences how data is stored and accessed during convolution operations. TensorFlow uses the NHWC tensor layout, which denotes the dimension order [batch, height, width, channels], whereas PyTorch was evaluated with the NCHW tensor layout, which denotes the order [batch, channels, height, width] [28]. Although both layouts represent the same input data, the different order of dimensions can affect memory access patterns, convolution performance, and overall training scalability.

To assess the practical impact of these layouts, we present the AUC and training time for each configuration discussed, allowing a direct comparison of predictive performance and computational efficiency.

We also addressed the use of FP16 in optimized versions, seeking to reduce the memory consumption of the computing cores [21]. The benefits of mixed precision often depend on adjustments that take into account the costs. For the MONAI variant, we used specialized mixed-precision toolkits for the optimization phase, avoiding loss of precision.

Four optimization techniques are consistently applied in TF-Opt, PT-Opt, and MN-Opt. Each of these optimizations addressed a different aspect of the training process and together, they formed the optimization base upon which specific structure improvements are added.

4.4 GPU-Accelerated Data Loading via NVIDIA DALI

In the basic variants where we have no optimizations, image decoding and preprocessing are managed by each framework's standard CPU based data pipeline. When the CPU cannot prepare batches as quickly as the GPU can consume them, the GPU becomes idle, causing a problem known as pipeline depletion, and training performance suffers the consequences. All three optimized variants replace the CPU side pipeline with NVIDIA DALI, which performs data decoding, resizing, and augmentation directly on the GPU. By moving these operations to the accelerator and overlapping data preparation with model computation, DALI reduces the transfer bottleneck between the host and the device and allows

the GPU to maintain higher utilization throughout training. The practical result is a significant increase in training performance, especially on GPUs with high computational capacity that would otherwise be underutilized while waiting for data, negatively impacting the total training time.

4.5 Mixed-Precision Training (FP16)

All optimized variants use AMP-based FP16 mixed-precision training [21], so that most forward and backward computations are performed in 16-bit floating-point. This helps reduce memory usage and enables matrix multiplications to leverage the GPU's Tensor Cores, which are optimized for reduced-precision arithmetic and can deliver higher throughput than standard FP32 units. In cost-aware optimization settings [22], these memory savings may also make it easier to train on GPUs with limited VRAM or to use larger effective batch sizes on the same hardware.

4.6 Cosine Annealing Learning Rate Schedule with Warmup

Rather than maintaining a constant learning rate or using step-wise decay, each optimized variant uses a cosine annealing schedule with a linear warmup phase. During warmup, the learning rate increases from a near-zero value to its peak, preventing the large, potentially destabilizing gradients that commonly arise in the first few epochs particularly when adapting a pre-trained backbone to a new classification task. After warmup, the cosine schedule smoothly reduces the learning rate toward a specified minimum, encouraging the optimizer to settle into flatter regions of the loss landscape. The specific number of warmup epochs varies slightly across frameworks. Still, the underlying strategy remains identical and provides a more stable, effective training trajectory than a flat learning rate.

4.7 Exponential Moving Average (EMA) of Model Weights

Each optimized variant maintains an exponential moving average (EMA) of the model parameters with a decay factor of 0.999. Rather than evaluating the model at its current parameter values, which may fluctuate due to stochastic gradient noise, EMA produces a smoothed version of the weights that reflects the running average of the optimization trajectory. This smoothed model is used at evaluation time and tends to generalize better than the instantaneous parameters, particularly in the later stages of training when the loss landscape becomes flatter. The optimizer oscillates around a local minimum.

4.8 Ablation Protocol

To isolate the contribution of each optimization used in the optimized pipeline, we performed a component-wise ablation study. Starting from the baseline configuration, we enabled one optimization at a time DALI AMP/FP16, cosine

learning-rate scheduling, and EMA—and compared these variants against the full optimized configuration. All other factors, including model architecture, dataset split, batch size, number of epochs, and evaluation protocol, were kept fixed. This design allows the effect of each component to be assessed relative to the same baseline, while also showing the overall behavior of the complete optimized stack.

4.9 Statistical Analysis

To assess whether optimization significantly changed predictive performance, we compared the run-level AUC values of the Base and Opt variants within each framework and GPU. Because Base and Opt runs were matched by run index under the same seed schedule, paired tests were applied to the run-wise AUC differences. Normality of these paired differences was evaluated using the Shapiro–Wilk test [30]. When normality was not rejected, a paired t-test was used; otherwise, a Wilcoxon signed-rank test [31] was applied. A significance level of $\alpha = 0.05$ was adopted for all comparisons.

5 Results

This section presents the quantitative evaluation of six training pipelines: TensorFlow Base (TF-Base), TensorFlow Optimized (TF-Opt), PyTorch Base (PT-Base), PyTorch Optimized (PT-Opt), MONAI Base (MN-Base), and MONAI Optimized (MN-Opt). Each experiment was executed on a single GPU. This design choice has an important implication: the inter-GPU communication fabric (e.g., NVLink, InfiniBand, or PCIe switches) remains entirely inactive. Therefore, any performance differences observed across the three platforms can be attributed exclusively to on-chip resources such as memory bandwidth, Tensor Core availability, and sustained clock frequency rather than to interconnect or topology effects.

5.1 Model Quality: AUC

AUC was used as the main quality metric because it shows how well the model separates the two classes. Since it does not depend on a single decision threshold, it is useful for comparing the effect of different pipelines on classification performance. Figure 3 summarizes the mean AUC obtained by each framework variant on each GPU platform.

The figure shows that changing the GPU had little effect on AUC as long as the training pipeline remained the same. Differences in discriminative performance were driven primarily by the software configuration rather than by the hardware itself.

The statistical analysis of the Base–Opt AUC differences confirmed that the effect of optimization was framework-dependent across all three GPUs. On the GH200, PyTorch showed a significant increase in AUC (0.9186 → 0.9855,

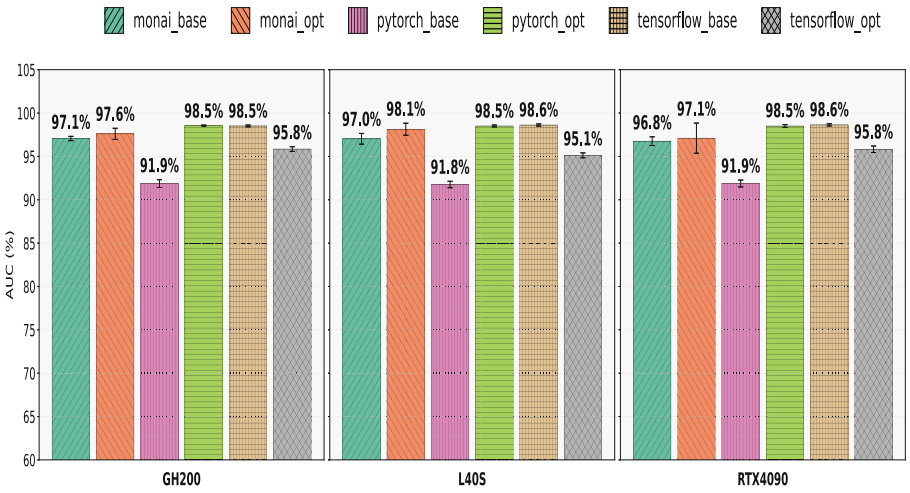


Fig. 3. Mean AUC per framework variant on each GPU platform.

$p = 5.32 \times 10^{-12}$), whereas TensorFlow showed a significant decrease (0.9851 → 0.9585, $p = 1.50 \times 10^{-10}$). In contrast, the MONAI difference on this GPU did not reach statistical significance (0.9706 → 0.9760, $p = 0.0588$).

On the L40S, all three frameworks showed significant Base–Opt differences: PyTorch ($p = 2.96 \times 10^{-12}$), TensorFlow ($p = 1.73 \times 10^{-10}$), and MONAI ($p = 0.00199$). On the RTX 4090, PyTorch ($p = 1.01 \times 10^{-12}$) and TensorFlow ($p = 1.04 \times 10^{-8}$) also remained significant, whereas MONAI again showed no significant difference (Wilcoxon, $p = 0.359$). Overall, these results indicate that optimization consistently altered AUC in PyTorch and TensorFlow, but produced smaller and less consistent effects in MONAI.

For TensorFlow, the baseline setting produced the best overall AUC values, with TF-Base remaining close to 98.5%–98.6% across all three GPUs. After optimization, performance declined on all platforms, reaching 95.8% on the GH200, 95.1% on the L40S, and 95.8% on the RTX 4090. Although the optimized version improved computational efficiency, this came at a clear cost in predictive performance.

PyTorch showed the opposite pattern. Its baseline results were the weakest in the study, with PT-Base ranging from 91.8% to 91.9%, but optimization substantially improved performance. PT-Opt reached about 98.5% across all three GPUs, representing the largest AUC gain observed in this study. This improvement was reproduced consistently across different hardware platforms.

MONAI occupied an intermediate position between these two cases. The baseline pipeline produced AUC values ranging from 96.8% to 97.1%, and optimization yielded smaller gains than in PyTorch, although still positive in most settings. MN-Opt reached 97.6% on the GH200 and 98.1% on the L40S, while the RTX 4090 remained at 97.1%. Overall, these results suggest that optimization

did not affect all frameworks equally: it reduced AUC in TensorFlow, strongly improved it in PyTorch, and produced more limited gains in MONAI.

5.2 Throughput

The throughput was included to show how many images each pipeline can process per second. This is important because it reflects how efficiently each framework uses the available GPU resources during training and helps in the final analysis of how each approach utilized the available resources in each process. Figure 4 summarizes the mean training throughput achieved by each framework variant on the GH200, L40S, and RTX 4090.

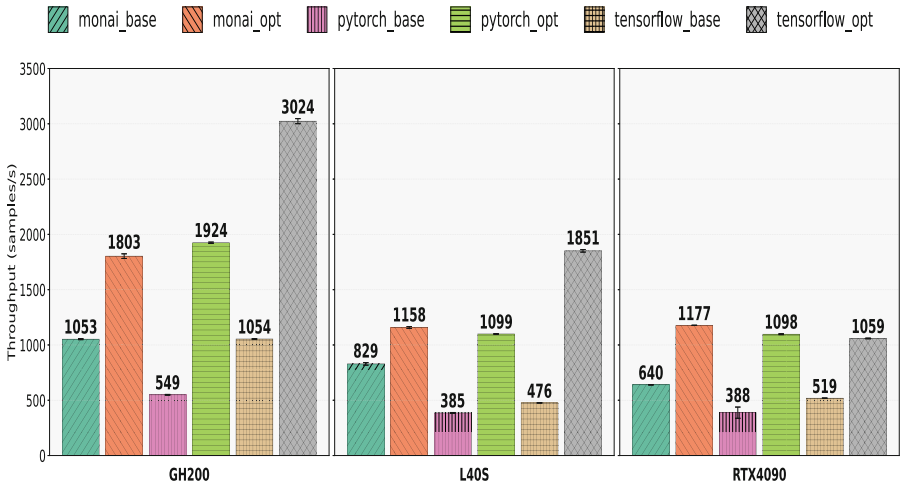


Fig. 4. Average training throughput (img/s) per framework variant on each selected GPU.

The figure shows that across all three frameworks, the optimized pipelines consistently achieved higher throughput than their respective baselines. However, because each optimized variant combines multiple modifications introduced in Sect. 4, these gains should be interpreted as the result of the full optimized pipeline rather than as the isolated effect of any single optimization.

TensorFlow delivered the highest absolute throughput values in this study. On the GH200, TF-Opt increased throughput from 1,054 to 3,024 img/s, yielding a 186.9% gain. On the L40S, throughput increased from 476 to 1,851 img/s, a 288.8% gain. On the RTX 4090, throughput increased from 519 to 1,059 img/s, corresponding to a 104.1% increase. Under the optimized TensorFlow configuration, the GH200 achieved 1.63× the throughput of the L40S and 2.86× that of the RTX 4090. These results indicate that the optimized TensorFlow pipeline

benefited more strongly from the GH200 and L40S platforms than from the RTX 4090.

PyTorch also improved throughput on all three platforms. On the GH200, PT-Opt increased throughput from 549 to 1,924 img/s, yielding a 250.4% gain. On the L40S, throughput increased from 385 to 1,099 img/s, a 185.3% gain. On the RTX 4090, throughput increased from 388 to 1,098 img/s, a 183.1% increase. Notably, the optimized PyTorch results on the L40S and RTX 4090 were nearly identical, differing by only about 1 img/s. In this case, the optimized PyTorch pipeline produced very similar throughput on these two GPUs despite their different hardware classes.

MONAI-Opt also improved throughput on all platforms. On the GH200, throughput increased from 1,053 to 1,803 img/s, a 71.3% gain. On the L40S GPU, throughput increased from 829 to 1,158 img/s, a 39.8% gain. On the RTX 4090, throughput increased from 640 to approximately 1,178 img/s, corresponding to an 84.1% increase. Although these relative gains were smaller than those observed for TensorFlow and PyTorch on some GPUs, MONAI-Opt remained competitive and achieved the highest optimized throughput on the RTX 4090 among the three frameworks.

In the final analysis, these results show that the interaction between the software stack and the hardware platform strongly influenced throughput. TF-Opt achieved the highest throughput on the GH200 at 3,024 img/s and on the L40S at 1,851 img/s. On the RTX 4090, however, the three optimized pipelines were much closer, ranging from approximately 1,059 to 1,178 img/s, with MONAI-Opt holding a slight lead. Finally, the throughput results suggest that optimization substantially improved training efficiency, but that the magnitude of this improvement depended on both the framework implementation and the GPU used.

5.3 Train Time

Training time was considered because it shows the total wall-clock time needed to complete training. This metric is useful for evaluating the practical cost of each pipeline. Figure 5 reports the mean wall-clock training time, in seconds, for the full 200-epoch run.

PT-Base is the slowest configuration on every platform, exceeding 5 000 s even on the GH200. The DALI + AMP + `torch.compile` stack of PT-Opt cuts training time by −70.3 % on the GH200 (5 007 → 1 488 s), −61.5 % on the L40S (6 836 → 2 629 s), and −60.6 % on the RTX 4090 (7 032 → 2 771 s). The consistent 60–70 % reduction across all three platforms indicates that PT-Base's bottleneck resides predominantly in the Python DataLoader and CPU-side TFRecord decoding: once DALI moves decoding to the GPU and `torch.compile` eliminates per-step Python dispatch overhead, and the GPU is kept saturated regardless of the underlying architecture. The additional 10 % gain observed on the GH200 (70 % vs. 60 %) is attributable to higher memory bandwidth allowing DALI's prefetch buffer to drain and refill faster.

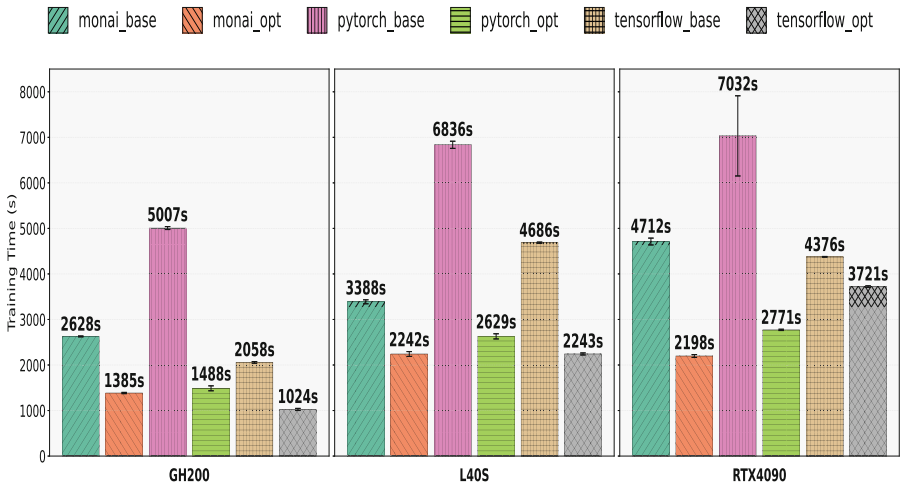


Fig. 5. Mean training time (seconds) per framework.

TF-Opt reduces training time by −50.2 % on the GH200 (2 058 → 1 024 s) and −52.1 % on the L40S (4 686 → 2 243 s), but only −15.0 % on the RTX 4090 (4 376 → 3 721 s). This asymmetry mirrors the throughput divergence discussed in Sect. 5.2: the XLA-compiled DALI pipeline does not extract comparable latency improvements from the consumer GPU architecture. While the L40S benefits from XLA kernel fusion with its larger shared-memory banks, RTX 4090's TDP management and scheduler constraints limit sustained utilization, preventing the same wall-time compression. Consequently, TF-Opt on the RTX 4090 still requires over one hour of training, whereas PT-Opt and MN-Opt finish in under 46 min on the same hardware.

In the MONAI-Opt approach, training time is reduced by −47.3 % on the GH200 (2 628 → 1 385 s), −33.8 % on the L40S (3 388 → 2 242 s), and −53.4 % on the RTX 4090 (4 712 → 2 198 s). The more moderate gains relative to PT-Opt and TF-Opt reflects MONAI modular design, where DALI integration is optional, and not all transforms are offloaded to the GPU by default. It is noteworthy that MN-Opt achieves the *fastest absolute time* on the RTX 4090 (2 198 s), surpassing even PT-Opt (2 771 s), because the cosine scheduler and warmup allow convergence with fewer wasted gradient steps, and the `channels_last` layout interacts favorably with the Ada Lovelace memory subsystem.

5.4 Memory

Table 2 reports the mean peak VRAM usage for the Base and Opt configurations across frameworks and GPUs.

Table 2. Mean peak VRAM usage (MB) for Base and Opt configurations by GPU and framework. Values are averaged across the 10 repetitions.

Framework	GPU	Base	Opt
TensorFlow	GH200	12215.5	1667.5
	L40S	12358.2	4370.5
	RTX 4090	10859.0	5096.8
PyTorch	GH200	18967.2	8364.8
	L40S	18911.0	18646.0
	RTX 4090	18726.0	18600.4
MONAI	GH200	7282.0	6772.0
	L40S	7400.0	6814.0
	RTX 4090	7400.0	6814.0

The decimal values in Table 2 result from averaging the peak VRAM measured across the 10 repetitions of each configuration. MONAI was the most stable and memory-efficient implementation, remaining in a narrow range of approximately 6.8–7.4 GB of VRAM across all GPUs, with a small but consistent reduction in the optimized version. In contrast, PyTorch-Base showed the highest memory footprint, reaching about 18.7–19.0 GB. PyTorch-Opt substantially reduced peak VRAM on the GH200, but only marginal changes were observed on the L40S and RTX 4090. TensorFlow showed the largest memory reduction after optimization: TensorFlow-Opt decreased peak VRAM from about 10.9–12.4 GB to 1.7–5.1 GB, a pattern consistent with the combined effect of mixed precision and backend-level optimizations. Overall, these results indicate that the impact of the optimized pipeline on memory usage was not uniform across frameworks or GPUs, but it was particularly pronounced in TensorFlow and, to a lesser extent, in MONAI.

5.5 Ablation Study of the Optimization Stack

To better understand the contribution of each optimization used in the final pipeline, we performed a component-wise ablation study. Starting from the baseline configuration, we enabled one optimization at a time (DALI, AMP/FP16, cosine learning-rate scheduling, and EMA), and also evaluated the full optimized configuration. Table 3 summarizes the training time, mean throughput, and final validation AUC for all configurations.

As shown in Table 3, AMP/FP16 was the main source of computational gains in this setup. When enabled alone, it reduced total training time from 3865.5 s to 2253.6 s and increased mean training throughput from 671.7 to 1327.6 img/s, while causing only a small drop in final validation AUC from 0.9858 to 0.9824. In contrast, DALI alone did not improve training throughput in this configuration and slightly reduced AUC, suggesting that the baseline was not primarily limited

Table 3. Training time, mean throughput, and final validation AUC for each experimental configuration.

Run	Time (s)	Mean Thr. (img/s)	Final AUC
baseline	3865.5	671.67	0.9858
dali_only	3895.0	659.48	0.9805
fp16_only	2253.6	1327.58	0.9824
cosine_only	3870.5	669.77	0.9851
ema_only	3869.6	668.70	0.9790
full_opt	2526.6	1090.02	0.9854

by the input pipeline. The cosine schedule had a nearly neutral effect on both accuracy and efficiency, indicating that its contribution is better interpreted as training stabilization than as a direct performance booster. EMA alone reduced final validation AUC to 0.9790 without improving training time or throughput, making it the weakest individual component in this ablation.

When all optimizations were combined, the final configuration preserved baseline-level predictive performance, with final validation AUC remaining practically unchanged (0.9858 vs. 0.9854), while reducing training time to 2526.6 s and increasing mean training throughput to 1090.0 img/s. These results indicate that the best overall trade-off was achieved by the combined optimized pipeline, with AMP accounting for most of the speedup and the remaining techniques contributing mainly to the final balance between efficiency and predictive performance.

5.6 Summary of Base-to-Optimized Gains

This section summarizes the overall effect of moving from the Base configuration to the Opt configuration for each framework and GPU. Table 4 reports the absolute AUC variation and the relative changes in throughput, training time, and peak memory usage. Positive values indicate an increase after optimization, whereas negative values indicate a reduction.

Overall, the results show that optimization improved computational efficiency in all frameworks, but with different effects on predictive performance. TensorFlow achieved the largest throughput gains, although with a reduction in AUC. PyTorch showed the strongest balance between AUC improvement and training-time reduction, while MONAI presented smaller but more stable changes across the evaluated platforms.

Table 4. Base-to-Opt changes in AUC, throughput, training time, and memory.

Framework	GPU	AUC	Thr.	Time	Memory
TensorFlow	GH200	−0.027	+186.9%	−50.2%	−86%
	L40S	−0.035	+288.8%	−52.1%	−65%
	RTX 4090	−0.028	+104.1%	−15.0%	−53%
PyTorch	GH200	+0.067	+250.4%	−70.3%	−56%
	L40S	+0.068	+185.3%	−61.5%	−1%
	RTX 4090	+0.066	+183.1%	−60.6%	−1%
MONAI	GH200	+0.005	+71.3%	−47.3%	−7%
	L40S	+0.010	+39.8%	−33.8%	−8%
	RTX 4090	+0.003	+84.1%	−53.4%	−8%

6 Conclusion

Finally, our study compared six training pipelines on a GPU for diabetic retinopathy detection using TensorFlow, PyTorch, and MONAI on three NVIDIA GPU platforms. Overall, AUC varied less between GPUs than between software pipelines, while throughput and training time were strongly influenced by both framework and hardware. TF-Opt achieved the highest throughput on the GH200 and L40S GPUs, while PT-Opt showed the best overall balance, reaching 98.5% AUC on all three GPUs, along with considerable efficiency gains.

All our results reflect the behavior of fully optimized pipelines, and not isolated effects of the structure, since each optimized version combined multiple changes, including GPU-based data loading, mixed precision, compiler optimizations, tensor layout adjustments, and modified training schedules. In this configuration, throughput increased by up to 288.8% and training time decreased by up to 70.3%. The findings suggest that pipeline optimization can make training workflows for diabetic retinopathy more practical in resource-constrained environments. Future work should evaluate larger datasets, multi-GPU environments, and other medical imaging tasks.

Acknowledgments. This research was conducted with support from Petrobras (concessão n.o 2020 /00182-5), CNPq and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr). Some experiments in this work used the PCAD infrastructure (http://gppd-hpc.inf.ufrgs.br), at INF/UFRGS.

References

1. Alyoubi, W.L., Shalash, W.M., Abulkhair, M.F.: Diabetic retinopathy detection through deep learning techniques: a review. Inform. Med. Unlocked **20**, 100377 (2020)
2. Tsiknakis, N., et al.: Deep learning for diabetic retinopathy detection and classification based on fundus images: a review. Comput. Biol. Med. **135**, 104599 (2021)
3. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2818–2826 (2016)
4. Mutawa, A.M., Alnajdi, S., Sruthi, S.: Transfer learning for diabetic retinopathy detection: a study of dataset combination and model performance. Appl. Sci. **13**(9), 5685 (2023)
5. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 248–255 (2009)
6. Bećirović, M., Kurtović, A., Smajlović, N., Kapo, M., Akagić, A.: Performance comparison of medical image classification systems using TensorFlow Keras, PyTorch, and JAX. arXiv arXiv:2507.14587 (2025)
7. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, vol. 32, pp. 8024–8035 (2019)
8. Li, S., et al.: PyTorch distributed: experiences on accelerating data parallel training. Proc. VLDB Endow. **13**(12), 3005–3018 (2020)
9. Abadi, M., et al.: TensorFlow: a system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI '16, pp. 265–283 (2016)
10. Nguyen, Q.-H., Doan, H., Thoai, N.: Performance evaluation of distributed training in Tensorflow 2. In: 2020 International Conference on Advanced Computing and Applications (ACOMP), pp. 155–159 (2020)
11. OpenXLA Project: XLA architecture. https://openxla.org/xla/architecture. Accessed 20 Mar 2026
12. Cardoso, M.J., Li, W., Brown, R., et al.: MONAI: an open-source framework for deep learning in healthcare. arXiv arXiv:2211.02701 (2022)
13. Sebastian, A., Elharrouss, O., Al-Maadeed, S., Almaadeed, N.: A survey on deep-learning-based diabetic retinopathy classification. Diagnostics **13**(3), 345 (2023)
14. Bhulakshmi, D., Rajput, D.S.: A systematic review on diabetic retinopathy detection and classification based on deep learning techniques using fundus images. PeerJ Comput. Sci. **10**, e1947 (2024)
15. Zhou, Y., Wang, B., Huang, L., Cui, S., Shao, L.: A benchmark for studying diabetic retinopathy: segmentation, grading, and transferability. IEEE Trans. Med. Imaging **40**(3), 818–828 (2021)
16. Martínez-Murcia, F.J., Ortiz, A., Ramírez, J., Górriz, J.M., Cruz, R.: Deep residual transfer learning for automatic diagnosis and grading of diabetic retinopathy. Neurocomputing **452**, 424–434 (2021)
17. Hacisoftaoglu, R.E., Karakaya, M., Sallam, A.B.: Deep learning frameworks for diabetic retinopathy detection with smartphone-based retinal imaging systems. Pattern Recogn. Lett. **135**, 409–417 (2020)
18. Alam, M.N., et al.: Contrastive learning-based pretraining improves representation and transferability of diabetic retinopathy classification models. Sci. Rep. **13**(1), 6047 (2023)

19. Goh, J.H.L., et al.: Comparative analysis of vision transformers and conventional convolutional neural networks in detecting referable diabetic retinopathy. Ophthalmol. Sci. **4**(6), 100552 (2024)
20. Gulshan, V., et al.: Performance of a deep-learning algorithm vs manual grading for detecting diabetic retinopathy in India. JAMA Ophthalmol. **137**(9), 987–993 (2019)
21. Dörrich, M., Fan, M., Kist, A.M.: Impact of mixed precision techniques on training and inference efficiency of deep neural networks. IEEE Access **11**, 57627–57634 (2023)
22. He, X., Sun, J., Chen, H., Li, D.: Campo: cost-aware performance optimization for mixed-precision neural network training. In: 2022 USENIX Annual Technical Conference, USENIX ATC '22, pp. 505–518 (2022)
23. Li, B., et al.: Optimizing data layout for training deep neural networks. In: Companion Proceedings of the Web Conference 2022, pp. 548–554 (2022)
24. Um, T., Oh, B., Seo, B., Kweun, M., Kim, G., Lee, W.-Y.: FastFlow: accelerating deep learning model training with smart offloading of input data pipeline. Proc. VLDB Endow. **16**(5), 1086–1099 (2023)
25. Leclerc, G., Ilyas, A., Engstrom, L., Park, S.M., Salman, H., Mądry, A.: FFCV: accelerating training by removing data bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12011–12020 (2023)
26. dos Reis, M.A., Künas, C.A., et al.: Advancing healthcare with artificial intelligence: diagnostic accuracy of machine learning algorithm in diagnosis of diabetic retinopathy in the Brazilian population. Diabetol. Metab. Syndr. **16**, 209 (2024)
27. Balouek, D., et al.: Adding virtualization capabilities to the Grid'5000 testbed. In: Ivanov, I.I., van Sinderen, M., Leymann, F., Shan, T. (eds.) CLOSER 2012. CCIS, vol. 367, pp. 3–20. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-04519-1_1
28. NVIDIA: Core Concepts. In: NVIDIA cuDNN Frontend Documentation. https://docs.nvidia.com/deeplearning/cudnn/frontend/latest/developer/core-concepts.html. Accessed 20 Mar 2026
29. Chan, H.-L., et al.: Deep neural network for the detections of fall and physical activities using foot pressures and inertial sensing. Sensors **23**(1), 495 (2023)
30. Shapiro, S.S., Wilk, M.B.: An analysis of variance test for normality (complete samples). Biometrika **52**(3/4), 591–611 (1965)
31. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bull. **1**(6), 80–83 (1945)

Topology-Aware Distributed Ensemble Learning Using the vCube Hierarchical Hypercube Overlay

Charles Giovane Salles, André Luiz Brun, and Luiz Antonio Rodrigues(✉)

Western Parana State University, Cascavel, Brazil
{charles.salles,andre.brun,luiz.rodrigues}@unioeste.br

Abstract. The increasing volume of distributed data poses challenges for scalable machine learning, including communication overhead, privacy concerns, and system robustness. Distributed learning approaches address these issues by allowing models to be trained locally while exchanging knowledge among nodes. However, the efficiency and resilience of such systems strongly depend on the communication topology used for model dissemination. This paper proposes a topology-aware distributed ensemble learning architecture based on the vCube hierarchical hypercube-like overlay. In this approach, nodes independently train local classifiers using private datasets and disseminate the resulting models through the vCube communication structure. This mechanism enables efficient and fault-tolerant model sharing without transferring raw data, allowing each node to construct a distributed ensemble classifier. The architecture is evaluated using a dataset with more than 1.4 million samples distributed across multiple nodes. Experiments consider heterogeneous data distributions and node failures. Results show that the proposed approach improves classification accuracy compared to a decentralized tree-based dissemination strategy while maintaining robustness under node failures and significantly reducing communication costs.

Keywords: Distributed Machine Learning · Fault Tolerance · vCube

1 Introduction

The rapid growth of data generated by digital systems has increased the demand for scalable machine learning solutions [20]. In many domains such as mobile sensing, healthcare, smart cities, and monitoring systems, data is generated and stored across geographically distributed locations. Although centralizing data can simplify model training, it introduces challenges including high communication costs, privacy concerns, and potential single points of failure [18].

Distributed learning has emerged as an alternative paradigm that enables models to be trained using decentralized data sources [19]. In this setting, each node performs local training using its own dataset and collaborates with other

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 321–339, 2026.
https://doi.org/10.1007/978-3-032-30491-9_21

nodes to share knowledge without transferring raw data. This strategy reduces communication overhead while preserving data privacy and enabling knowledge integration across the network.

Several architectures have been proposed to support distributed learning [10,14]. Centralized approaches rely on parameter servers to aggregate models, which may introduce communication bottlenecks and single points of failure. Hierarchical structures such as tree-based architectures improve scalability but remain vulnerable to failures in upper-level nodes. Peer-to-peer approaches provide greater robustness by distributing responsibilities among nodes, but require efficient mechanisms to coordinate model dissemination and integration.

Ensemble learning is another strategy commonly used in distributed environments, where multiple classifiers trained on different datasets are combined to produce a final prediction [2,17]. Distributed ensemble methods are particularly attractive when nodes train models using local data and exchange them for collaborative inference. However, the efficiency and robustness of these systems strongly depend on the communication topology used for disseminating models.

In this work, we explore the use of the vCube topology [3], a hierarchical hypercube-based communication structure designed for scalable and fault-tolerant distributed systems. The vCube topology organizes nodes into clusters that grow logarithmically, enabling efficient information dissemination while maintaining resilience to node failures. Although vCube has been applied in several distributed system scenarios, its potential in distributed machine learning environments remains largely unexplored.

This paper proposes a distributed ensemble learning architecture that leverages the vCube topology to disseminate locally trained classifiers across nodes. Each node trains a model using its local dataset and shares the resulting classifier following the vCube communication protocol. After dissemination, nodes obtain the classifiers generated across the network and construct a distributed ensemble model for prediction. This strategy enables collaborative knowledge integration while maintaining decentralized training and robustness to node failures.

The proposed architecture is evaluated using a human activity recognition dataset containing more than 1.4 million samples. Experiments consider multiple data distribution scenarios, including balanced distributions, heterogeneous dataset sizes, and non-stratified class distributions. We also evaluate system behavior under node failures to analyze the fault-tolerance properties of the vCube topology. The results are compared with a decentralized tree-based architecture proposed in [19].

This paper makes the following contributions:

1. A topology-aware distributed ensemble learning architecture based on the vCube hierarchical hypercube overlay.
2. A fault-tolerant model dissemination strategy that tolerates node failures without interrupting ensemble construction.
3. An experimental evaluation under heterogeneous data distributions and node failures.

4. An empirical comparison with a decentralized tree topology, demonstrating improved robustness and accuracy.

The remainder of this paper is organized as follows. Section 2 presents background concepts and related work. Section 3 describes the proposed methodology. Section 4 details the configuration of the experiments performed while Sect. 5 presents and discusses the experimental results. Finally, Sect. 6 concludes the paper and outlines directions for future work.

2 Background and Related Work

This section presents the main concepts related to distributed machine learning, ensemble learning, and communication topologies used in decentralized systems, including vCube. In addition, we review previous work addressing distributed model training and knowledge sharing in large-scale environments.

2.1 Distributed Machine Learning and Ensemble Learning

The rapid growth of data volumes has motivated the development of distributed machine learning techniques capable of processing data stored across multiple locations. In many real-world scenarios, data is generated and maintained in geographically distributed environments, making centralized training impractical due to communication costs, privacy concerns, and scalability limitations.

Distributed machine learning enables multiple nodes to collaboratively train models without transferring raw datasets. In this setting, each node processes its local data and contributes to the learning process through model updates or knowledge exchange. Several architectures have been proposed to support distributed learning, including centralized parameter servers, hierarchical aggregation structures, and fully decentralized peer-to-peer systems [19]. While centralized approaches simplify coordination, they may introduce communication bottlenecks and single points of failure. Decentralized architectures mitigate these issues by allowing nodes to communicate directly, often using structured or unstructured communication graphs.

Ensemble learning provides a complementary strategy for distributed environments by combining multiple classifiers to improve prediction accuracy and robustness [2]. In this approach, nodes independently train classifiers using their local datasets, and the resulting models are aggregated to produce a global prediction system. This strategy is particularly attractive when datasets cannot be centralized due to privacy or communication constraints. Several ensemble combination strategies have been proposed, including majority voting, sum rules, and product rules [17].

2.2 The vCube Topology

The vCube [3] is a hierarchical virtual topology derived from distributed diagnose algorithms [5]. It relies on an active failure detection mechanism in which

each process periodically tests its peers. If a tested process responds within the expected time window, it is considered correct; otherwise, it is marked as suspect.

The topology considers a distributed system composed of n processes identified from 0 to $n-1$. Each pair of nodes can communicate directly, forming a fully connected network. Nodes may fail and recover, and in our scenario each node represents a site that stores data used for training classification models.

vCube organizes nodes into clusters $s = 1, ..., \log_2 n$, where cluster s contains 2^{s-1} processes. The function $c_{i,s}$ (Eq. 1) defines the ordered list of processes in each cluster, where $\oplus$ denotes the bitwise exclusive OR (XOR) operator.

$$c_{i,s} = \{i \oplus 2^{s-1}, c_{i\oplus 2^{s-1},1}, ..., c_{i\oplus 2^{s-1},s-1}\} \tag{1}$$

Figure 1 illustrates the results of $c_{i,s}$ for a system with eight processes. For each node i, an edge (j, i) exists such that j is the first correct node in $c_{i,s}$ for $s = 1... \log_2 n$. When a node failure or recovery is detected, the testing edges are recalculated. In the example, node n_0 initially tests n_4 in cluster 3, but after n_4 fails it starts testing n_5, the next correct node in $c_{0,3}$.

s	$c_{0,s}$	$c_{1,s}$	$c_{2,s}$	$c_{3,s}$	$c_{4,s}$	$c_{5,s}$	$c_{6,s}$	$c_{7,s}$
1	1	0	3	2	5	4	7	6
2	2,3	3,2	0,1	1,0	6,7	7,6	4,5	5,4
3	4,5,6,7	5,4,7,6	6,7,4,5	7,6,5,4	0,1,2,3	1,0,3,2	2,3,0,1	3,2,1,0

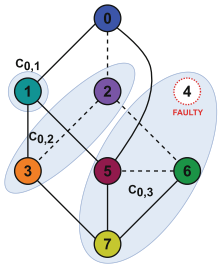


Fig. 1. Clusters of a vCube with $2^3 = 8$ nodes; n_4 is fault.

One key advantage of vCube is its ability to dynamically adapt to node failures. When a node fails, communication can be redirected to alternative nodes within the same cluster, preserving the dissemination process. vCube performs testing rounds with logarithmic latency $\log_2 n$, while the maximum number of tests is bounded by $n \log_2 n$ [5].

Due to these properties, vCube has been applied in several distributed system contexts, including failure detection, monitoring, and scalable information dissemination [1,7,15,16].

2.3 Related Work

Several studies have investigated distributed learning architectures designed to train machine learning models across multiple nodes without centralizing datasets. These approaches differ mainly in how models are trained, exchanged, and aggregated.

Federated learning is one of the most widely adopted paradigms for decentralized training. Clients train models locally and periodically share updates with a central aggregator that produces a global model. The Federated Averaging (FedAvg) algorithm is a well-known example and has been widely applied in mobile and edge computing environments [13]. Subsequent studies addressed challenges such as heterogeneous data distributions, communication efficiency, and system robustness [8]. However, many implementations rely on centralized aggregation, which may introduce bottlenecks and single points of failure.

To address these limitations, decentralized learning approaches allow nodes to communicate directly without a central coordinator. These methods typically rely on peer-to-peer communication graphs to propagate model updates. For instance, decentralized stochastic gradient descent enables nodes to exchange parameters with neighbors and gradually converge to a shared solution [10]. Gossip-based learning follows a similar idea, spreading updates through randomized communication patterns.

Another research direction explores distributed ensemble learning, where nodes independently train classifiers and combine them during inference. Ensemble methods improve predictive performance by aggregating multiple models [2]. In distributed environments, nodes exchange trained classifiers instead of model parameters or training data, enabling collaborative inference across datasets [17].

Communication topology plays a key role in distributed learning systems. Centralized architectures rely on parameter servers, while hierarchical approaches aggregate models at multiple levels [19]. Although hierarchical structures reduce communication overhead, failures near the root may degrade system performance. Fully decentralized architectures distribute communication responsibilities across nodes, improving resilience.

The vCube topology was originally proposed as a scalable failure detection protocol for distributed systems [3]. By organizing nodes into logarithmic clusters, it enables efficient information dissemination while maintaining resilience to failures. Despite its advantages, its application to distributed machine learning remains largely unexplored.

In contrast to traditional federated learning methods that rely on iterative parameter aggregation, this work investigates a distributed ensemble learning architecture in which nodes independently train local models and disseminate them using the vCube communication structure. By leveraging the fault-tolerant properties of vCube, the proposed approach enables scalable model sharing and robust classifier integration in distributed environments, reducing synchronization overhead.

3 Proposed Architecture

This section describes the distributed learning architecture proposed in this work. The architecture consists of three components: (i) local model training at each node, (ii) model dissemination using the vCube topology, and (iii) distributed ensemble construction. The objective is to enable collaborative knowledge sharing among distributed nodes without transferring raw data.

3.1 System Model

We consider a distributed system composed of N nodes interconnected through a communication network. Each node n_i stores a local dataset D_i, which remains private and is not shared with other nodes. Nodes can independently train models, exchange trained models with peers, and may fail during execution. Communication follows the hierarchical hypercube structure defined by the vCube protocol.

Each node trains a local classification model using its dataset:

$$M_i = Train(D_i) \tag{2}$$

After training, models are disseminated across the network using the vCube communication structure. Once dissemination is completed, each node obtains the models trained by all nodes and constructs a distributed ensemble classifier.

Model dissemination occurs through the vCube topology [3], described in Sect. 2.2. The topology organizes nodes into clusters of increasing size, enabling efficient information propagation.

For a system with N nodes, the dissemination process requires $\log_2 N$ communication rounds. During round s, node i exchanges its current models with the first active node in cluster $C_{i,s}$.

Algorithm 1 summarizes the dissemination procedure.

Algorithm 1. Distributed Model Dissemination using vCube

1: $ModelSet \leftarrow \{M_i\}$
2: **for** $s = 1$ to $\log_2 N$ **do**
3: $\quad j \leftarrow$ first active node in cluster $C_{i,s}$
4: $\quad$ send $ModelSet$ to node j
5: $\quad$ receive $ModelSet_j$ from node j
6: $\quad ModelSet \leftarrow ModelSet \cup ModelSet_j$
7: **end for**
8: **return** $ModelSet$

Figure 2 illustrates the communication procedure in a system with $N = 8$ processes. Initially (step A), each node trains its classification model locally, relying on its own data and not considering models from other nodes.

In the next phase (step B), the model exchange process begins. Following the vCube topology, each node transmits its trained model to a predefined counterpart, as determined by the function $c_{i,s}$. In this example, communication follows a structured pairing scheme in which nodes exchange models within predefined groups: node pairs (0,1), (2,3), (4,5), and (6,7) share their respective models trained in step A.

In step B, each node shares its respective models with its neighbor in the first cluster defined by vCube. In the example, the nodes 0 and 1 exchange their models, as do the pairs $(2, 3)$, $(4, 5)$ and $(6, 7)$.

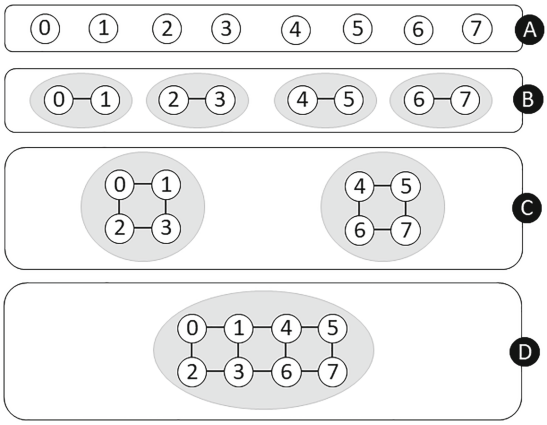


Fig. 2. Communication between 8 nodes based on the vCube topology.

In the third step (C), the pairs formed in the previous step are combined so that the models are now shared by all nodes in each group consisting of four elements (clusters $0-3$ and $4-7$). In the last step (step D), all elements are part of the same group, which means that all nodes in the network know all eight models.

3.2 Distributed Ensemble Construction

After dissemination, each node stores the set of models:

$$\mathcal{M} = \{M_1, M_2, ..., M_N\} \tag{3}$$

For an input sample x, each classifier produces a prediction:

$$y_i = M_i(x) \tag{4}$$

Predictions are combined using ensemble rules. In this work, we evaluate three strategies:

Majority Voting

$$y = \text{mode}(y_1, y_2, ..., y_N) \tag{5}$$

Sum Rule

$$y = \arg\max_{c} \sum_{i=1}^{N} P_i(c|x) \tag{6}$$

Product Rule

$$y = \arg\max_{c} \prod_{i=1}^{N} P_i(c|x) \tag{7}$$

This ensemble allows nodes to integrate knowledge learned from datasets distributed across the network.

4 Experimental Setup

This section presents the classifier, the dataset, and the three test scenarios used to evaluate our decentralized ensemble learning proposal.

4.1 Implementation Details

The classification algorithms were implemented using the WEKA platform [6], which provides a Java library with several machine learning algorithms and tools for data preprocessing and performance evaluation.

We adopted a Multi-Layer Perceptron (MLP) as the base classifier due to its suitability for multi-class problems. The network consists of 12 input neurons, three hidden layers with 16, 32, and 64 neurons, and an output layer with six neurons corresponding to the target classes. Hidden layers use sigmoid activation functions. The model was trained using WEKA default parameters, with learning rate 0.3, momentum 0.2, and 500 epochs.

Both the MLP training and the vCube communication strategy were implemented in Java 8 (64-bit). Experiments were executed on a dedicated machine running Windows 11 with an Intel Core i7 (8th generation) processor, 12 GB RAM, and 1 TB HDD storage.

In the proposed architecture, nodes exchange trained models without performing weight aggregation. Model integration occurs through prediction-based combination, where classifiers share their outputs while keeping their original parameters unchanged.

Execution time is used as a performance metric to compare centralized and distributed scenarios. Since models are trained independently on local data, the distributed approach aims to reduce training time while maintaining coordination through opinion-based model integration.

4.2 The Dataset

We use the dataset "MotionSense: Sensor-based human activity and attribute recognition" [12] to evaluate the performance of our distributed learning proposal. This dataset consists of time series data generated by accelerometer and gyroscope sensors (altitude, gravity, user acceleration, and rotation rate). Data were collected from 6 activities performed by 24 participants of different genders, ages, weight and height. Each participant performed 3 trials of the Downstairs

activity, 3 trials of the Upstairs activity, 3 trials of the Walking activity, 2 trials of the Jogging activity, 2 trials of the Sitting activity, and 2 trials of the Standing activity. All these tests were conducted in the same environment and under the same conditions. A total of 1,412,865 data set items were generated and distributed as shown in Table 1.

Table 1. Distribution of instances among the six classes present in the dataset

Class	Number of Instances	Proportion
Downstairs	131,856	9.33%
Jogging	134,231	9.5%
Upstairs	157,285	11.13%
Standing	306,427	21.69%
Sitting	338,778	23.98%
Walking	344,288	24.37%

In this way, the dataset is suitable for the proposed work as it contains a considerable number of records, which is common in distributed data scenarios.

4.3 Test Scenarios

The first experiment consists of a centralized classification process, referred to as *global classification*. In this setting, the entire dataset is available at a single node and split into training and test sets, preserving the original class proportions. The training set contains 70% of the original dataset (989,004 samples), while the remaining 30% (423,861 samples) is used for evaluation.

In the distributed setting, eight nodes (numbered from 0 to 7) receive partitions of the training data to build local models. Data distribution follows three different scenarios.

In the first scenario, data is evenly distributed among nodes. The dataset contains approximately 990,000 records, so each node receives about 123,000 samples while preserving the original class proportions.

In the second scenario, nodes receive different numbers of samples, but class proportions remain unchanged. Thus, reductions in dataset size are applied proportionally across all classes.

In the third scenario, training data is distributed without preserving class stratification, creating an unbalanced distribution. This scenario simulates situations in which nodes contain more information about certain classes than others, a common characteristic of non-IID data distributions [11].

These scenarios allow us to evaluate the impact of different levels of data heterogeneity on the proposed architecture.

In the distributed setting, nodes share only trained models rather than raw data. Model integration is therefore performed through classifier combination

strategies [9], including *Sum Rule* (aggregation of probability outputs), *Product Rule* (multiplication of class probabilities), and *Majority Voting* (selection of the most frequent prediction).

To evaluate performance, the same test dataset (30% of the original data) is used for both centralized and distributed experiments.

For the distributed scenarios, the vCube topology is used to disseminate trained models among nodes, allowing each node to obtain models trained across the network. The resulting system forms a system of multiple classifiers (SMC) composed of eight classifiers, labeled C_i, where i indicates the node where the classifier was trained (C_0 to C_7).

Experiments consider both fault-free and faulty conditions. Nodes may become temporarily unavailable due to network issues, hardware faults, or maintenance, reflecting realistic distributed environments. Because the decision process is decentralized, the vCube architecture remains operational even when some nodes are unavailable. In such cases, predictions are computed using the remaining models, which may slightly reduce ensemble diversity but does not interrupt system operation.

5 Results

In this section, we present the results obtained for each of the three scenarios outlined in the previous section. The results are first analyzed individually and then consolidated. We conducted tests under both fault-free and faulty conditions. For comparison, we initially implemented a centralized approach using the full dataset, followed by simulations with a decentralized tree-based approach.

In all simulations, only the training instances were loaded, as the test data (representing 30% of the original dataset) was excluded. In distributed tests, we allocated all classes present in the input file (Stairs Down, Stairs Up, Walking, Jogging, Sitting, and Standing) among the nodes based on the specific scenario.

For each isolated class, the percentage of data allocated to each node was defined. For instance, when distributing a class across eight nodes, each node received 12.5% of the available instances.

Although the experiments were conducted with eight nodes, the vCube topology is defined for networks of size 2^d and scales logarithmically with respect to the number of communication rounds. Therefore, the dissemination mechanism remains applicable to larger systems, where the benefits of hierarchical communication structures become even more significant.

5.1 Global Classification Approach

The global classification, which simulates the centralization of training data, was initially performed as a baseline for comparison. We used 70% of the dataset for training and 30% for testing.

This approach offers the advantage of having all available training examples in a single location, allowing the model to learn from the complete dataset.

However, it comes with drawbacks, including the need to consolidate data at a central point, which can create network bottlenecks and expose potentially confidential information to eavesdropping attacks.

In this scenario, the classifier required 20 h and 13 min to train, achieving an accuracy of 82.98%. This is a strong result, considering the problem involves six classes, where random classification would yield an accuracy of only 16%.

5.2 Tests Without Failure Nodes

We first analyze the results in a scenario where nodes operate independently, without communication. Table 2 presents the training time and individual accuracy of each node across the three scenarios described in Sect. 4.

In the first scenario, each node received an equal amount of data from each class. Training times ranged from 2 h and 29 min to 2 h and 55 min, which was expected since all nodes processed the same volume of data. Compared to the global training strategy (20 h and 13 min), this configuration significantly reduces the training time due to the parallel execution of local training processes.

Regarding classification performance, most classifiers (five in total) achieved similar accuracy values, around 43%, while classifiers C_6 and C_7 obtained higher accuracies. Although the datasets maintain the same class proportions, the actual instances assigned to each node differ due to the random partitioning of the training data. As a result, the test set may contain patterns that are closer to the training data of certain nodes, which can lead to variations in classification performance.

Table 2. Individual training on each node in the three scenarios

Classifier	Scenario 1		Scenario 2		Scenario 3	
	Time	Accuracy (%)	Time	Accuracy (%)	Time	Accuracy (%)
C_0	2h50	43.21	4h11	51.80	1h52	32.83
C_1	2h36	43.96	4h16	52.92	1h56	39.16
C_2	2h29	45.41	3h57	47.34	2h57	45.00
C_3	2h29	45.57	4h09	49.94	2h39	47.72
C_4	2h40	38.74	1h02	42.40	2h43	38.37
C_5	2h29	42.53	0h53	42.30	2h04	33.24
C_6	2h38	56.62	0h54	32.45	3h15	58.77
C_7	2h55	54.89	1h01	33.14	3h01	55.64

In scenario 2, the distribution was performed as follows: nodes 0, 1, 2 and 3 received 80% of the training data volume, i.e., each received 20% of the datasets, while nodes 4, 5, 6, and 7 shared the other 20% of the data, i.e., each one received 5% of the datasets. Despite the imbalance in the number of instances present at each node in the network, the relationship between the classes remains following

the original set. As expected, the more information concentrated in a node of the network, the higher the training timing and the accuracy is also better.

In scenario 3, the data was randomly distributed in such a way that both the number of data sets and the proportion of classes present in each node can vary greatly. The configuration of the instances between the eight nodes is shown in Table 3. The values correspond to the percentage of each class present in each node. The Set size column shows the number of instances in each node.

Table 3. Distribution of instances across the nodes for the third scenario

Node	Classes (%)						Set Size
	Downstairs	Upstairs	Walking	Jogging	Sitting	Standing	
0	30	8	8	8	8	8	99,429
1	8	30	8	8	8	8	103,343
2	8	8	30	8	8	8	131,293
3	8	8	8	30	8	8	126,311
4	8	8	8	8	30	8	132,141
5	8	8	8	8	8	30	99,793
6	15	15	15	15	15	15	148,448
7	15	15	15	15	15	15	148,448

The accuracy and the time spent training the models are also shown in Table 2. It can be seen from the values that half of the nodes performed below 40%, with C_0 at 32.83%, C_1 at 39.16%, C_4 at 38.37% and C_5 at 33.24%. The classifiers C_2 and C_3 had a classification accuracy of 45% and 47.72%. Finally, the models C_6 and C_7, which received the most data, had an accuracy of over 55% predicting correctly 58.77% and 55.64% of the instances, respectively.

As expected and demonstrated in the three scenarios, nodes with a similar number of records had a similar training time, again confirming a direct relationship between the number of samples and the learning time. We can see that classifiers with low accuracy trained their models faster, while models with higher accuracy took longer.

Once each model is trained locally, the nodes start to communicate with neighbors, according to the vCube topology, and combine the models hierarchically. Table 4 shows the accuracy rounds to each scenario and combination rule (sum, product and vote).

Step 1. In the first round, each node exchanges messages with its nearest neighbor and shares its classification model. In the first scenario, the accuracy of the approach increased with the sum rule and vote rules. When the product rule is used, the increase in accuracy is smaller and, in some cases, the percentage of correct answers decreases. The group formed by nodes 6 and 7, achieves a

higher performance compared to the others. This can be explained by the fact that these nodes obtained the highest accuracies during individual training.

In the second scenario, the first two groups achieve better results in all approaches than the last two groups. This is because in this scenario, nodes 1, 2, 3 and 4 contain 80% of the data sets, while nodes 5, 6, 7 and 8 only had 20% of the samples in the training set.

In scenario 3, the first group achieved a hit rate of 48.21% with the sum rule, reflecting the low individual performance of the nodes in this group. Similarly, the third group achieved a hit rate of 43.21%. The second group had a success rate of 59.91%, which was higher than the previous groups. Finally, the last group achieved a hit rate of 67.61%, reflecting the high performance of the nodes in this group (mainly due to the sets containing the largest amount of training instances). Concerning other classifier combination approaches in this scenario, a similar behavior as for the sum rule can be observed, i.e. for both majority voting and the product rules, groups 1 and 3 perform worse than the others. The second group achieved an average performance, while the fourth group achieved the best performance.

Table 4. Accuracy (%) of the vCube-based SMC for the first scenario (without failures) over the three phases of the process

Step	Groups	Scenario 1			Scenario 2			Scenario 3		
		Sum	Prod.	Vote	Sum	Prod.	Vote	Sum	Prod.	Vote
1	C_0, C_1	53.96	44.62	51.61	63.81	51.66	61.53	48.21	31.76	47.37
	C_2, C_3	58.37	48.26	54.15	61.33	51.31	55.04	59.91	46.07	56.04
	C_4, C_5	50.44	39.41	47.95	50.12	43.94	47.51	43.21	35.74	40.22
	C_6, C_7	69.33	64.29	67.51	37.71	31.71	37.18	67.61	64.71	64.41
2	C_0, ... C_3	60.02	48.46	58.46	66.79	56.22	65.01	52.31	42.49	51.76
	C_4, ..., C_7	61.35	51.51	60.84	45.19	37.48	42.93	61.04	52.16	58.11
3	C_0, ..., C_7	66.23	51.65	64.66	61.21	46.71	61.12	66.04	50.15	64.93

Step 2. In the second round, there are two groups, each consisting of four models. In the first scenario, when combining two classifiers, the accuracy of the first group is 60.02% according to the sum rule, a value slightly higher than the group formed from classifiers C_3 and C_4 (in the previous round), i.e. a small gain was achieved. For the majority voting, the gain was bigger, as the group obtained an accuracy of 58.46%, while the accuracy of classifiers C_3 and C_4 in the previous round was 54.15%. For the product rule, the accuracy of the first group was 48.46%, i.e. a minimal accuracy gain was achieved compared to the previous round. In this case, the approach has a much lower performance than the product and the majority rules. The second group ($C_4 - C_7$) achieves the best accuracy in the three combination scenarios. This is because the nodes with

the best individual performances (classifiers C_6 and C_7) are in this group. For the sum rule, the accuracy was 61.35%, achieving a lower performance than the group of models C_6 and C_7 (in the previous round), which was 69.33%, i.e. the performance of the models trained on nodes 4 and 5 had a negative impact on the accuracy of the second group. For the majority rule, the accuracy of the group was 60.84%, which also represents a decrease in accuracy compared to the groups formed in the previous round. Finally, for the product rule, the accuracy of the group was 51.51%, significantly lower than in the previous round.

In scenario 2, the accuracy of the first group based on the sum, majority voting and product rules have significant increase in accuracy when the four classification models were combined. For the second group, when classification models with low performance were combined, the resulting classification model tended to have lower performance.

In the third scenario, the accuracy of the first group was 52.31% for the sum rule, 51.76% for the majority rule and 42.49% for the product rule. In this group, nodes that performed poorly pulled the overall success rate. When the classifiers from the second group were combined, the accuracy was 61.04% for the sum rule, 58.11% for the majority rule and 52.16% for the product rule. As with the previous group, it was found that despite the nodes with good individual performance, the overall accuracy of the group tended to decrease due to the influence of some weaker classifiers.

Step 3. In the last round of execution, a single group is formed from all classifiers, so that each node has all classification models in the network, allowing the creation of a combination model from all models. In Scenario 1, the sum rule achieved the best accuracy with 66.23% correct answers, followed by the majority rule with an accuracy of 64.66% and the product rule with the worst performance. In Scenario 2, the sum rule also achieved the best performance with an accuracy of 61.21%. Despite the better performance, the result was very similar to that of the majority rule, whose accuracy was 61.12%. Finally, the product rule achieved a performance of 46.71%, mainly penalized by nodes with low performance. In Scenario 3, despite the low hit rate of most nodes in the network, the accuracy increases significantly when combining classification models. This is because the nodes can offer complementary information about the data of the entire system. This fact is confirmed by the fact that the hit rate of the SMC was 66.04% for the sum rule, 64.93% for the majority rule and 50.15% for the product rule.

It is important to note that the proposed solution supports varying the number of distributed nodes, which can lead to reduced training time due to parallelization. However, when the overall dataset remains fixed, increasing the number of nodes may result in smaller local datasets per node, potentially degrading the accuracy of local models and consequently, the performance of the aggregated global model.

Tests with Node Failures. In these tests, one node is considered fault. For a more robust evaluation, we conducted eight executions of the vCube algorithm, systematically removing one node per execution to simulate a failure in model transmission. This left seven nodes available for the combination of classifiers.

Table 5 presents the accuracy values obtained after the third iteration of vCube. Each row represents a scenario where a specific network node is considered faulty. The best performance for each of the eight cases is highlighted in bold, while the worst-case value is underlined.

Table 5. Performance considering each node in the network as a point of failure

Fault Node	Scenario 1			Scenario 2			Scenario 3		
	Sum	Prod.	Vote	Sum	Prod.	Vote	Sum	Prod.	Vote
0	66.43	65.39	51.88	59.01	58.62	45.99	65.82	64.41	52.43
1	64.32	63.39	52.69	57.45	56.77	45.30	**67.75**	66.28	50.61
2	64.31	63.34	51.79	58.28	58.31	44.89	65.37	64.88	52.47
3	64.30	62.86	50.70	60.14	58.99	45.32	61.93	60.65	51.09
4	**68.03**	67.32	55.10	61.72	61.03	46.50	64.38	63.58	50.91
5	65.89	63.99	51.85	61.82	60.36	46.36	66.10	64.73	50.10
6	63.22	61.77	48.62	61.22	61.34	49.91	63.41	62.02	45.46
7	64.04	62.40	48.95	**62.18**	61.20	48.61	62.11	60.97	47.55

Although the results are comparable to Scenario 1 without failure, it was found that the accuracy of the combined model increases when a low-performing node is excluded, as can be seen when Node 4 was excluded and the model accuracy was 68.03%. Similarly, the performance of the combined model decreases when a node with good assertiveness is excluded, as can be seen when Node 6 was excluded and the performance of the combined model dropped to 63.22%.

In the second scenario, the best performance over the eight executions (62.18%) was achieved with the sum combination strategy when we excluded node 7, which was already expected as it is one of the nodes that only received 5% of the training data. The worst hit rate occurred when node 2 was excluded. Considering that this is one of the elements that received the most training data (20%), it was expected that it could lead to a lower performance of the system when it is removed. With this in mind, it should be re-emphasized how important it is that a fault-tolerant system such as vCube ensures that the process only runs when all nodes in the network can submit their models, as the absence of a model can have a detrimental effect on the accuracy of the entire network.

In scenario 3, as noted in the previous scenarios, the best performance is achieved when a node with low representation is present. In this case, this was the case when node 1 was faulty, resulting in an accuracy of 67.75% for the sum rule. The worst performance was achieved when node 7 was excluded, resulting in a hit rate of 45.46% for the combination based on the product rule.

The results after combining the remaining models showed that the system can handle the failure of a network node. However, the individual performance of the classes directly depends on the distribution of the classes among the nodes. If the instances of a less frequent class are stored in a failed node, the system may not be able to handle that class.

By comprehensively analyzing the obtained results, the combination strategy with the product rule always obtained the worst results. The fusion of classifiers based on the sum rule, on the other hand, showed the best performance and only just outperformed the combination based on the majority rule. In addition to the performance of the combination techniques, they always perform worse than strategies using vCube, regardless of the scenario. The only exception is the use of the product rule, which performed worse than the individual classification.

Furthermore, it was observed that even in cases where one of the nodes fails, the vCube performance was always better than the best individual performance, regardless of which node had a communication failure. This shows that vCube is suitable as a strategy for model integration of isolated trained models and is a good alternative for the management of distributed learning systems.

It should also be emphasized that the classification model generated in each node was on average 45KB in size, i.e., in a scenario with 8 nodes in the system, each node can receive 7 classification models simultaneously, generating a total load of 315 KB. On the other hand, if it were possible to move the data for training the classification model to a specific node, 113 MB of information would have to be moved, an amount of data that is approximately 360 times larger.

5.3 Comparison with Baseline Approaches

To compare the approach proposed in this paper, a centralized ensembling and a decentralized tree [19] simulation were performed to evaluate the average accuracy when each of the network nodes fails.

Table 6 presents a comparison of the results obtained by the decentralized and vCube approaches in Scenario 1. In general, the distributed learning system with vCube has a higher average accuracy than the decentralized tree topology because in this strategy, more than one classification model is lost when a node fails, especially the nodes closest to the root of the tree.

Table 6. Comparison between the decentralized tree and the vCube approach

Topology	Scenario 1			Scenario 2			Scenario 3		
	Sum	Vote	Prod.	Sum	Vote	Prod.	Sum	Vote	Prod.
vCube	**65.07**	63.81	51.45	60.23	59.58	46.61	64.61	63.44	50.08
Decentralized Tree	54.89	53.70	44.57	49.92	48.94	40.12	52.45	50.94	42.46

The comparison of the results achieved with the two strategies for Scenario 2 is also shown in Table 6. As in Scenario 1, the vCube-based topology has a higher

average accuracy compared to the decentralized tree topology, again due to the loss of the classification models contained in the nodes of the latter approach.

Finally, Table 6 outlines the results of the two approaches in Scenario 3. As observed in the first two scenarios, the peer-to-peer topology has a higher average accuracy than the centralized tree topology.

In a centralized model (omitted in the Table 6), the accuracy result would be equivalent to vCube, but the failure of the central node would render the unified model unavailable. An all-to-all strategy would guarantee the sharing of the remaining models in the event of a failure of one or more nodes. However, this requires a quadratic number of model shares, whereas in vCube, the number of shares is at most $n \log n$.

6 Conclusion

This work presented a distributed ensemble learning architecture based on the vCube topology, a hierarchical hypercube structure designed to support scalable and fault-tolerant communication among distributed nodes. The proposed architecture allows nodes to train models locally and disseminate them efficiently without transferring raw data.

The experimental evaluation used a dataset with approximately 1.4 million samples distributed across eight nodes under three scenarios that varied in data balance and class stratification. We evaluated three configurations: (i) independent classifiers without communication, (ii) distributed model sharing using the vCube topology, and (iii) scenarios involving node failures.

Results show that the vCube-based architecture improves classification accuracy compared to standalone classifiers and outperforms a decentralized tree-based communication topology. Moreover, the vCube structure demonstrated resilience to node failures, allowing the system to continue operating even when nodes become temporarily unavailable.

Another important finding is that distributed training significantly reduces computational time compared to centralized training while minimizing communication overhead. Across all scenarios, classifier combination using summation and majority voting rules consistently outperformed individual classifiers, highlighting the benefits of model dissemination and ensemble integration.

These results emphasize the importance of communication topology in distributed learning systems. By leveraging the logarithmic communication structure of vCube, the proposed approach enables scalable model dissemination while maintaining robustness in dynamic environments. Although experiments were conducted with eight nodes, the topology supports networks of size 2^d, allowing logarithmic scaling with respect to communication rounds.

Future work will extend the evaluation to larger networks, additional datasets, and more diverse data distributions to better understand the impact of data heterogeneity. Another direction is to investigate communication costs and scalability when disseminating models in larger and more dynamic environments [4]. We also plan to explore adaptive model integration strategies and alternative

learning algorithms, including deep learning models, to further improve ensemble diversity and prediction performance.

References

1. de Araujo, J.P., Arantes, L., Duarte, E.P., Rodrigues, L.A., Sens, P.: VCube-PS: a causal broadcast topic-based publish/subscribe system. JPDC **125**, 18–30 (2019). https://doi.org/10.1016/j.jpdc.2018.10.011
2. Dietterich, T.G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45014-9_1
3. Duarte, E.P., Bona, L.C.E., Ruoso, V.K.: VCube: a provably scalable distributed diagnosis algorithm. In: 5th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems, pp. 17–22 (2014). https://doi.org/10.1109/ScalA.2014.14
4. Duarte, E.P., Weber, A., Fonseca, K.V.: Distributed diagnosis of dynamic events in partitionable arbitrary topology networks. IEEE Trans. Parallel Distrib. Syst. **23**(8), 1415–1426 (2011). https://doi.org/10.1109/TPDS.2011.284
5. Duarte, E.P., Jr., Rodrigues, L.A., Camargo, E.T., Turchetti, R.C.: The missing piece: a distributed system-level diagnosis model for the implementation of unreliable failure detectors. Computing **105**(12), 2821–2845 (2023). https://doi.org/10.1007/s00607-023-01211-8
6. Frank, E., Hall, M.A., Witten, I.H.: The WEKA Workbench. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Francisco, California, Burlington (2016)
7. Jeanneau, D., Rodrigues, L.A., Arantes, L., Duarte, E.P., Jr.: An autonomic hierarchical reliable broadcast protocol for asynchronous distributed systems with failure detection. JBCS **23**, 1–14 (2017). https://doi.org/10.1186/s13173-017-0064-9
8. Kairouz, P., et al.: Advances and open problems in federated learning. Found. Trends Mach. Learn. **14**(1–2), 1–210 (2021). https://doi.org/10.1561/2200000083
9. Kittler, J., Hatef, M., Duin, R., Matas, J.: On combining classifiers. IEEE Trans. Pattern Anal. Mach. Intell. **20**(3), 226–239 (1998). https://doi.org/10.1109/34.667881
10. Lian, X., Zhang, C., Zhang, H., Hsieh, C.J., Zhang, W., Liu, J.: Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, pp. 5336–5346. Curran Associates Inc., Red Hook, NY, USA (2017). https://dl.acm.org/doi/10.5555/3295222.3295285
11. Lu, Z., Pan, H., Dai, Y., Si, X., Zhang, Y.: Federated learning with non-IID data: a survey. IEEE Internet Things J. **11**(11), 19188–19209 (2024). https://doi.org/10.1109/JIOT.2024.3376548
12. Malekzadeh, M., Clegg, R.G., Cavallaro, A., Haddadi, H.: Mobile sensor data anonymization. In: Proceedings of the International Conference on Internet of Things Design and Implementation, IoTDI 2019, pp. 49–58. ACM, New York (2019). https://doi.org/10.1145/3302505.3310068
13. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: International Conference on Artificial Intelligence and Statistics, pp. 1273–1282. PMLR (2017). https://proceedings.mlr.press/v54/mcmahan17a

14. Peteiro-Barral, D., Guijarro-Berdinas, B.: A survey of methods for distributed machine learning. Prog. Artif. Intell. **2**(1), 1–11 (2013). https://doi.org/10.1007/s13748-012-0035-5
15. Rodrigues, L.A., Arantes, L., Duarte, E.P.: An autonomic majority quorum system. In: IEEE 30th AINA, pp. 524–531. IEEE (2016). https://doi.org/10.1109/AINA.2016.73
16. Rodrigues, L.A., Duarte, E.P., de Araujo, J.P., Arantes, L., Sens, P.: Bundling messages to reduce the cost of tree-based broadcast algorithms. In: LADC, pp. 115–124. IEEE (2018). https://doi.org/10.1109/LADC.2018.00022
17. Rokach, L.: Ensemble-based classifiers. Artif. Intell. Rev. **33**(1–2), 1–39 (2009). https://doi.org/10.1007/s10462-009-9124-7
18. Saeed, N., Ashour, M., Mashaly, M.: Comprehensive review of federated learning challenges: a data preparation viewpoint. J. Big Data **12**(1) (2025). https://doi.org/10.1186/s40537-025-01195-6
19. Verbraeken, J., Wolting, M., Katzy, J., Kloppenburg, J., Verbelen, T., Rellermeyer, J.S.: A survey on distributed machine learning. ACM Comput. Surveys (CSUR) **53**(2), 1–33 (2020). https://doi.org/10.1145/3377454
20. Zaharia, M., et al.: Apache spark: a unified engine for big data processing. Commun. ACM **59**(11), 56–65 (2016). https://doi.org/10.1145/2934664

Automating the Technical Validation of FHIR Implementation Guides: A Data-Oriented Framework Within the HfPT Project

Ivo A. Rodrigues[1,2], Leopoldo O. Silva[1,2(✉)], Cátia Pereira[4], Rute Almeida[3], Rui C. Ribeiro[1,2], and Francisco J. Duarte[1,2]

[1] CCG/ZGDV Institute, University of Minho, Azurém Campus, 4800-058 Guimarães, Portugal
leopoldo.silva@ccg.pt
[2] ALGORITMI Research Centre, University of Minho, 4710-057 Braga, Portugal
[3] RISE-Health, Faculty of Medicine, University of Porto, Alameda Prof. Hernâni Monteiro, 4200-319 Porto, Portugal
[4] Healthy Systems, Rua Engenheiro Ferreira Dias, 728 - 40, 4100-246 Porto, Portugal

Abstract. Several HL7 FHIR Implementation Guides were developed under the "Health From Portugal" initiative, particularly with Project A4.4.3 in mind, to address particular healthcare concerns in Portugal. Maintaining the accuracy and clarity of these Implementation Guides is crucial to guarantee compatibility at the national level. Still, conventional manual validation techniques usually take a long time, are prone to errors, and are difficult to scale. This paper discusses the creation and execution of an automated system meant for technical validation and certification assistance for FHIR Implementation Guides. We improved the execution of the official Java FHIR validator on large datasets by using a Python-based orchestration layer, which turned difficult *OperationOutcome* results into simple, well-organized formats. Effective gathering of error measurements and formulation of useful recommendations to improve IG are possible with this approach.

Keywords: HL7 FHIR · Software Engineering · Automated Quality Assurance · Workflow Automation · Data-Oriented Analysis · Interoperability Standards

1 Introduction

The digital change in healthcare systems depends greatly on the smooth sharing of clinical and administrative data across various platforms [1]. In this setting, the HL7 FHIR (Fast Healthcare Interoperability Resources) standard [2] has

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 340–356, 2026.
https://doi.org/10.1007/978-3-032-30491-9_22

become the global common standard for healthcare compatibility, providing a flexible system based on current web technologies. However, the natural flexibility of FHIR requires the development of Implementation Guides (IGs), which are groups of profiles, terminologies, and business rules that tailor the basic standard to particular clinical, administrative, and national needs.

In Portugal, the national interoperability strategy, set by Portaria no 137/2017 [3], has seen significant progress due to the joint efforts of the Shared Services of the Ministry of Health (SPMS) [4] and the HL7 Portugal group [5]. A key achievement in this process is the "Health From Portugal" (HfPT) initiative [6], which has promoted the creation of a strong collection of IGs. These guides utilize the National FHIR Baseline (PT Core) [7] as their foundational semantics, broadening it to include particular situations such as monitoring chronic illnesses and managing hospital workflows. To promote the widespread adoption of these standards, it is essential to establish a validation system based on data that identifies any non-compliance at an early stage, ensuring that the Implementation Guides are technically prepared before undergoing the official HL7 certification process.

Making sure these Implementation Guidess and the data they generate are technically and mathematically correct can help us to better understand complex systems and improve decision-making. It is a tough work to be semantically right [8]. Historically, validation techniques have relied on unit-level testing or manual checking, which are sluggish, suspicious vulnerable to human errors and challenging to implement over the complicated architecture of systems based on FHIR [9]. Though the official HL7 validation tools frequently produce extremely technical and thorough results like the FHIR resource of OperationOutcome. This results can be difficult to be used and understood for non-expert stakeholders [10]. This causes a major delay in the technical certification process of national standards.

This paper outlines an automated system that links knowledge-based validation with a data-driven analysis layer, enabling systematic certification of FHIR standards in the HfPT project. By using a Python-driven orchestration layer on top of the HL7 Java Validator, the suggested solution allows for automated batch processing of resources across various FHIR versions and IG updates. The framework converts raw validation logs into organized, simplified formats, facilitating the systematic collection of error metrics and the creation of useful recommendations. Through the examination of two separate national use cases, the resp-ig [11] and the workflow-ep-ig [12], we show how this automated method improves the quality, consistency, and scalability of the FHIR certification process in Portugal.

2 Previous Work

The robust use of HL7 FHIR as an international standard for interoperability in healthcare is fundamentally based on its ability to maintain data uniformity through profiles and implementation guidelines [13]. However, the mere existence

of profiles and implementation guidelines does not guarantee the much-desired easy sharing of data. To achieve this, very rigorous validation methods are needed to reduce the gap between theoretical definitions and real-world applications [14]. Thus, this section presents a brief overview of the current technological environment regarding FHIR validation tools, existing techniques for ensuring compliance with implementation guidelines, and the specific problems faced by large-scale digital health projects in dealing with national standards with multiple versions. By reviewing the official HL7 validation system and its incorporation into third-party libraries, we establish the necessary basis for understanding the shortcomings of current manual methods and the need for automated coordination.

At the center of the FHIR conformance framework is the official HL7 FHIR Validator CLI [15], a Java-based standard tool made to conduct thorough technical evaluations of resources against basic specifications and specific profiles. This tool functions by importing Implementation Guide packages and terminology resources to evaluate four key aspects: structural integrity, cardinality constraints, value set bindings, and complex business rules represented via FHIRPath invariants. Although the HL7 FHIR Validator CLI is the best option for accuracy, the ecosystem is further improved by important reference libraries, particularly [16] and the Firely SDK for .NET systems [17]. These libraries feature integrated validation systems that are typically part of server-side configurations to ensure data compliance in real-time. Nonetheless, while they are quite effective on their own, these tools primarily serve as fundamental elements that require considerable manual configuration when used in extensive multi-IG settings. This often results in complex *OperationOutcome* reports that are difficult to merge without sophisticated coordination [18].

Even with the progress made in atomic validation methods, there remains a significant gap in the architecture, especially concerning the coordination of these tools for diverse and extensive datasets. Existing frameworks are mainly designed for checking individual units within transactional settings. However, there is insufficient direct assistance for managing structured batch processes that can accommodate various versions and different implementation guidelines at the same time. Additionally, the results generated by these systems are typically located within the common FHIR *OperationOutcome* resource [18], which is fundamentally complex and very technical. Although this aspect is crucial for communications between machines, it presents a major difficulty for stakeholders without specialized knowledge, like clinical implementation guides architects and certification organizations. [10]. At present, there is an evident deficiency in all-encompassing solutions that can unify these distinct outcomes into ongoing statistical reports or translate intricate error logs into simpler, actionable advice for programmers. Consequently, a more sophisticated framework is required to address this shortcoming, converting basic validation information into strategic insights that enhance a thorough quality assurance and certification procedure for nationwide interoperability standards.

Driven by the combined efforts of the Ministry of Health's Shared Services (SPMS) [4] and the local HL7 Portugal (HL7 PT) chapter [5], the evolution of Portugal's digital health landscape reveals a conscious desire to match international standards. From isolated pilot programs, the national interoperability plan has grown over the last ten years into a well-organized governance system directed at developing accepted Implementation Guides [4]. This development led to the establishment of the National FHIR Baseline (PT Core) [7], which acts as a common foundation for developing extensions and profiles customized to the particular administrative and clinical requirements of the Portuguese healthcare system. More sophisticated and field-specific Implementation Guides, such as those for hospital admission procedures [12] and chronic respiratory care [11], have been developed as these standards have improved to meet the complex needs of the national health programs. Still, the increasing number and complexity of these Portuguese profiles underline the need of a single certification organization [4]. Without a regular, automated validation system, the rapid expansion of the national IG directory might cause inconsistencies between application-level messages exchanged within public and private healthcare settings and clinical criteria.

3 Research Methodology

The approach used for this study is based on the Design Science Research Methodology for Information Systems (DSRM-IS) [19], which is a framework tailored for developing and assessing new IT artifacts aimed at addressing intricate issues within organizations. The first step began with identifying a significant operational challenge in the HfPT project: the existing manual FHIR validation processes have scalability and consistency issues. These problems hinder the rapid certification of national health standards. After this was identified, the main objective of the proposed solution was defined as developing an automated, data-driven system capable of performing complete and repeatable validations. Finally, it was also desired that this system would allow complex technical records to be converted into organized and useful information for various stakeholders.

The design and development phase concentrated on building a modular artifact that combines the canonical HL7 Java Validator CLI with a Python-based orchestration layer. This design was developed particularly to manage the ingestion of several Implementation Guide versions and different datasets, therefore guaranteeing the normalization of raw validation outputs into a single central data repository. The framework was used across two different national use cases, resp-ig and the workflow-ep-ig, to show how the artifact can be used in the real world. It looked at both internal reference examples and synthetic application-level messages.

Using automated pivot tables and statistical grouping, the artifact was assessed using a quantitative analysis of the created output to ascertain the framework's effectiveness in detecting structural, semantic, and syntax faults

throughout several FHIR versions. This methodical analysis validates that the artifact not only greatly lowers the time needed for IG certification but also raises the overall quality of the interoperability criteria by revealing repeated failure patterns often missed in conventional manual processes. Following the iterative cycles of the DSRM-IS, this study guarantees that the derived automation framework contributes significantly and robustly to the digital health governance infrastructure in Portugal.

4 The Proposed Automation Framework

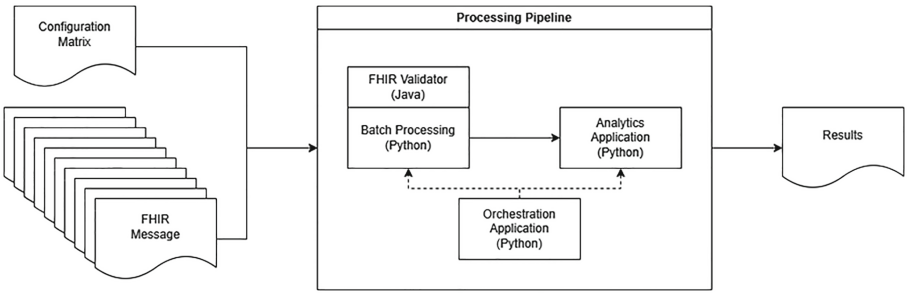


Fig. 1. Architectural workflow of the automated framework, illustrating the hybrid integration between the canonical Java-based validator and the Python-based components for orchestration, batch processing, and data analytics.

The core output of this research is the architecture of the proposed automation framework, created to deliver a scalable, repeatable, and data-driven quality assurance (QA) process for interoperability standards based on FHIR. This architecture acts as a high-level orchestration layer bridging the strategic needs of national digital health governance and low-level, atomic validation engines. The solution allows for the methodical intake of many datasets, from internal Implementation Guide (IG) reference examples to external application-level messages, across many FHIR versions and IG revisions by combining the canonical HL7 Java Validator CLI [15] with a flexible Python-based processing engine. The framework's modular architecture guarantees that the validation process is now a continuous and automated pipeline able to turn raw technical output, like the intricate *OperationOutcome* resource [18], into organized, multi-dimensional analytical reports instead of a fragmented, manual effort. The framework guarantees the technical readiness of interoperability profiles inside the HfPT project [6] by enabling the identification of repeated failure patterns and the creation of standardized maturity indicators via this orchestration.

This architecture acts as a high-level orchestration layer as it connects the strategic needs of national digital health governance and low-level atomic validation mechanisms. Thus, this solution allows for the methodical introduction

of many datasets, from internal reference examples from the Implementation Guide (IG) to external application-level messages, across various FHIR versions and IG revisions, combining the canonical HL7 Java Validator CLI [15] with a flexible Python-based processing mechanism. The modular architecture of this framework ensures that the validation process is now a continuous and automated pipeline capable of transforming raw technical output, such as the intricate *OperationOutcome* resource [18], into organized multidimensional analytical reports, rather than a manual and fragmented effort. The framework ensures the technical readiness of interoperability profiles within the HfPT project [6] as it allows for the identification of the most frequently recurring failure patterns and the creation of standardized maturity indicators through this orchestration.

The framework's system architecture is built on a hybrid integration model (Fig. 1), which combines the high-level flexibility of a Python-based orchestration layer with the specialized compliance of the Java-based validator. This architecture guarantees a clear division of responsibilities: the canonical HL7 Java engine manages the fundamental validation logic, while specialized Python components handle orchestration, batch management, and data analysis. A Python controller inside this design controls the execution environment by dynamically creating the command-line arguments needed to call the Java Archive (JAR) file of the validator. This programmatic invocation enables the exact definition of target FHIR versions, the dynamic loading of pertinent IG packages from both local and online repositories, and the mapping of several input data routes. The orchestrator allows for the methodical processing of varied message batches by abstracting the fundamental complexity of the Java execution environment. The framework therefore keeps perfect alignment with the official HL7 validation logic while acquiring the essential ability to automate sophisticated testing cases, such as multi-version comparisons, which are inherently challenging to handle via the native CLI alone.

The normalization process for data serves as an essential bridge that transforms unprocessed and diverse validation outcomes into a clear and organized analytical framework. Central to this operation is the intake of the FHIR *OperationOutcome* resource, which, while adhering to standards, is both lengthy and complex, thereby often hindering straightforward long-term or statistical evaluations. The structure employs a mapping technique to sort and organize essential diagnostic elements, including severity grades, error messages, and FHIRPath positions, into a streamlined and basic JSON schema (Table 1) to address this issue. Each issue is distinctly associated in this format with its corresponding input factors, which include the specific FHIR edition, the IG URL, and the identifier for the original message. In the final stage of the process, these standardized JSON objects are consolidated into table-based datasets to facilitate the creation of multi-dimensional pivot tables. These tables provide a high-level overview of compliance trends and allow for the methodical grouping and counting of issues by category and version, so enabling the automated identification of recurring abnormalities across huge datasets, a task that would be virtually unattainable by manual log inspection.

Table 1. Mapping of diagnostic fields.

Original Field (FHIR)	Simplified Field	Analytical Objective
`issue.severity`	`severity`	Enables direct filtering by impact (e.g., Error vs. Warning).
`issue.code`	`issue_type`	Provides a standardized technical categorization of the failure.
`issue.details.text`	`description`	Translates low-level system logs into human-readable error descriptions.
`issue.expression`	`location`	Maps the error to the exact FHIRPath location within the resource.
(Metadata)	`fhir_version`	Allows for cross-version longitudinal analysis (e.g., R3 vs. R4).
(Metadata)	`ig_url`	Ensures traceability of the specific Implementation Guide version tested.

Table 2. Example of the structured configuration matrix used to orchestrate simultaneous validation across multiple FHIR versions and implementation guide revisions.

Test ID	FHIR Version	Implementation Guide URL	Input Data Path
`resp-v3`	3.0.0	https://hl7-pt.github.io/resp-ig/	`./data/v3/resp-ig/`
`resp-v4`	4.0.0	https://hl7-pt.github.io/resp-ig/	`./data/v4/resp-ig/`
`wf-ep-v4`	4.0.0	https://hl7-pt.github.io/workflow-ep-ig/	`./data/v4/workflow/`
`pt-core`	4.0.0	https://hl7-pt.github.io/core/	`./samples/core/`

The batch processing approach is based on a matrix-driven methodology that lets one evaluate many interoperability situations at once in one execution cycle. The framework can organize validation across many test cases without needing hand adjustment by employing a structured configuration matrix (Table 2), which includes the particular mapping of FHIR versions, IG URLs, and target resource locations. Particularly important for controlling the shift between regular versions is this design; this is shown by the comparative analysis between versions 3.0.0 and 4.0.0 of the national IGs. Moreover, the approach is tuned to manage large datasets including outside application-level messages generated by healthcare software providers as well as the internal reference examples supplied in the guidelines. The framework ensures that every resource is evaluated within its particular environment while maintaining rigorous isolation between different test runs by automating the repetitive execution of the validation engine. This methodical approach essentially reduces the possibility of human error in parameter selection and offers a reliable, scalable basis for the longitudinal quality assurance of the growing portfolio of interoperability standards created under the HfPT project.

5 Case Studies: Domain Specifics

Within the HfPT initiative [6], different Implementation Guides (IGs) were developed [6] and two of these were chosen as the main examples to highlight the adaptability and resilience of our proposed structure. Within the Portuguese digital health ecosystem, these IGs represent various architectural functions, and thus one focuses on specialized clinical monitoring and the other on the structural management of patient workflows. Therefore, these two cases were chosen because they complement each other, as they allow an evaluation of the structure in relation to horizontal criteria, focused on the structure, and vertical needs, intensive in clinical terms, thus offering a complete testing ground for the diverse interoperability demands of the national health system.

The *resp-ig* [11] is a vertical clinical specification that aims to assist in the screening and treatment of chronic respiratory diseases. From a technical standpoint, this Implementation Guide has a high level of clinical complexity as it extensively uses specialized profiles of observation, condition, and medication administration resources. This guide also includes several customized extensions and links elements to certain medical vocabulary systems, such as LOINC and SNOMED-CT, along with unique coding systems created for Portuguese clinical research projects, in order to capture the subtleties of respiratory pathology. However, validating this Implementation Guide presents a significant challenge because it employs sophisticated questionnaire resources that require strict adherence to FHIRPath invariants and conditional logic to ensure data quality at the point of care. By using the automation framework of *resp-ig*, we hope to confirm the semantic integrity of clinical data and the correct execution of specialized extensions vital for long-term monitoring of patients in respiratory medicine.

The *workflow-ep-ig* [12] represents a standard for horizontal structure, while the previous case study was more focused on clinical aspects. Primarily concentrating on the *Encounter* resource and related administrative metadata, this IG defines the fundamental architecture for guiding the patient's path across the healthcare system. The letter "ep" stands for "episódio", which serves as the primary anchor for all clinical and administrative data produced during a hospital stay or outpatient visit. Strictly specifying state transitions via encounter status codes, this IG technically stresses the integrity of resource referencing guaranteeing that episodes are appropriately linked to patients, practitioners, and organizations. The validation of *workflow-ep-ig* is essential for institutional management and billing systems as it ensures the consistent use of institutional identifiers and admission types throughout the national health network. The automation framework evaluates the IG's capacity to maintain structural coherence throughout these high-level administrative processes, therefore guaranteeing that the structural backbone of the digital health record stays strong and compliant.

Table 3 summarizes the scope of the validation study and thus categorizes the IGs by function and focus. This classification shows the framework's versatility in handling the IGs' structure, as one is highly specialized in the clinical field

Table 3. Characteristics and scope of the selected FHIR Implementation Guides.

IG Identifier	Focus Type	Primary FHIR Resources	Clinical/Administrative Domain
resp-ig	Vertical	*Observation, Condition, Questionnaire, MedicationAdministration*	Chronic Respiratory Disease Monitoring
workflow-ep-ig	Horizontal	*Patient, Organization, Location, Encounter*	Patient Admission and Episode Management

while the other deals with broad administrative standards within the national data infrastructure.

6 Results

By applying this automated structure to the IGs selected by us, we obtained a rich collection of metrics. For the first case, the *resp-ig*, a total of 9 resources were analyzed, and we observed that from version 3.0.0 to version 4.0.0 there was a 70.4% reduction in serious errors, decreasing from 71 to 21. In addition, we detected a 285% increase in informative messages, which increased from 20 to 77. This indicates a successful shift towards improved semantic validation after overcoming structural obstacles. These results, outlined in Table 4, underscore the development of the respiratory care profiles within the HfPT initiative [6].

The performance indicators outlined in Table 4 show a change in the level of issues that marks the development of respiratory care profiles. This evolution from major structural obstacles to data confirmations indicates that as the Implementation Guide becomes more stable, the validation engine can perform more thorough semantic evaluations. This trend is further demonstrated in Table 4, where the notable decrease in errors is visually contrasted with the increase in informational messages, indicating a readiness for extensive interoperability testing.

Table 4. Quantitative validation metrics for the Respiratory Care IG.

FHIR version	Total Resources*	Total Issues	Errors	Warnings	Information
3.0.0	9	101	71	10	20
4.0.0	9	112	21	14	77
Change (%)	0%	+10.9%	-70.4%	+40%	+285%

*Canonical reference resources defining the resp-ig specification profiles.

A corresponding and even more extreme development was noted in *workflow-ep-ig*, as shown in Table 5. In the six message groups evaluated, the framework

achieved complete resolution of major structural faults, decreasing from an average of 13 errors per message in version 3. 0. 0 to none in version 4. 0. 0. Nevertheless, while the severe errors disappeared, the overall number of warnings remained high, reaching a total of 177 issues in version 4. 0. 0. These numerical findings indicate that although administrative criteria reached structural accuracy more quickly than clinical guidelines, both eventually hit a point where the emphasis of validation changed from simple syntax adherence to more intricate relational and terminology standards.

Table 5. Quantitative validation metrics for the Workflow IG.

FHIR version	Total Resources*	Total Issues	Errors	Warnings	Information
3.0.0	6	237	78	159	0
4.0.0	6	177	0	177	0
Change (%)	0%	−25.3%	−100%	+11.32%	+0%

*Canonical reference resources defining the workflow-ep-ig specification profiles.

The classification of the total 627 identified problems offers a detailed perspective on the technical difficulties present in various areas. Structural failures showed the most significant change in both data sets. In the *resp-ig*, these dropped from 41 to 1, while in the *workflow-ep-ig*, the 78 structural mistakes present in version 3. 0. 0, which included absent mandatory components in the `MessageHeader` and unrecognized attributes, were completely removed in version 4. 0. 0. This progress demonstrates the success of the iterative development process in aligning the national IGs with fundamental FHIR profiles. On the other hand, syntax and semantic errors displayed different trends. Syntax mistakes, particularly invalid elements, continued to pose a significant challenge for the *resp-ig* ($n = 13$ in version 4. 0. 0), mainly due to non-compliant resource identifiers that included underscores. In contrast, *workflow-ep-ig* did not present syntax problems ($n = 0$ in version 4.0.0), but encountered notable semantic and terminological issues. Our framework logged 54 invalid code warnings and 144 not-found warnings, related to unresolved participant types and the absence of national *CodeSystems*, such as the `location-physical-type` identifier.

Figure 2 illustrates the transition from high structural non-compliance in the earlier version to a more refined profile in version 4.0.0. In this later iteration, structural and invariant errors were nearly eliminated, leaving only residual syntax (invalid) and terminology (code-invalid) challenges. This shift signals a maturation process where basic resource integrity is achieved, allowing the validation engine to surface more specific data quality issues that were previously obscured by architectural misalignments.

Our automated system detected a significant technical trend, which is the effect of the validation scope, as we can observe in Fig. 3 With this, we understand why version 4.0.0 of *resp-ig* generates a higher number of logs compared to the

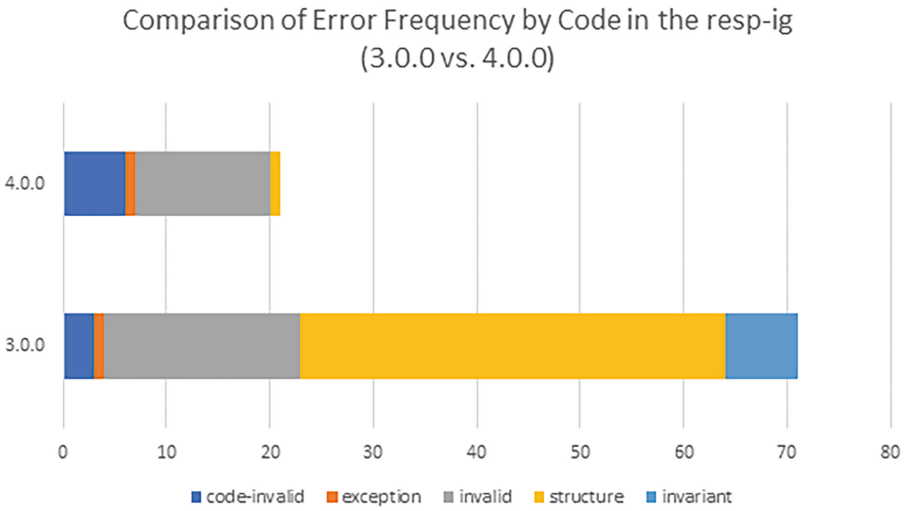


Fig. 2. Comparison of error frequency by issue category in the *resp-ig*.

previous version. The increase in logs is linked to the adoption of more detailed and complete business rules that identify inconsistencies that went unnoticed previously. In version 3.0.0, the frequent occurrence of structural errors ($n = 41$) is an obstacle and, therefore, often led the validation mechanism to prematurely interrupt processing a situation referred to as *short-circuit validation*. When the fundamental structure of an FHIR resource is faulty, the mechanism has difficulty in adequately evaluating more complex components, such as intricate slices, specific extensions, or internal business rules. Concerning the *short-circuit validation* issue, our system now suggests a step-by-step method that focuses on fixing the structure first. We found that requiring the validator to overlook structural components or simulate absent elements might result in incorrect positives in the semantic and business-rule aspects. Instead, our tool supports this sequential approach by offering transparent and well-organized feedback regarding structural obstacles. This enables developers to systematically address fundamental issues, thereby progressively unlocking more complex validation layers in a safe and dependable way. The gradual unlocking of validation layers is clearly illustrated in Fig. 3, where correcting structural mistakes led to an increase in the identification of informative and semantic messages.

In comparison, the upgrade to version 4.0.0 resulted in a significant decrease in structural problems, falling to almost zero ($n = 1$). By addressing these key architectural challenges, the framework allowed the validator to perform a deeper and more comprehensive examination of the entire feature tree. This improvement in clarity led to a 285% increase in informative messages, from 20 to 77, which means successful slice identifications and confirmations of informative rules. This relationship offers concrete proof that the advancement of the Implementation Guide is not only focused on error reduction but also on

improving analytical examination. Our research indicates that a standard can be genuinely verified for semantic correctness and accurate exchange of clinical data only after achieving structural reliability.

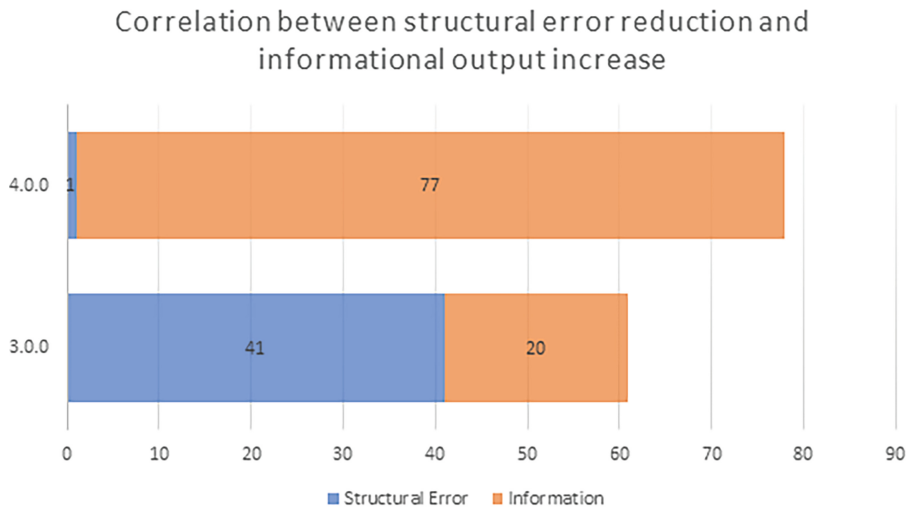


Fig. 3. Correlation between structural error reduction and informational output increase in the *resp-ig*.

Finally, a new aspect of referential integrity emerged as a significant obstacle for *workflow-ep-ig*. The system identified 138 information problems related to inaccessible resources, particularly *Encounter*, *Patient*, *Location*, and *Practitioner*, within the message packets. This observation reveals that, although each feature may conform in terms of structure, the logical connections between them in a clinical workflow still represent a vital area for technical checks. These findings imply that, as structural development progresses, the emphasis of the quality assurance process should shift to ensuring the consistency of the entire exchange of clinical messages, which is crucial for the practical application of national interoperability standards. These results are important as they show that FHIR resources, while they may be valid on their own, can encounter failures at the system level if the connections between them (such as between *Encounter*, *Patient* are disrupted. This underscores the significance of our orchestration layer, which extends beyond just validating individual resources to guarantee the overall functional integrity of the complete clinical message bundle.

The expanded analysis between the two interoperability guides offers a clear illustration of the development journey of national interoperability standards. Version 3.0.0 of both guides presented considerable structural interference, where fundamental architectural inconsistencies often masked more significant problems. However, when we examine the results of version 4.0.0, we detect a more refined and structurally sound environment. In *workflow-ep-ig*, we observe the

removal of all structural errors, which went from 78 to none. This indicates a state of technical readability for administrative headings. In *resp-ig*, by observing the almost complete elimination of persistent and structural errors, which went from 41 occurrences to just one, we realize that the internal clinical structure has become consistent.

Thus, this development highlights several aspects of non-compliance. *resp-ig* demonstrates stabilization in technical improvements, particularly with regard to naming conventions and internationalization codes, such as the use of `PTpt` tags. On the other hand, *workflow-ep-ig* indicates challenges in maintaining relational integrity, where resources may be synchronously correct but are not properly aligned in the message flow.

In short, our automation framework indicates that as an Implementation Guide evolves structurally, the main obstacle to technical certification changes. Thus, it is clear that both Implementation Guides have achieved a high level of structural quality, but the final phase of the quality assurance process requires specific contributions from the relevant area to ensure a smooth implementation within the Portuguese healthcare system.

7 Discussion and Actionable Recommendations

The main internal benefit of the framework is its capability to identify earlier failures in the reference models of the Implementation Guides. Our research indicates that both clinical and administrative guides necessitate different approaches for refinement to guarantee they are technically prepared. For the *resp-ig*, attention must be directed towards enhancing the technical aspects of the metadata. Ongoing syntax mistakes, like using underscores in identifiers such as `Communication.id`, reveal a flaw in manual quality assurance that could hinder the automated input into FHIR servers. In addition, the lack of clarity caused by the failure to address proprietary CodeSystems, like the *Inspirers* system, implies that IG authors need to verify that all terminology resources are either included in the IG package or registered with a national terminology server.

For the *workflow-ep-ig*, the emphasis on improvement turns to maintaining referential integrity. Even though version 4. 0. 0 reached a high standard of structure, the significant number of unattainable resource logs suggests that the logical arrangement of message groups is still not fully realized. IG creators ought to update the reference examples to confirm that all resources, such as *Patient*, *Practitioner*, and *Location*, are properly connected through the `MessageHeader.focus` or `Bundle.entry.request` elements, thus offering a consistent path for the validator [15].

From an external perspective, this system serves as a link for software developers, as it converts complex *OperationOutcome* records into simple compliance guides. By identifying unconventional language codes, such as the use of `PTpt` instead of the accepted `pt-PT`, along with problematic status transitions such as the `'not done'` condition in medication administration, gives developers clear directions for making necessary changes at the application level. In addition,

the framework emphasizes an essential need for grouping organization. Vendors adopting *workflow-ep-ig* are encouraged to extend their focus beyond the compliance of individual features to consider relational integrity. A validation report produced by this framework does not only indicate that a message has failed, but also identifies which specific link in the clinical workflow is faulty. In this way, we have an innovative approach that reduces the development burden on software developers. This helps ensure that messages at the application level are in strict compliance with national standards and thus promotes a more reliable digital health ecosystem.

The automation implemented through this system has transformed the assessment of the HfPT project's technical readiness from a time-consuming manual process to a structured governance model. The most notable effect is the speed of the feedback loop, where replacing manual checks with automated group verification allows national participants to analyze the effects of a single profile modification on various features in a matter of minutes. Thus, this system also offers an unbiased assessment of the maturity of national standards. The ability to visualize the transition from structural problems to semantic challenges allows project leaders to allocate specialized resources more strategically. A high level of structural inconsistencies in initial versions suggests the need for an architectural review, while continuous terminology alerts in subsequent versions indicate the need for deeper infrastructure changes, such as updates to terminology servers. In conclusion, this data-driven strategy ensures that the transition from a preliminary specification to a fully developed national standard is fast and technically sound. The main value of the proposed framework lies in its ability to convert granular technical records into strategic and actionable insights for diverse stakeholders.

Table 6. Actionable Recommendations for Implementation Guide resp-ig and Application Refinement

Finding	Impact	Target	Recommendation
Underscores in Ids	Prevents ingestion in FHIR Servers	IG Authors	Replace _ with - in all resource identifiers.
Invalid Language Code	Breaks localization/search	Software Vendors	Standardize on BCP-47 tags (use pt-PT instead of PTpt).
Unresolved CodeSystems	Semantic validation failure	Project Governance	Bundle proprietary CodeSystems within the IG package.
Deprecated Extensions	Future interoperability risk	IG Authors	Replace `ordinalValue` extension with `itemWeight`.
Broken References	Data integrity loss	Software Vendors	Ensure Encounter resources always link to a valid Patient.

As summarized in Table 6, the findings are categorized not only by their technical impact but also by the specific stakeholder responsible for remediation, thus bridging the gap between low-level validation and national digital health governance.

By differentiating between Implementation Guide authors, software vendors, and project oversight, the framework ensures that validation comments are sent to the correct organizational level. For example, issues such as outdated extensions or unusual syntax in identifiers are recognized as flaws that occur early in the process. When these issues are addressed at the Implementation Guide author level, it is possible to ensure that official reference examples can act as a reliable source of truth for the entire ecosystem. On the other hand, issues related to localization, such as BCP-47 codes, and relational integrity are directed to software vendors, offering a straightforward plan for necessary application-level modifications for technical readiness. Finally, when we talk about broad issues such as the challenge of dealing with proprietary terms like the *Inspirers* CodeSystem, these are identified as governance tasks. All these issues require comprehensive solutions, such as registering code systems in centralized national terminological databases. This specific method transforms FHIR validation from a reactive pass/fail system into a proactive quality improvement process that safeguards the long-term robustness of the Portuguese interoperability framework.

8 Conclusion

The main achievement of this research is the creation of a robust tool that transforms FHIR validation from a disorganized manual process into an automated, data-driven process aligned with the national certification schedule. By integrating the official HL7 validation mechanism through a Python automation layer, it's possible to achieve significant efficiency gains, allowing for the simultaneous processing of hundreds of resources across multiple versions in a much shorter time than traditional manual assessments. Additionally, the structure provided an outstanding degree of technical precision for the HfPT initiative [6]. As shown in the chosen examples for *resp-ig* and the *workflow-ep-ig*, the structure effectively pinpointed minor yet critical compliance problems, such as irregular resource IDs, wrong localization codes, and faulty reference links, which frequently go unnoticed in manual evaluations. Therefore, the dual approach emphasized the various growth challenges encountered by both clinical and administrative protocols, providing national parties with a straightforward way to verify a variety of interoperability models.

Although the system is very efficient, it does have drawbacks due to the existing global FHIR landscape. Its effectiveness and diagnostic precision are heavily linked to the standard Java HL7 validator CLI, which means it can be influenced by any lags in execution or variations in interpretation within the main framework. Additionally, the structure encounters a meaning gap concerning the resolution of terminology, as it has difficulty in addressing proprietary or specific CodeSystems, like the *Inspirers* system, without depending on outside terminology servers or local collections, resulting in a major hindrance. These relationships emphasize that although automation enhances the process, the overall success of technical validation is deeply connected to the presence and dependability of national terminology systems.

The subsequent stage of this study will concentrate on incorporating the automation framework into Continuous Integration and Continuous Deployment (CI/CD) processes. By integrating verification within the ongoing development workflow, IG creators and software providers can achieve built-in compliance, obtaining immediate feedback as they alter profiles or messages. Additionally, the framework will be expanded to include the complete range of the HfPT project, including the Implementation Guides for Adverse Drug Reactions (*adr-ig*), Neonatology (*obs-neonat-ig*), and Community Health (*community-health-ig*). This development strategy aims to establish the framework as a foundational element of the Portuguese health information governance approach, which supports the overarching objective of maintaining national standards that are strong, dependable, and genuinely interoperable.

Acknowledgments. Authors would like to acknowledge the project HfPT - Health from Portugal, co-financed by the RRP (Recovery and Resilience Plan) and the NextGenerationEU Funds: Notice 2022-C05i0101-02 Mobilizing Agendas/Alliances for reindustrialization, Project no C630926586-00465198.

R.A. acknowledges support from Fundação para a Ciência e a Tecnologia, I.P., under the Scientific Employment Stimulus - FCT Tenure (Grant 2023.13694.TENURE.056).

References

1. Sinaci, A.A., et al.: A data transformation methodology to create findable, accessible, interoperable, and reusable health data: Software design, development, and evaluation study. J. Med. Internet Res. **25**, e42822 (2023). https://doi.org/10.2196/42822
2. HL7 International: HL7 FHIR - Fast Healthcare Interoperability Resources Overview (2026). https://www.hl7.org/fhir/overview.html. Accessed 18 Mar 2026
3. Ministério da Saúde: Portaria no. 137/2017: Define os princípios e as normas de interoperabilidade de dados e sistemas de informação de saúde." Diário da República no. 75/2017, Série I, 2017. https://dre.pt/dre/detalhe/portaria/137-2017-106927891
4. SPMS - Serviços Partilhados do Ministério da Saúde: SPMS - Serviços Partilhados do Ministério da Saúde - Home (2024). https://www.spms.min-saude.pt/. Accessed 18 Mar 2026
5. HL7 Portugal: HL7 Portugal - Health Level Seven International (2024). https://www.hl7.pt/. Accessed 18 Mar 2026
6. Health from Portugal: Health from Portugal (HfPT) - Agenda Mobilizadora para o Setor da Saúde (2024). https://www.healthfromportugal.pt/pt/. Accessed 18 Mar 2026
7. HL7 Portugal: HL7 Portugal FHIR National Baseline (PT Core) (2024). https://hl7-pt.github.io/core/. Accessed 18 Mar 2026
8. Leroux, H., Metke-Jimenez, A., Lawley, M.: Towards achieving semantic interoperability of clinical study data with fhir. J. Biomed. Semantics **8**, 41 (2017). https://doi.org/10.1186/s13326-017-0148-7

9. Walonoski, J., Scanlon, R., Dowling, C., Hyland, M., Ettema, R., Posnack, S.: Validation and testing of fast healthcare interoperability resources standards compliance: Data analysis. JMIR Med. Inform. **6**, e10870 (2018). https://doi.org/10.2196/10870
10. T. Benson and G. Grieve, *Principles of health interoperability: NOMED CT, HL7 and FHIR*.07 2016.https://doi.org/10.1007/978-3-319-30370-3
11. HL7 Portugal: Respiratory Care FHIR Implementation Guide (resp-ig) (2024). https://hl7-pt.github.io/resp-ig/. Accessed 18 Mar 2026
12. HL7 Portugal: Workflow Episode of Care FHIR Implementation Guide (workflow-ep-ig) (2026). https://hl7-pt.github.io/workflow-ep-ig/. Accessed 18 Mar 2026
13. Ayaz, M., Pasha, M.F., Alzahrani, M.Y., Budiarto, R., Stiawan, D.: The fast health interoperability resources (fhir) standard: systematic literature review of implementations, applications, challenges and opportunities. JMIR Med. Inform. **9**, e21929 (2021). https://doi.org/10.2196/21929
14. Nan, J., Xu, L.-Q.: Designing interoperable health care services based on fast healthcare interoperability resources: Literature review. JMIR Med. Inform. **11**, e44842 (2023). https://doi.org/10.2196/44842
15. HL7 International: Using the FHIR Validator (2024). https://confluence.hl7.org/spaces/FHIR/pages/35718580/Using+the+FHIR+Validator. Accessed 18 Mar 2026
16. Smile Digital Health: HAPI FHIR: The Open Source FHIR API for Java (2026). https://hapifhir.io/. Accessed 19 Mar 2026
17. Firely: Firely SDK for .NET architectures (2026). https://firely-team.github.io/firely-net-sdk/. Accessed 19 Mar 2026
18. HL7 International: FHIR OperationOutcome Resource Definition. Accessed 18 Mar 2026. https://www.hl7.org/fhir/operationoutcome.html (2026)
19. Peffers, K., Tuunanen, T., Rothenberger, M., Chatterjee, S.: A design science research methodology for information systems research. J. Manage. Inf. Syst. **24**, 45–77 (2007)

Characterizing Lossless GPU Data Compression Across AMD CDNA and RDNA Architectures

Cristiano Künas[1(✉)], Gabriel Freytag[1], Jean Luca Bez[2], Thiago Araújo[1], and Philippe Navaux[1]

[1] Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil
{cakunas,gfreytag,tsaraujo,navaux}@inf.ufrgs.br
[2] Lawrence Berkeley National Laboratory, Berkeley, CA, USA
jlbez@lbl.gov

Abstract. Data movement remains a major bottleneck in high-performance computing workflows, making GPU-accelerated compression an increasingly important optimization. While NVIDIA platforms benefit from mature compression libraries such as nvCOMP, the performance of practical lossless compression on AMD GPUs remains undercharacterized. This paper addresses that gap by porting LZ4, Snappy, and Cascaded from CUDA to HIP and evaluating them across four AMD GPUs spanning three CDNA generations (MI50, MI210, MI300X) and RDNA 3 (RX 7900 XT), using both synthetic datasets and seismic simulation data. Our study makes three contributions. First, we provide a cross-generational characterization of lossless GPU compression on recent AMD accelerators. Second, we present a functional CUDA-to-HIP port of three widely used algorithms, establishing an open baseline for future AMD-specific optimization. Third, we analyze how architectural differences shape compression and decompression behavior across algorithms and datasets. The results show that the MI300X achieves up to 11× higher decompression throughput than the MI50, while RDNA 3 is competitive for compression workloads characterized by irregular memory accesses. We also show that transfers dominate end-to-end execution time on all evaluated platforms, indicating that the main benefits of GPU compression are most likely to emerge in GPU-resident workflows. For realistic seismic floating-point data, lossless compression ratios remain modest ($1.03 - 1.04\times$), suggesting that error-bounded lossy compression is a promising direction for future work.

Keywords: GPU Computing · Data Compression · LZ4 · AMD · HIP · ROCm · High-Performance Computing

1 Introduction

Modern scientific computing faces an increasingly critical challenge: the volume of data generated by simulations is growing faster than our ability to move it

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 357–374, 2026.
https://doi.org/10.1007/978-3-032-30491-9_23

efficiently. High-performance computing (HPC) applications like seismic simulation, climate modeling, and fluid dynamics produce datasets ranging from gigabytes to petabytes, making data movement the dominant bottleneck in many workflows [8]. This I/O wall has motivated significant interest in lossless data compression [10,16,24,26], which is already a critical component in storage systems, from parallel file systems in HPC clusters to cloud-based object stores, where it reduces both capacity costs and transfer overhead.

GPU-accelerated compression has emerged as an attractive solution, leveraging the parallel architecture of graphics processors to achieve throughputs that can exceed storage system bandwidths. On NVIDIA platforms, the nvCOMP library [26] has become the standard, offering highly optimized implementations of algorithms such as LZ4 [10], Snappy [16], and Cascaded [24] that achieve high compression throughput on modern GPUs. This mature ecosystem has enabled researchers to integrate high-performance compression into their CUDA-based workflows seamlessly.

However, the HPC landscape is becoming increasingly heterogeneous. AMD GPUs now power several of the world's most powerful supercomputers, including El Capitan at Lawrence Livermore National Laboratory and Frontier at Oak Ridge National Laboratory, the two highest-ranked systems on the TOP500 list [28], both powered by AMD hardware. This shift creates an urgent need for portable, high-performance software tools capable of running efficiently on GPUs. Unfortunately, a significant **tool gap** exists, while the CUDA ecosystem benefits from years of mature compression libraries, AMD users lack equivalent tools. Existing community efforts provide basic functionality but have not received the same level of optimization, leaving researchers with AMD hardware at a disadvantage for data-intensive workflows.

This paper addresses this gap through three contributions. First, we provide a systematic cross-generational characterization of lossless compression performance on AMD GPUs, including CDNA (Compute DNA) and RDNA (Radeon DNA) architectures, covering both synthetic and seismic datasets. Second, we present a functional CUDA-to-HIP port of three practical algorithms (LZ4, Snappy, and Cascaded), derived from nvCOMP 2.2, thereby establishing an open AMD-compatible baseline for future research and optimization. Finally, we analyze how architectural features such as cache hierarchy, memory bandwidth, and wavefront behavior shape compression and decompression throughput, exposing distinct performance across AMD GPU families.

Our evaluation spans four AMD GPUs (MI50, MI210, MI300X, and RX 7900 XT), using synthetic inputs and realistic TTI seismic wavefields. The results show that decompression throughput scales strongly across generations (MI300X achieves up to 11× the MI50 baseline), that CDNA and RDNA architectures exhibit complementary strengths (RDNA 3 is competitive for compression with irregular access patterns, while CDNA 3 dominates decompression), and that transfer overhead dominates end-to-end time on all platforms, indicating that kernel-level gains require integration into GPU-resident workflows.

The remainder of this paper is organized as follows. Section 2 reviews related work and positions our study within the broader GPU-compression literature. Section 3 provides background on GPU-based compression and motivates the need for AMD-compatible tools. Section 4 describes our porting approach and implementation. Section 5 details the experimental methodology. Section 6 presents the cross-generational performance analysis. Section 7 discusses the implications of the results, limitations of the current study, and directions for future work. Section 8 concludes the paper.

2 Related Work

Data compression has become a central technique for mitigating the widening gap between computational capability and storage or communication performance in HPC systems. Earlier I/O and resilience work already established this motivation: *Hello ADIOS* [22] discussed the growing importance of data-management efficiency in leadership-class systems, while *VeloC* [23] showed that checkpointing frameworks increasingly benefit from techniques that reduce data volume and hide I/O costs. These studies make clear why fast accelerator-side compression is valuable, but they do not characterize how such techniques behave on modern AMD GPUs.

A substantial body of research has explored GPU compression for scientific data, especially in the error-bounded lossy setting. *cuSZ* [27] demonstrated one of the first efficient CUDA-based implementations of SZ on GPUs, and later work such as *cuSZp* [18], *cuSZp2* [17], and *FZ-GPU* [29] pushed the design space toward higher throughput, stronger kernel fusion, and improved compression ratio on NVIDIA devices. Collectively, these works show that GPU compression is now a mature optimization target. However, their implementations and evaluations are overwhelmingly CUDA-centric, so they provide limited guidance for users targeting AMD accelerators.

Lossless GPU compression has likewise advanced, again largely within the NVIDIA ecosystem. *GPULZ* [30] optimized LZSS-style compression for multi-byte data on modern GPUs, while *ndzip-gpu* [19] targeted efficient lossless compression of scientific floating-point arrays. More recently, *FCBench* emphasized that compressor behavior depends strongly on both dataset characteristics and the runtime bottlenecks exposed by a given platform [9]. This line of work is relevant to our study because it highlights that throughput and compression ratio are not intrinsic properties of an algorithm alone, but emerge from the interaction between algorithm, data, compiler toolchain, and hardware architecture.

A smaller but important set of works has started to address portability and broader cross-platform evaluation. *HPDR* argues for portable high-performance data reduction across heterogeneous systems [8]. These efforts move the field beyond single-platform peak results. Still, they do not provide the cross-generational AMD-centric baseline needed by practitioners deploying on AMD-based systems such as El Capitan or Frontier, nor do they focus on the practical question of how CUDA-originated compression kernels behave after translation to HIP.

A particularly relevant recent study is Burtchell and Burtscher (2025), which characterizes parallel data-compression performance across compilers and GPUs [6]. Both papers adopt a measurement-driven perspective, but differ in focus: whereas that work examines compression behavior across compiler and platform combinations broadly, ours specifically targets the AMD ecosystem, emphasizing CUDA-to-HIP portability and cross-generational behavior across CDNA and RDNA architectures.

Our work is positioned at the intersection of compression research, performance portability, and AMD software support. While `nvCOMP` offers a mature GPU-compression stack on NVIDIA [26], `hipCOMP-core` remains an early-access preview on AMD [4]. In contrast to prior work that primarily proposes new CUDA-based compressors or evaluates on NVIDIA-centric platforms, we provide a quantitative, cross-generational characterization on AMD hardware and report on an initial CUDA-to-HIP port, establishing a baseline for the remaining software and optimization gap.

3 Background and Motivation

3.1 GPU-Based Data Compression

In scientific computing, lossless compression is often preferred as it guarantees exact data reconstruction. Among lossless algorithms, LZ4 [10] offers a favorable balance between speed and ratio, Snappy [16] prioritizes minimal latency, and Cascaded compression applies multiple encoding stages (delta + RLE (Run-Length Encoding) + bit-packing) for higher ratios on structured data. All three can be parallelized across GPU threads, with their implementation details described in Sect. 4.3.

NVIDIA's nvCOMP library [26] provides optimized CUDA implementations of these and other algorithms, achieving throughputs that can saturate the memory bandwidth of high-end GPUs. Version 2.2, which serves as the codebase for our HIP (Heterogeneous-Interface for Portability) port, remains widely deployed. However, nvCOMP is tightly coupled to the CUDA ecosystem, limiting its applicability to other platforms.

3.2 The AMD GPU Landscape

AMD has made substantial inroads in high-performance computing, with its GPUs now powering some of the world's fastest supercomputers. This success is built on two distinct GPU architectures optimized for different markets.

The CDNA architecture is geared towards data center and HPC workloads, with an emphasis on high computational throughput, high-bandwidth memory (HBM), and features specifically designed for scientific computing. The evolution from CDNA (MI50/MI100) to CDNA 2 (MI210/MI250X) and finally to CDNA 3 (MI300X) brought substantial improvements in both raw performance and memory bandwidth. The MI300X, in particular, features 304 compute units

and 192 GB of HBM3 memory with over 5 TB/s of bandwidth, making it well-suited for memory-intensive workloads like data compression.

RDNA targets gaming and consumer applications but has found use in cost-sensitive HPC deployments. RDNA 3, exemplified by the RX 7900 XT, introduces a chiplet-based design with separate compute and memory dies, achieving competitive compute performance at lower power consumption. While not designed for datacenter use, RDNA GPUs offer an interesting point of comparison for understanding how architectural choices affect compression performance. Table 1 summarizes the key specifications of the GPUs evaluated in this work.

Table 1. AMD GPU specifications evaluated in this study.

GPU	Arch.	CUs	Memory	BW (GB/s)	TDP (W)
MI50	CDNA	60	16 GB HBM2	1,024	300
MI210	CDNA 2	104	64 GB HBM2e	1,638	300
MI300X	CDNA 3	304	192 GB HBM3	5,300	750
RX 7900 XT	RDNA 3	84	20 GB GDDR6	800	315

AMD's software ecosystem centers on ROCm (Radeon Open Compute), an open-source platform that provides tools for GPU computing. The HIP programming model enables developers to write code that targets both AMD and NVIDIA GPUs with minimal modifications, thereby facilitating portability. However, while HIP simplifies porting CUDA code, achieving optimal performance on AMD hardware often requires architecture-specific optimizations that go beyond simple API translation.

3.3 The Compression Tool Gap

Despite AMD's growing presence in HPC, the availability of optimized GPU compression tools has not kept pace with the hardware. The CUDA ecosystem benefits from nvCOMP—a vendor-supported, production-ready library available since 2020, offering architecture-tuned implementations of LZ4, Snappy, Cascaded, zstd, ANS, and others. In contrast, the HIP/ROCm ecosystem relies on community-driven efforts such as hipCOMP, which appeared in 2022 and remains experimental, with limited algorithm coverage (LZ4 and Snappy), direct-port optimization, and sparse documentation.

This asymmetry creates practical challenges for the growing number of HPC systems built on AMD GPUs. Scientists working on AMD-equipped systems—including El Capitan, Frontier, and others—cannot directly benefit from the mature CUDA compression ecosystem and must either port code themselves, use suboptimal tools, or forgo GPU-accelerated compression entirely. This situation motivates our work: by characterizing the current state of compression on AMD GPUs and providing an open implementation, we aim to bridge this gap and contribute to a vendor-agnostic compression ecosystem for HPC.

4 GPU Compression Library for AMD Platforms

This section describes the implementation of a GPU-accelerated lossless compression library targeting AMD GPUs via the HIP programming model. Our goal was to create a functional and portable implementation suitable for cross-architecture performance characterization, prioritizing correctness and cross-platform compatibility over architecture-specific optimization.

4.1 Porting Approach

Our implementation is based on the open-source nvCOMP 2.2 codebase [26], which provides a batched compression API that processes multiple independent data chunks in parallel—a design well-suited to GPU architectures. We chose this codebase as a starting point because it contains well-structured, GPU-optimized implementations of the three target algorithms (LZ4, Snappy, Cascaded) and its batched API model maps naturally to the HIP programming model. Our porting methodology follows the standard CUDA-to-HIP translation workflow documented in the HIP programming guide [3] and involves three phases.

Phase 1: API Translation. We used AMD's `hipify-perl` tool [3] to perform the initial mechanical translation of CUDA source code to HIP. The tool handled approximately 90% of the API surface automatically; the remaining manual modifications involved replacing CUB [25] device primitives with hipCUB equivalents, adapting warp-level intrinsics, and resolving header conflicts between CUDA and HIP include paths.

Phase 2: Build System Adaptation. We restructured the CMake build to detect the HIP compiler (`hipcc`) and target specific GPU architectures via their GFX identifiers (e.g., `gfx906` for MI50, `gfx942` for MI300X, `gfx1100` for RX 7900 XT). A build-time option controls whether the library compiles with 32-thread or 64-thread wavefront assumptions (Sect. 4.2).

Phase 3: Correctness Validation. We validated the port using a correctness check: for every test input, we verified that `decompress(compress(data)) == data` with bit-exact equality across all chunk sizes, data types, and algorithms. This round-trip validation is a standard technique for verifying lossless compression implementations and ensures that no data corruption was introduced during the porting process.

4.2 Wavefront Size Adaptation

NVIDIA GPUs use 32-thread *warps*, while AMD CDNA GPUs default to 64-thread *wavefronts* [2]; RDNA architectures use 32-thread wavefronts natively [3]. This difference affects compression kernels that rely on warp-level primitives (ballot, shuffle, reduction). To not rewrite all warp-level logic, we adopted a compile-time strategy, defining a `warpsize` constant and a `lane_mask_t` type selected at build time (64-bit masks with `warpsize = 64` for CDNA targets, and

32-bit masks with `warpsize = 32` for RDNA 3), preserving correctness across architectures while exploiting the wider wavefront when available. However, it does not address deeper optimization opportunities, such as restructuring thread cooperation patterns to better match AMD's memory hierarchy, which we leave for future work.

4.3 Chunked Compression Model

All three algorithms use a *batched chunked* compression model [26], where input data is divided into fixed-size chunks compressed independently in parallel (Fig. 1). Each chunk is assigned to one or more thread blocks, enabling thousands to be processed concurrently. The algorithms differ in their thread-to-chunk mapping. **LZ4** assigns one warp per chunk with a private hash table (64 KB) for match-finding. **Snappy** uses two cooperating warps (match-finding + encoding). And **Cascaded** applies a multi-stage pipeline (delta + RLE + bit-packing) with 128 threads per 4 KB partition. These different parallelization strategies lead to distinct performance characteristics across architectures, as analyzed in Sect. 6.

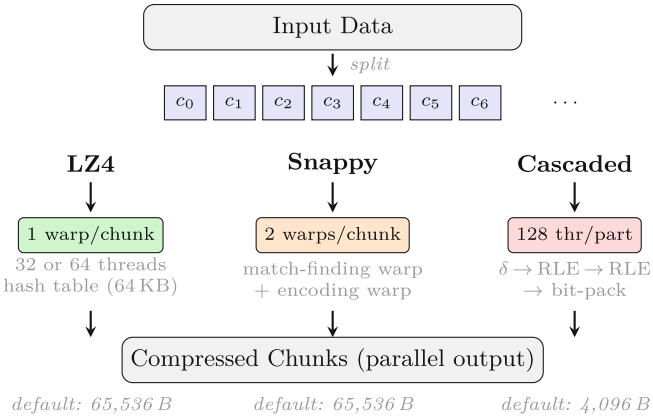


Fig. 1. Batched chunked compression model. Input data is split into independent chunks processed in parallel. Each algorithm assigns different thread configurations per chunk: LZ4 uses a single warp, Snappy uses two cooperating warps, and Cascaded uses 128 threads per 4 KB partition with a multi-stage encoding pipeline.

4.4 Benchmark Suite and Reproducibility

To ensure reproducibility across diverse GPU platforms, we containerized the complete build and execution environment using Singularity [20], the standard container runtime in HPC due to its rootless execution model and native GPU

passthrough. Our container image is based on Ubuntu 22.04 with ROCm 7.0.1 and accepts a GPU architecture argument at build time, ensuring that the same software stack is used across all platforms while targeting architecture-specific binaries.

We developed a benchmark suite that compiles alongside the compression library, with one executable per algorithm. Throughput is measured using GPU-side event timing (`hipEvent`), which captures only kernel execution time and excludes host-device data transfer overhead [7,12]. Throughput is reported in GB/s of *uncompressed* data:

$$\text{Throughput (GB/s)} = \frac{\text{uncompressed size (bytes)}}{t_{\text{kernel}} \times 10^9} \tag{1}$$

where t_{kernel} is the mean kernel execution time in seconds over all timed iterations. An automation layer orchestrates all algorithm×dataset×chunk-size combinations, collecting results into structured CSV files.

5 Experimental Methodology

This section details the experimental setup, dataset design, and measurement protocol used to evaluate compression performance across AMD GPU generations. Our methodology is designed to enable reproducible comparisons and to isolate GPU-level performance from system-level variability.

5.1 Hardware Platforms

We evaluated compression performance on four AMD GPUs spanning three architectural generations, as detailed in Table 1. The MI50 (CDNA) serves as our baseline, representing first-generation compute-focused AMD GPUs with 16 GB of HBM2 memory and 1,024 GB/s bandwidth. The MI210 represents CDNA 2, the architecture widely deployed in systems such as LUMI [11] at CSC Finland. The MI300X is AMD's current flagship datacenter accelerator based on CDNA 3, featuring 192 GB of HBM3 memory with over 5.3 TB/s of aggregate bandwidth. The RX 7900 XT provides a consumer-grade RDNA 3 comparison point, featuring a chiplet-based design with 20 GB GDDR6 and AMD's Infinity Cache.

Computing Environments. The MI50, MI210, and MI300X experiments were conducted using the Grid'5000 testbed [5], a large-scale distributed infrastructure for experimental research in computer science. The RX 7900 XT experiments were performed at the PCAD[1] laboratory at UFRGS. On all platforms, experiment execution used Singularity containers with identical ROCm 7.0.1 userspace (Sect. 4.4), differing only in the GPU architecture target used at build time.

[1] https://pcad.inf.ufrgs.br/.

5.2 Benchmark Datasets

We designed four dataset categories spanning the compressibility spectrum (Fig. 2), combining synthetic patterns with real scientific data [7,27].

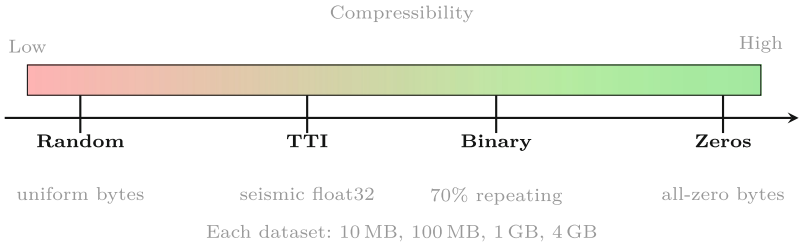


Fig. 2. Benchmark dataset compressibility spectrum.

Three **synthetic datasets** provide controlled evaluation. *Zeros* (all-zero bytes, maximum compressibility), *Random* (uniform bytes from `/dev/urandom`, incompressible), and *Binary* (70% repeating sequences, 30% random, moderate compressibility).

TTI (Tilted Transversely Isotropic) Wavefield Data provides a realistic benchmark from computational geophysics [1,14]. The dataset was extracted from a 3D seismic wave-propagation simulation on a 448^3 grid with 201 timesteps, producing 67.3 GB of single-precision floating-point data in RSF format [15]. We extract timesteps from the middle of the simulation (starting at timestep 100) to avoid early timesteps dominated by near-zero initialization values that would inflate compression ratios.

Each dataset category was evaluated at four sizes: Small (10 MB), Medium (100 MB), Large (1 GB), and XLarge (4 GB), yielding 4 data types $\times$ 4 sizes = 16 test files and 48 benchmark configurations per GPU.

5.3 Measurement Protocol

Our measurement protocol follows established GPU benchmarking practices [7,12] and is designed to minimize measurement noise while capturing representative performance. Throughput is measured using GPU-side event timing (`hipEvent`), isolating kernel execution from host-device transfers, as detailed in Sect. 4.4.

Each benchmark configuration executes $w = 3$ warmup iterations followed by $n = 10$ timed iterations, allowing the GPU clock to stabilize, JIT compilation to complete, and memory caches to reach steady state. We report the arithmetic mean over timed iterations. All experiments use the default chunk sizes described in Sect. 4.3.

We report three primary metrics: *compression throughput* (GB/s), computed as uncompressed data size divided by compression kernel time (Eq. 1); *decompression throughput* (GB/s), using decompression kernel time; and *compression ratio*, the original size divided by compressed size (higher values indicate better compression). Cross-generation comparisons use *speedup* relative to the MI50 baseline, calculated as the ratio of throughput on the target GPU to throughput on the MI50 for the same algorithm, dataset, and size.

6 Results: Cross-Generation Analysis

This section presents our experimental evaluation of LZ4, Snappy, and Cascaded compression across AMD GPU generations. We analyze throughput across algorithms and data types, execution time composition, compression ratios, scaling behavior, and cross-generation speedup.

6.1 Compression and Decompression Throughput

Figure 3 presents compression and decompression throughput for the XLarge (4 GB) dataset, the most representative size, across all three algorithms, four data types, and four GPUs.

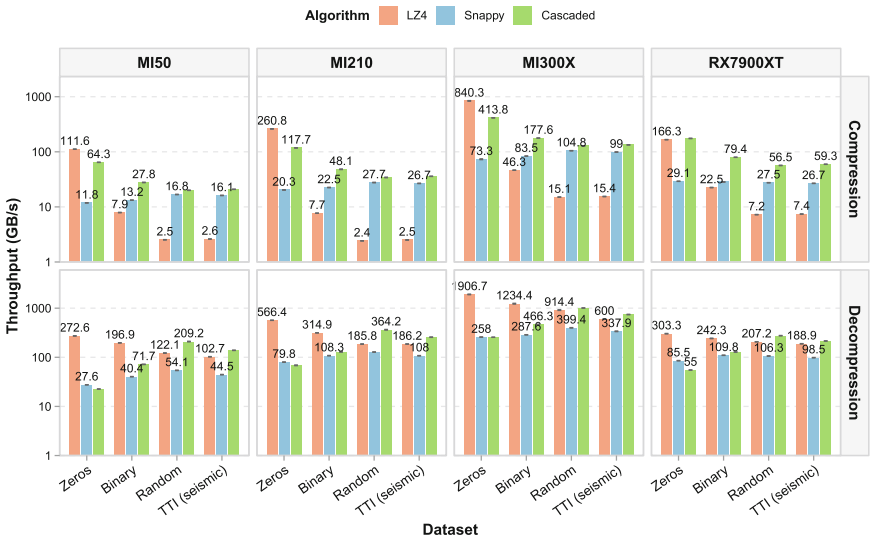


Fig. 3. Compression (top) and decompression (bottom) throughput for the 4 GB dataset across GPUs, algorithms, and data types. Higher is better.

Two fundamental patterns emerge. First, **data compressibility dramatically affects LZ4 throughput**. On the MI300X, LZ4 reaches 840 GB/s compressing zeros but only 15 GB/s for TTI seismic data, a difference exceeding 50×. This sensitivity arises because highly compressible data yields very

small outputs, so the kernel spends less time on hash-table match searches. Cascaded, by contrast, maintains remarkably stable throughput across data types (414 GB/s for zeros vs. 134 GB/s for TTI on the MI300X), due to its fixed delta+RLE+bitpacking pipeline executing the same computational stages regardless of data content.

Second, **compression and decompression favor different architectures**. For compression, the consumer-grade RX 7900 XT matches or exceeds the datacenter MI300X across most datasets. We attribute this to RDNA 3's 96 MB Infinity Cache providing low-latency access to the hash tables (up to 64 KB per warp) that dominate compression's random access pattern, combined with higher clock speeds (up to 2.5 GHz vs. 2.1 GHz). For decompression, the MI300X dominates, reaching 1907 GB/s on zeros. Its 5.3 TB/s HBM3 bandwidth fully serves decompression's sequential streaming pattern of reading compressed data and writing large contiguous output blocks.

The MI210 (CDNA 2) occupies the expected intermediate position between the MI50 and MI300X. For decompression, the MI210 achieves 1.5–3.0× speedup over the MI50 across algorithms, consistent with its 1.6× higher HBM bandwidth (1,638 GB/s vs. 1,024 GB/s). For compression, gains are more modest (1.7× for Snappy and Cascaded on TTI data), while LZ4 compression shows no improvement over the MI50, with both achieving ∼2.5 GB/s for TTI. This confirms that LZ4 compression is bound by hash-table latency rather than memory bandwidth.

6.2 Execution Time Breakdown

While throughput metrics isolate kernel performance, practical deployments must also transfer data between host and device. Figure 4 decomposes total execution time into four phases for the TTI dataset at 4 GB, namely host-to-device (H2D) transfer, compression, decompression, and device-to-host (D2H) transfer.

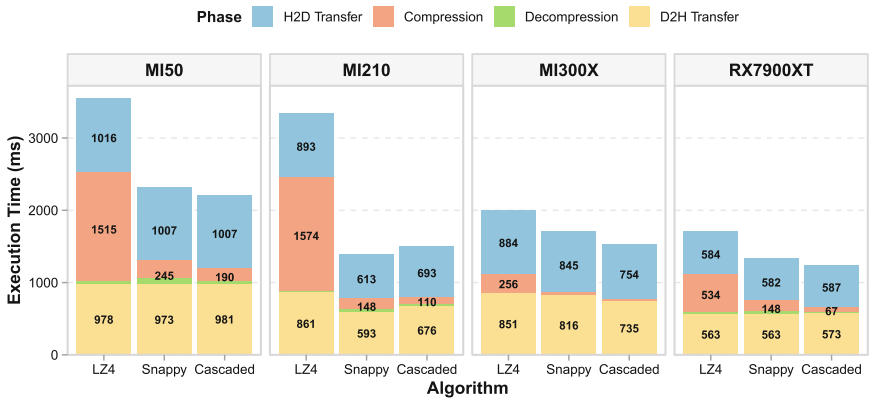


Fig. 4. Execution time breakdown for TTI seismic data (4 GB). Transfer phases (H2D and D2H) dominate total time on all GPUs, especially for Cascaded and Snappy.

Data transfer dominates total execution time across all GPUs. On the MI50, LZ4 compression still represents a significant fraction of total time (1515 ms, ∼43%), but even here, H2D and D2H transfers account for the majority. On the MI300X, faster kernels shift the balance further, with LZ4 compression dropping to 256 ms (∼13% of total), while transfers consume the remaining time. For Cascaded on the RX 7900 XT, compression takes only 67 ms, under 5% of total execution.

This has a practical implication. Optimizing compression kernels alone yields diminishing returns without addressing transfer overhead. Integration approaches that overlap compression with transfers, or that operate directly on GPU-resident data, such as compressing simulation output before writing to storage, would better exploit the kernel-level performance gains.

6.3 Compression Ratio

Table 2 presents compression ratios across algorithms and data types for the 4 GB dataset. Ratios are GPU-independent, depending only on the algorithm and data content.

Table 2. Compression ratio by algorithm and dataset (4 GB input). Ratios are GPU-independent.

Algorithm	TTI (seismic)	Binary	Random	Zeros
LZ4	1.04	3.26	1.00	245
Snappy	1.03	2.96	1.00	21.3
Cascaded	1.03	3.04	1.00	92.0

LZ4 achieves the highest ratios for highly compressible data (245× for zeros), but offers low compression for TTI seismic data (1.04×) and random data (1.00×). Snappy exhibits similar trends with lower peak ratios (21.3× for zeros).

For the TTI scientific dataset, our use case, all three algorithms achieve near-identical compression (1.03–1.04×), indicating that single-precision seismic wavefield data contains limited byte-level redundancy exploitable by lossless methods. This result suggests that, for this class of seismic floating-point data, lossless compression is best viewed as a baseline rather than as a high-impact data-reduction solution, motivating future investigation of error-bounded lossy approaches.

6.4 Throughput Scaling

Figure 5 shows how compression throughput scales with input size for the TTI dataset, the most relevant workload for our target application.

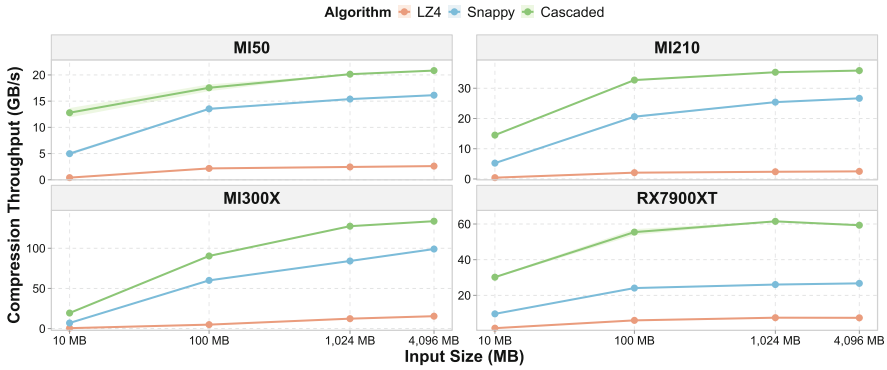


Fig. 5. Compression throughput scaling with input size for TTI seismic data across GPU generations. Throughput increases with input size as parallelism is better utilized.

All algorithms exhibit increasing throughput from 10 MB to 4 GB, reflecting improved GPU utilization at larger input sizes. At 10 MB, kernel launch overhead and low parallelism limit performance across all GPUs. By 1 GB, most configurations approach peak throughput, with only borderline gains at 4 GB.

On the MI300X, Cascaded and Snappy show steep scaling from 100 MB to 1 GB, reflecting the GPU's need for more data to saturate its 304 compute units. The MI50, with 60 compute units, saturates earlier and shows flatter scaling. The MI210, with 104 compute units, exhibits intermediate scaling behavior, saturating at approximately 1 GB input size. LZ4 shows the flattest scaling across all GPUs for TTI data, consistent with its hash-table-bound compression being less sensitive to increased data parallelism.

6.5 Cross-Generation Speedup

Figures 6 and 7 summarize compression and decompression speedup relative to the MI50 baseline across all datasets and input sizes.

The speedup results highlight a fundamental **compression/decompression asymmetry** across architectures. For compression (Fig. 6), the MI300X achieves up to 8× speedup for Snappy and Cascaded at large input sizes, while LZ4 gains are more modest (6×), limited by hash-table latency that does not scale with bandwidth. The RX 7900 XT shows consistent compression speedups across algorithms and sizes (up to 4×), benefiting from its Infinity Cache and higher clock frequency.

For decompression (Fig. 7), the MI300X reaches up to 11× speedup, with LZ4 and Snappy benefiting most from HBM3 bandwidth. The RX 7900 XT's decompression speedup is more modest (up to 3×), constrained by its 800 GB/s GDDR6 bandwidth.

The MI210 demonstrates moderate generational gains, achieving up to 2.4× decompression speedup and 1.7× compression speedup over the MI50 for Snappy

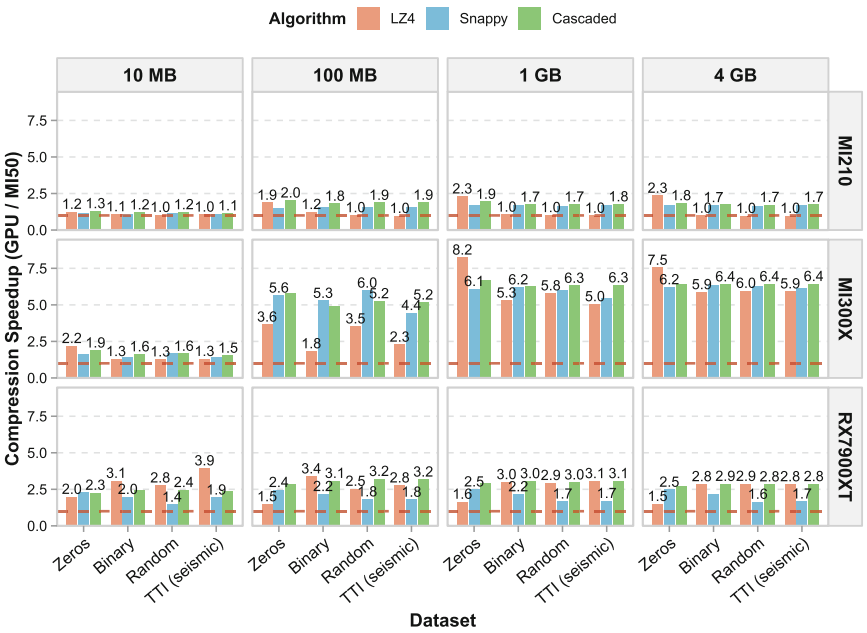


Fig. 6. Speedup relative to MI50 baseline for compression across datasets and input sizes. The dashed line marks MI50 baseline (1×).

and Cascaded on TTI data. The LZ4 compression on the MI210 shows virtually no improvement over the MI50 (∼2.5 GB/s on both), while decompression improves by 1.8×. These gains align with the MI210's 1.6× higher memory bandwidth, suggesting near-linear scaling with bandwidth for decompression-bound workloads.

This asymmetry has practical implications for hardware selection. Applications that primarily decompress data (e.g., analysis pipelines reading compressed datasets) benefit from HBM-equipped CDNA hardware. At the same time, workloads dominated by compression may find RDNA hardware competitive or benefit from architecture-specific kernel optimizations on CDNA.

7 Discussion

The compression-decompression asymmetry reported in Sect. 6 reflects a fundamental difference in memory access patterns. Compression relies on small, random hash-table reads that benefit from RDNA 3's Infinity Cache, while decompression follows a sequential streaming pattern that favors raw HBM bandwidth. The MI210's decompression scaling closely tracks its bandwidth advantage, whereas LZ4 compression remains latency-bound across all CDNA generations. These findings suggest that hardware selection should consider the

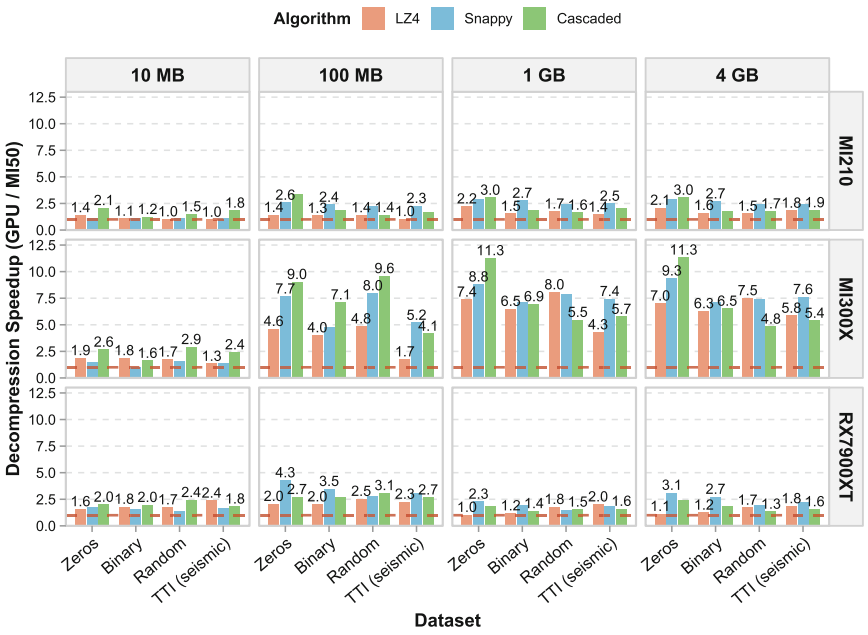


Fig. 7. Speedup relative to MI50 baseline for decompression across datasets and input sizes. The dashed line marks MI50 baseline (1×).

compression-to-decompression ratio of the target workload, not just peak bandwidth or compute specifications.

Despite the current optimization headroom, GPU compression on AMD platforms already provides tangible value in I/O-bound workflows. For instance, writing a 4 GB checkpoint to a parallel file system at 10 GB/s takes 0.4 s without compression. For highly compressible data such as sparse matrices or partially converged fields (compression ratios of 3× or higher, as observed for the binary dataset), the reduction in I/O volume can significantly reduce checkpoint time. For decompression-dominated workflows, the MI300X's throughput (Fig. 3) means that storage bandwidth is almost always the bottleneck, so compression ratio translates directly to end-to-end speedup.

Our characterization also reveals concrete optimization opportunities. A CDNA-optimized LZ4 kernel could redesign the current one-warp-per-chunk model to exploit full 64-thread wavefronts, and move the 64 KB hash table into the Local Data Share (LDS), replacing global-memory lookups with single-cycle LDS reads and directly addressing the latency bottleneck identified in Sect. 6. For floating-point scientific data, byte-shuffling or XOR-based pre-conditioning could expose redundancy invisible to general-purpose compressors, potentially improving the low lossless ratios observed for seismic data (Table 2). Where application constraints permit bounded error, error-bounded lossy compressors such as SZ [13] and ZFP [21] routinely achieve 10–100× ratios on similar data [27],

making their integration a natural next step. Finally, a unified cross-vendor API that compiles to optimized backends on both AMD and NVIDIA platforms would broaden the practical impact of this work.

This study has limitations. The HIP implementation prioritizes portability and correctness over AMD-specific tuning, so results represent a practical baseline rather than peak achievable performance. Throughput measurements isolate kernel execution time, with transfer costs analyzed separately. The evaluation covers three lossless algorithms with fixed chunking, and additional algorithms and tuning configurations remain for future work. Finally, compression behavior is inherently data-dependent, so that different scientific workloads may exhibit different trade-offs.

8 Conclusion

This paper presented a cross-generational characterization of lossless GPU data compression across AMD CDNA and RDNA architectures, evaluating LZ4, Snappy, and Cascaded on four GPUs spanning the MI50, MI210, MI300X, and RX 7900 XT. Beyond presenting raw benchmark results, the study examined how architectural differences shape compression and decompression behavior and established an AMD-compatible baseline through a functional CUDA-to-HIP port.

The results show a clear compression–decompression asymmetry across architectures. RDNA 3 is competitive for compression workloads with irregular memory-access behavior, whereas CDNA 3 strongly favors decompression through substantially higher HBM bandwidth, with the MI300X reaching up to 11× the MI50 baseline. At the same time, the execution-time breakdown shows that PCIe transfer overhead dominates end-to-end time on all evaluated platforms, indicating that the full benefits of GPU compression are most likely to arise when compression is integrated into GPU-resident workflows rather than used as an isolated kernel optimization.

For realistic seismic floating-point data, all evaluated lossless methods achieve only modest compression ratios (1.03–1.04×). This does not weaken the relevance of the study; rather, it clarifies the role of lossless compression as a portability and systems baseline and points to a more promising next step for this application domain: error-bounded lossy compression. Future work will therefore focus on AMD-specific kernel optimizations, broader algorithm coverage, and integration with portable data-reduction pipelines for heterogeneous HPC environments.

Acknowledgments. This study was partially supported by Petrobras grant no. 2020/00182-5, by CNPq/MCTI/FNDCT - Universal under grants 408755/2025-3, and by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. This effort was also partially supported by the U.S. Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research (ASCR) under contract number DE-AC02-05CH11231 with LBNL.

Some experiments in this work used the PCAD infrastructure, https://pcad.inf.ufrgs.br/, at INF/UFRGS. Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).

References

1. Alkhalifah, T.: An acoustic wave equation for anisotropic media. Geophysics **65**(4), 1239–1250 (2000)
2. AMD: CDNA 2 architecture white paper. Technical report, Advanced Micro Devices (2021). https://www.amd.com/content/dam/amd/en/documents/instinct-business-docs/white-papers/amd-cdna2-white-paper.pdf
3. AMD: HIP: Heterogeneous-compute interface for portability (2025). https://github.com/ROCm/HIP. Accessed 19 Feb 2026
4. AMD ROCm: hipCOMP-core: HIP port of compression algorithms (2025). https://github.com/ROCm/hipCOMP-core. Accessed 20 Feb 2026
5. Balouek, D., et al.: Adding virtualization capabilities to the Grid'5000 testbed. In: Ivanov, I.I., van Sinderen, M., Leymann, F., Shan, T. (eds.) CLOSER 2012. CCIS, vol. 367, pp. 3–20. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-04519-1_1
6. Burtchell, B.A., Burtscher, M.: Characterizing the performance of parallel data-compression algorithms across compilers and GPUs. In: Proceedings of the SC '25 Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 269–278. ACM (2025). https://doi.org/10.1145/3731599.3767369
7. Che, S., et al.: Rodinia: a benchmark suite for heterogeneous computing. In: IEEE International Symposium on Workload Characterization (IISWC), pp. 44–54 (2009). https://doi.org/10.1109/IISWC.2009.5306797
8. Chen, J., et al.: HPDR: High-performance portable scientific data reduction framework. In: 2025 IEEE IPDPS, pp. 1104–1116. IEEE (2025). https://doi.org/10.1109/IPDPS64566.2025.00101
9. Chen, X., et al.: Fcbench: cross-domain benchmarking of lossless compression for floating-point data. Proc. VLDB Endowment **17**(6), 1418–1431 (2024). https://doi.org/10.14778/3648160.3648180
10. Collet, Y.: LZ4 - extremely fast compression (2025). https://lz4.github.io/lz4/. Accessed 20 Feb 2026
11. CSC – IT Center for Science: LUMI supercomputer (2025). https://www.lumi-supercomputer.eu/. Accessed 25 Feb 2025
12. Danalis, A., et al.: The scalable heterogeneous computing (SHOC) benchmark suite. In: Workshop on General-Purpose Computation on Graphics Processing Units (GPGPU), pp. 63–74 (2010). https://doi.org/10.1145/1735688.1735702
13. Di, S., Cappello, F.: Fast error-bounded lossy HPC data compression with SZ. In: IEEE IPDPS, pp. 730–739 (2016). https://doi.org/10.1109/IPDPS.2016.11
14. Fletcher, R.P., Du, X., Fowler, P.J.: Reverse time migration in tilted transversely isotropic (TTI) media. Geophysics **74**(6), WCA179–WCA187 (2009)
15. Fomel, S., Sava, P., Vlad, I., Liu, Y., Bashkardin, V.: Madagascar: open-source software project for multidimensional data analysis and reproducible computational experiments (2013)

16. Google: Snappy: a fast compressor/decompressor (2025). https://github.com/google/snappy. Accessed 20 Feb 2026
17. Huang, Y., Di, S., Li, G., Cappello, F.: cuSZp2: a GPU lossy compressor with extreme throughput and optimized compression ratio. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE (2024). https://doi.org/10.1109/SC41406.2024.00021
18. Huang, Y., Di, S., Yu, X., Li, G., Cappello, F.: cuSZp: an ultra-fast GPU error-bounded lossy compression framework with optimized end-to-end performance. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1–13 (2023)
19. Knorr, F., Thoman, P., Fahringer, T.: ndzip-GPU: efficient lossless compression of scientific floating-point data on GPUs. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1–14. ACM (2021). https://doi.org/10.1145/3458817.3476224
20. Kurtzer, G.M., Sochat, V., Bauer, M.W.: Singularity: scientific containers for mobility of compute. PLoS ONE **12**(5), e0177459 (2017). https://doi.org/10.1371/journal.pone.0177459
21. Lindstrom, P.: Fixed-rate compressed floating-point arrays. IEEE TVCG **20**(12), 2674–2683 (2014). https://doi.org/10.1109/TVCG.2014.2346458
22. Liu, Q., et al.: Hello ADIOS: the challenges and lessons of developing leadership class I/O frameworks. Concurrency Comput. Pract. Experience **26**(7), 1453–1473 (2014). https://doi.org/10.1002/cpe.3125
23. Nicolae, B., Moody, A., Gonsiorowski, E., Mohror, K., Cappello, F.: VeloC: towards high performance adaptive asynchronous checkpointing at large scale. In: IEEE IPDPS, pp. 911–920 (2019). https://doi.org/10.1109/IPDPS.2019.00099
24. NVIDIA: Cascaded compression — nvCOMP documentation (2024). https://docs.nvidia.com/cuda/nvcomp/cascaded.html. Accessed 19 Mar 2025
25. NVIDIA Corporation: CUB: Cooperative primitives for CUDA C++ (2025). https://nvidia.github.io/cccl/cub/. Accessed 22 Feb 2026
26. NVIDIA Corporation: nvcomp: High-speed data compression library for nvidia GPUs (2025). https://developer.nvidia.com/nvcomp. Accessed 20 Feb 2026
27. Tian, J., et al.: cuSZ: an efficient GPU-based error-bounded lossy compression framework for scientific data. In: ACM PACT, pp. 3–15 (2020). https://doi.org/10.1145/3410463.3414624
28. TOP500: TOP500 list – November 2025. https://top500.org/lists/top500/2025/11/ (2025), 66th edn, published November, 2025. Accessed 20 Mar 2026
29. Zhang, B., et al.: FZ-GPU: a fast and high-ratio lossy compressor for scientific computing applications on GPUs. In: Proceedings of the 32nd International Symposium on High-Performance Parallel and Distributed Computing, pp. 129–142. ACM (2023). https://doi.org/10.1145/3588195.3592994
30. Zhang, B., et al.: GPULZ: Optimizing LZSS lossless compression for multi-byte data on modern GPUs. In: Proceedings of the 37th ACM International Conference on Supercomputing, pp. 348–359. ACM (2023). https://doi.org/10.1145/3577193.3593706

A Systematic Literature Review of Network-on-Chip Architectures: Topologies, Routing, Reliability, and Emerging Technologies

Asrani Lit(✉), Rohana Sapawi, Shamsiah Suhaili, Sharifah Masniah Wan Masra, and Kuryati Kipli

Department of Electrical Engineering, Faculty of Engineering, Universiti Malaysia Sarawak, 94300 Kota Samarahan, Malaysia
{lasrani,srohana,sushamsiah,wmmasnia,kkuryati}@unimas.my

Abstract. The increasing core count of contemporary systems-on-chip (SoCs) has driven a fundamental shift from shared bus architectures to scalable on-chip interconnection networks. Network-on-Chip (NoC) has emerged as the de facto paradigm for providing high-bandwidth, low-latency, and energy-efficient communication among heterogeneous processing elements. Over two decades of research have produced a rich body of work on NoC topologies, routing algorithms, flow-control schemes, quality-of-service (QoS) mechanisms, reliability enhancement, and emerging technologies such as wireless, optical, and three-dimensional (3D) NoCs. This paper presents a systematic literature review (SLR) of Network-on-Chip architectures from 2002 to 2025 following a transparent and reproducible methodology. We formulate five research questions covering research themes, design challenges, technological trends, evaluation methodologies, and open problems. A structured search over major digital libraries is conducted, and primary studies are classified along eight dimensions: topology design, routing algorithms, flow control and buffering, QoS and real-time guarantees, fault tolerance and reliability, energy and thermal awareness, heterogeneous and emerging NoCs, and machine-learning-driven NoC optimization. The review highlights a progression from regular 2D meshes with deterministic routing to heterogeneous hybrid interconnects with adaptive, often learning-based routing and cross-layer co-optimization. We identify research gaps in holistic security, standardized benchmarks for AI-centric workloads, integrated ML frameworks for NoC design space exploration, and cross-technology co-design in hybrid wired–wireless–optical systems. The findings provide a consolidated reference for researchers and practitioners, and outline promising directions for next-generation NoC research.

A. Lit, R. Sapawi, S. Suhaili, S. M. Wan Masra and K. Kipli—Contributed equally to this work.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 375–390, 2026.
https://doi.org/10.1007/978-3-032-30491-9_24

Keywords: Network-on-Chip (NoC) · routing · topology · reliability · wireless NoC · optical NoC · 3D NoC · systematic literature review (SLR) · machine learning

1 Introduction

The transition to manycore and heterogeneous system-on-chip (SoC) platforms has imposed stringent demands on the on-chip interconnection fabric. Shared buses and crossbars, once adequate for a small number of cores, fail to scale gracefully due to bandwidth limitations, complex arbitration, and prohibitive power and area overheads [1]. Network-on-Chip (NoC) emerged as a scalable and modular on-chip communication paradigm by leveraging concepts from off-chip interconnection networks and packet-switched communication [2].

NoC-based interconnects have been widely explored in academic research and adopted in industrial designs, particularly in multicore CPUs, GPUs, and heterogeneous SoCs. Research has spanned a broad spectrum, including topology design [3,4], routing algorithms [5,6], flow control and buffering [7,8], quality-of-service (QoS) mechanisms [9], reliability and fault tolerance [10,11], energy and thermal management [12,13], and emerging technologies such as wireless NoC (WiNoC) [14], optical NoC (ONoC) [15], and 3D NoC [16,17]. More recently, machine learning (ML) has been applied to NoC design space exploration and adaptive control [18,19].

Despite the breadth of work, the literature remains fragmented across subtopics and technologies. Several surveys exist [20–22], but many predate the latest developments in hybrid wired–wireless–optical interconnects, ML-assisted optimization, and AI-centric workloads. Furthermore, a comprehensive and systematically conducted review, following established SLR guidelines, is still lacking for Network-on-Chip.

This paper addresses this gap by providing a systematic literature review of NoC architectures and related design methodologies over the period 2002–2024. We follow a structured process, including research question formulation, search strategy definition, study selection, quality assessment, data extraction, and thematic synthesis.

1.1 Research Questions

These research questions are formulated to provide a structured and comprehensive investigation of the existing body of literature. They aim to identify prevailing research trends, dominant methodologies, and technological advancements reported in prior studies, while also uncovering existing limitations and unresolved challenges. Collectively, the RQs serve as the foundation for the review protocol, ensuring a transparent, reproducible, and objective synthesis of evidence aligned with the objectives of this study. The systematic literature review is guided by the following research questions (RQs):

- **RQ1:** What are the dominant research themes and trends in Network-on-Chip architectures?
- **RQ2:** Which NoC design challenges (latency, throughput, energy, reliability, scalability, QoS) have been most extensively studied?
- **RQ3:** How have emerging NoC technologies (wireless, optical, 3D, hybrid) evolved in the literature?
- **RQ4:** What methodologies, tools, and benchmarks are used to evaluate NoC proposals?
- **RQ5:** What research gaps and future directions can be identified from the existing body of work?

1.2 Contributions

The main contributions of this paper are:

- A systematic and reproducible survey protocol for NoC literature aligned with general SLR guidelines.
- A taxonomy of NoC research organized into eight dimensions: topology, routing, flow control, QoS, reliability, energy/thermal, emerging technologies, and ML-driven optimization.
- Comparative analysis of prominent topologies, routing algorithms, and technology platforms, highlighting trade-offs among area, latency, bandwidth, and energy.
- A synthesis of evaluation methodologies, including simulation frameworks (e.g., BookSim, Noxim, Garnet), analytical models, and experimental platforms.
- Identification of research gaps and promising directions for next-generation NoC systems, including security, AI-driven workloads, and hybrid interconnects.

1.3 Paper Organization

The remainder of this paper is organized as follows. Section 2 presents background on NoC architectures. Section 3 details the SLR methodology. Section 4 presents the classification and thematic synthesis of the selected studies. Section 5 discusses key findings and research gaps. Section 6 outlines future research directions, and Sect. 7 concludes the paper.

2 Background on Network-on-Chip

2.1 Basic NoC Architecture

A typical NoC consists of routers interconnected via links in a chosen topology. Processing elements (PEs), such as CPU cores, accelerators, or memory controllers, connect to the NoC through network interfaces. Packets are segmented

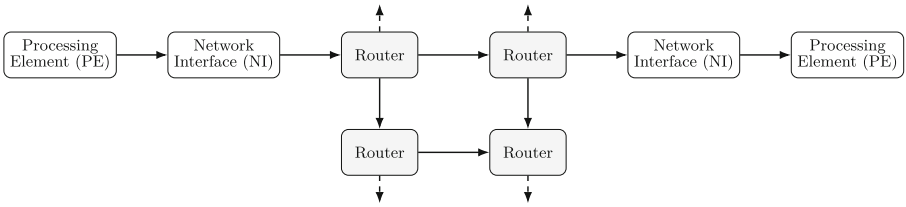


Fig. 1. Basic Network-on-Chip (NoC) architecture showing processing elements (PEs) connected via network interfaces (NIs) to a router-based NoC fabric.

into flits and forwarded across routers using a specific flow-control scheme, often wormhole switching [1]. Routing algorithms determine the path that packets take from sources to destinations, subject to constraints such as deadlock freedom, minimal path length, and congestion avoidance [5].

Figure 1 shows a basic Network-on-Chip architecture comprises processing elements interconnected through network interfaces, routers, and physical links arranged in a regular topology such as a two-dimensional mesh. Data communication is packet-based, with routers performing routing, buffering, and arbitration functions to ensure efficient and scalable on-chip communication. By leveraging networking principles, NoCs overcome the scalability and performance limitations of traditional bus-based interconnects and have become a foundational communication fabric in modern many-core systems.

2.2 Design Objectives and Constraints

As the number of processing elements in modern systems-on-chip continues to increase, efficient on-chip communication has become a critical design challenge. Traditional bus-based interconnects suffer from limited scalability, high contention, and excessive power consumption, motivating the adoption of Network-on-Chip architectures as a structured and scalable communication fabric. Key NoC design objectives include:

- **Performance:** Low average latency, high throughput under realistic workloads.
- **Energy Efficiency:** Minimization of dynamic and static power consumed by routers and links.
- **Scalability:** Support for increasing numbers of cores without disproportionate cost.
- **Reliability:** Robustness against transient and permanent faults.
- **QoS and Real-Time Guarantees:** Predictable performance for latency-critical flows.
- **Technology Constraints:** Impact of process scaling, 3D stacking, and novel interconnect technologies.

These objectives are often conflicting, leading to complex trade-offs that drive research into novel topologies, adaptive routing, power management, and cross-layer optimization.

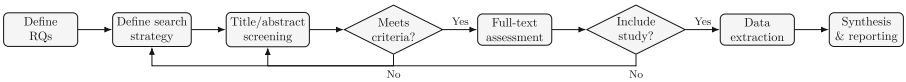


Fig. 2. Systematic literature review (SLR) protocol followed in this work.

3 Systematic Review Methodology

3.1 Protocol and Guidelines

The SLR follows general guidelines such as those by Kitchenham for software engineering SLRs, adapted to hardware and NoC research. The protocol includes the following stages: (i) planning, (ii) conducting, and (iii) reporting. Figure 2 illustrates the overall process followed in this work.

3.2 Search Strategy

Digital libraries searched include IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink, and Scopus. The primary search string was constructed using combinations of keywords:

- "network-on-chip" OR "NoC"
- AND ("routing" OR "topology" OR "flow control" OR "QoS" OR "reliability" OR "fault tolerance" OR "wireless" OR "optical" OR "3D")

The search period was set from 2002, corresponding roughly to the emergence of NoC as a research area [2], to the end of 2025. Additional references were obtained via backward and forward snowballing.

3.3 Inclusion and Exclusion Criteria

To ensure the relevance, quality, and consistency of the selected studies, clear inclusion and exclusion criteria were defined prior to the screening process.

Inclusion criteria:

- Peer-reviewed journal and conference papers.
- Primary studies presenting new NoC architectures, algorithms, or analysis.
- Papers focusing on topology, routing, flow control, QoS, reliability, energy/thermal, or emerging NoC technologies.

Exclusion criteria:

- Non-peer-reviewed reports, theses, and patents.
- Works focusing solely on off-chip interconnects or FPGA interconnects without NoC context.
- Papers with insufficient methodological detail or without quantitative or formal analysis.

3.4 Quality Assessment and Data Extraction

Each candidate paper was evaluated using a quality checklist addressing clarity of objectives, soundness of methodology, adequacy of evaluation, and completeness of reporting. For each included study, we extracted:

- Bibliographic data (authors, year, venue).
- NoC dimension(s) addressed (topology, routing, etc.).
- Target technology (2D, 3D, wireless, optical, hybrid).
- Methodology (simulation, analytical modeling, emulation, prototype).
- Evaluation metrics (latency, throughput, energy, area, reliability, QoS).

3.5 Classification Scheme

To systematically analyze and compare the selected studies, the extracted literature was classified into eight primary research categories based on their main technical focus and contribution.

1. **Topology design:** Studies addressing the structural organization of NoC nodes and links, including regular and irregular topologies, with the aim of improving scalability, latency, bandwidth, and physical layout efficiency.
2. **Routing algorithms:** Works proposing deterministic, adaptive, or hybrid routing strategies to optimize path selection, reduce congestion, and ensure deadlock- and livelock-free communication.
3. **Flow control and buffering:** Research focusing on packet switching techniques, buffer management, virtual channels, and congestion control mechanisms to enhance throughput and resource utilization.
4. **QoS and real-time support:** Studies that incorporate quality-of-service guarantees, traffic prioritization, and real-time constraints to support heterogeneous and latency-sensitive applications.
5. **Fault tolerance and reliability:** Contributions addressing transient and permanent faults through error detection, recovery mechanisms, redundant routing, and reliability-aware architectures.
6. **Energy- and thermal-aware design:** Works aiming to minimize power consumption and thermal hotspots via energy-efficient routing, voltage/frequency scaling, and thermal-aware resource management.
7. **Emerging technologies (wireless, optical, 3D, hybrid):** Studies exploring non-traditional NoC implementations, including wireless, photonic, three-dimensional, and hybrid interconnects to overcome the limitations of conventional electrical NoCs.
8. **Machine-learning-driven NoC optimization:** Research leveraging machine learning techniques for adaptive routing, congestion prediction, fault management, and design-space exploration.

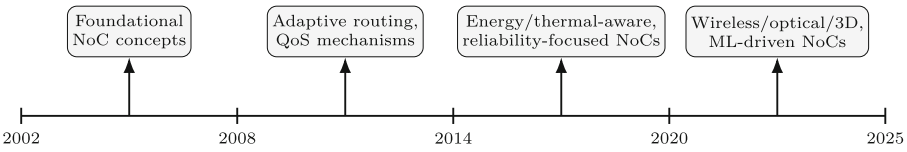


Fig. 3. Conceptual timeline of major research waves in Network-on-Chip (NoC) literature.

4 Results and Thematic Synthesis

4.1 Publication Trends

Figure 3 provides a high-level timeline of key developments in NoC research. Early work focused on regular 2D meshes and deterministic routing [2,20]; the subsequent period introduced topology variants, adaptive routing, and QoS mechanisms [5,9]. The last decade has seen rapid growth in emerging technologies (WiNoC, ONoC, 3D NoC) and ML-assisted design [14,15,18].

4.2 Taxonomy of NoC Research

To provide a structured overview of the research landscape, this study organizes the existing Network-on-Chip literature into a comprehensive taxonomy. The taxonomy captures the major design dimensions, architectural considerations, and optimization objectives that have emerged over time in NoC research. Figure 4 presents the taxonomy used in this review, illustrating the hierarchical classification of studies across core architectural components, communication mechanisms, system-level requirements, and emerging technologies. This taxonomy serves as a unifying framework for analyzing trends, identifying research gaps, and comparing representative works across different NoC domains.

4.3 Topologies

Table 1 summarizes representative NoC topologies and their key characteristics. Mesh remains the most prevalent due to its layout regularity and implementation simplicity [2,20]. Variants such as torus, flattened butterfly, and fat-tree offer reduced hop count at the cost of increased wiring complexity [4]. 3D topologies exploit through-silicon vias (TSVs) to reduce communication distance [16,17].

4.4 Routing Algorithms

Routing algorithms significantly influence NoC performance and robustness. We group routing schemes into deterministic, partially adaptive, fully adaptive, and fault-tolerant categories. Table 2 summarizes representative routing families.

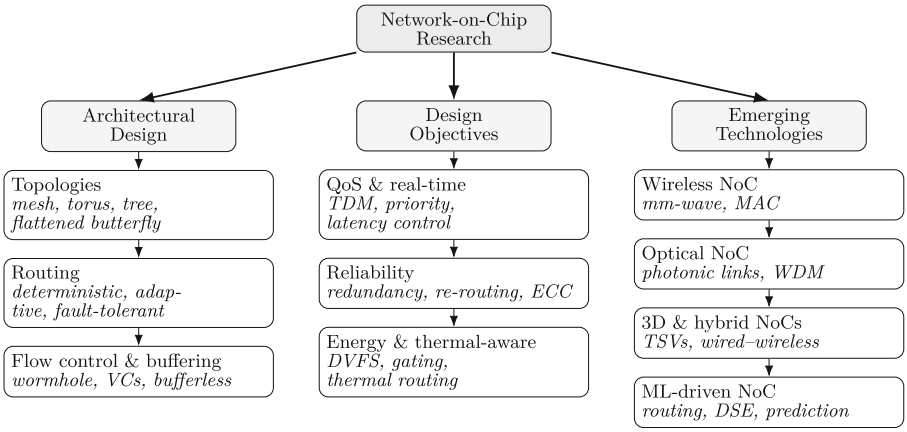


Fig. 4. Taxonomy of Network-on-Chip research.

Table 1. Representative NoC topologies and qualitative comparison.

Topology	Examples	Average hop count	Layout regularity	Routing complexity	Typical applications
2D Mesh	[2,20]	Medium	High	Low (e.g., XY)	General-purpose manycore SoCs
2D Torus	[5]	Low–medium	Medium	Low–medium	Latency-aware systems
Fat-tree / Tree	[3,23]	Low	Medium–low	Medium–high	Hierarchical, clustered SoCs
Flattened Butterfly	[4]	Low	Low	Medium–high	High-radix switches, HPC-style SoCs
Custom / Irregular	[24]	Application-dependent	Low	High	Application-specific NoCs (ASNoCs)
3D Mesh / Torus	[16,17]	Low	Medium	Medium	3D-stacked manycore systems
Hybrid Wired–Wireless	[14,25]	Very low (for long-range flows)	Medium	High	Latency-critical, large-scale NoCs

Table 2. Taxonomy of NoC routing algorithms.

Category	Representative algorithms	Key properties	Typical benefits / limitations
Deterministic	XY, YX, dimension-order [5]	Fixed path for each source–destination pair	Simple hardware, predictable latency; limited congestion handling
Turn-model based	West-first, Negative-first, Odd–Even [26,27]	Restricted turns to avoid cycles	Deadlock-free with modest adaptivity
Partially adaptive	DyAD, minimal adaptive [6]	Adaptive within constraints	Better congestion handling; slightly more complex hardware
Fully adaptive	RCA, OFAR, local congestion-aware [12,28]	High path flexibility	Best performance under load; requires VCs and careful deadlock handling
Fault-tolerant	FTR, LBDR, reconfiguration-based [11,29]	Avoids faulty links/nodes	Improved reliability; additional area and complexity
Wireless/optical-aware	WiNoC-aware and ONoC-aware routing [14,15]	Exploit long-range fast links	Reduced hop count and latency; requires technology-specific design
ML-driven	Reinforcement learning, predictive routing [18,19]	Use ML models to select paths or predict congestion	Potentially higher performance; overhead of training and hardware support

4.5 Flow Control and Buffering

Most NoCs adopt wormhole flow control with virtual channels (VCs) to balance buffer usage and performance [1,8]. Bufferless (deflection) routing is explored to reduce leakage power in deeply scaled technologies [30]. Studies investigate VC allocation, router micro-architecture, and credit-based or on/off flow control mechanisms to improve throughput and energy efficiency.

4.6 QoS and Real-Time Support

QoS mechanisms provide differentiated services for traffic classes with distinct latency, jitter, and bandwidth requirements. Approaches include TDM-based arbitration, priority-based schedulers, and rate control [9,20]. Real-time NoCs for automotive and avionics systems extend these mechanisms with worst-case latency analysis and formal verification [31].

4.7 Fault Tolerance and Reliability

As feature sizes shrink, NoCs become more susceptible to soft errors, wear-out, and TSV failures in 3D stacks. Fault-tolerant techniques include redundancy, fault-aware routing, reconfigurable topologies, and error control coding [10,11,29]. Many works integrate detection and recovery mechanisms to maintain connectivity in the presence of multiple simultaneous faults.

4.8 Energy and Thermal-Aware NoC

Energy consumption arises from dynamic switching in routers and links as well as static leakage. Thermal hotspots degrade reliability and performance. Works in this category propose DVFS, power-gating, clock-gating, and thermal-aware routing [12,13]. In 3D NoCs, vertical communication and hotspot mitigation are central issues [17].

Table 3. Comparison of emerging NoC technologies.

Technology	Key advantages	Main challenges	Typical use-cases	Representative works
Wireless NoC (WiNoC)	Long-range shortcuts, reduced hop count, global broadcast	Interference, antenna design, MAC protocol design, energy of transceivers	Large-scale NoCs, latency-critical traffic	[14,25]
Optical NoC (ONoC)	Very high bandwidth, low latency over long distances	Laser power, thermal sensitivity, fabrication complexity	High-performance computing, bandwidth-intensive SoCs	[15,32]
3D NoC	Shorter distances, reduced footprint, improved performance	TSV reliability, thermal management, manufacturing cost	3D stacked processor-memory systems	[16,17]
ML-driven NoC	Adaptive optimization of routing, power, and resources	Training data requirements, hardware cost of ML models	Dynamic and heterogeneous workloads, AI accelerators	[18,19,33]

4.9 Emerging Technologies and ML-Driven NoCs

Wireless NoC (WiNoC) uses on-chip mm-wave transceivers to create long-range shortcuts that reduce hop count and average latency [14,25]. Optical NoC (ONoC) leverages silicon photonics and wavelength-division multiplexing (WDM) to provide high-bandwidth, low-latency links [15,32]. 3D NoCs exploit TSVs to stack multiple logic and memory layers [16]. ML-driven NoC work uses supervised and reinforcement learning for traffic prediction, adaptive routing, admission control, and design space exploration [18,19,33]. Table 3 summarizes key characteristics of these emerging platforms.

5 Discussion

5.1 Shift from Regular to Hybrid Topologies

While the 2D mesh topology remains dominant in both academic prototypes and commercial manycore processors due to its layout regularity and ease of implementation, the literature reveals a gradual shift toward more complex and hybrid NoC topologies. These include architectures augmented with express links, hierarchical hubs, wireless shortcuts, 3D stacked vertical links, or photonic interconnects to reduce average hop count and communication latency [34,35]. Such hybrid designs are particularly attractive for large-scale and heterogeneous systems, where communication locality and traffic imbalance are pronounced [36,37].

Despite these advantages, industrial adoption still favors regular structures because of their predictable timing behavior, simpler physical design, and lower verification overhead. As a result, hybrid topologies are often explored in constrained or partially deployed forms, highlighting an ongoing trade-off between performance scalability and design complexity.

5.2 From Deterministic to Learning-Based Routing

Routing strategies in NoCs have evolved from deterministic dimension-order schemes, such as XY routing, toward adaptive and congestion-aware algorithms capable of improving throughput and fault tolerance under non-uniform traffic conditions [5]. More recently, machine learning (ML)-based routing approaches have gained increasing attention, replacing hand-crafted heuristics with data-driven policies that learn from runtime traffic patterns and network states [18,19,38].

These learning-based methods, often relying on reinforcement learning or deep neural networks, have demonstrated promising gains in latency reduction and load balancing. However, several challenges hinder their widespread adoption, including hardware overhead, training stability, explainability of routing decisions, and formal verification of correctness and deadlock freedom [39,40]. Addressing these concerns remains a critical research direction for practical deployment.

5.3 Growing Focus on Reliability, Energy, and Thermal Issues

With continued technology scaling and increased integration density, reliability, energy efficiency, and thermal management have emerged as first-order design constraints in NoC architectures. Phenomena such as process variation, aging, soft errors, and thermal hotspots can significantly degrade network performance and lifetime [41,42]. Consequently, recent studies increasingly adopt cross-layer approaches that jointly consider micro-architectural techniques, routing strategies, and system-level management policies [43,44].

This trend reflects a broader shift away from isolated optimizations toward holistic NoC design frameworks that balance performance, power, and reliability objectives simultaneously. Such approaches are especially relevant for 3D-integrated and heterogeneous systems, where thermal coupling and reliability concerns are exacerbated.

5.4 Evaluation Methodologies and Benchmarks

Simulation-based evaluation remains the dominant methodology for NoC research, with widely used frameworks including BookSim, Garnet (Gem5), and Noxim. Most studies rely on synthetic traffic patterns such as uniform random, transpose, or hotspot traffic to stress-test network behavior. A smaller subset employs application-driven traces derived from benchmark suites such as PARSEC, SPLASH-2, or custom workloads.

However, there is still a notable lack of standardized evaluation suites specifically targeting modern AI workloads and heterogeneous accelerator-rich systems. As NoCs increasingly serve as the communication backbone for DNN accelerators and dataflow architectures, recent works have emphasized the need for benchmark traces that capture multicast, bursty, and synchronization-heavy traffic patterns characteristic of AI applications [45–47]. The absence of such standardized benchmarks continues to limit fair comparison and reproducibility across studies.

6 Future Research Directions

The systematic analysis suggests several promising avenues for future work:

6.1 Security and Trust in NoC

Security has only recently begun to receive systematic attention in NoC research, despite the growing deployment of manycore systems in safety- and mission-critical applications. Potential threats include side-channel attacks, denial-of-service via congestion, malicious routers, and hardware Trojans embedded at design or fabrication stages [48,49]. Future work requires robust, low-overhead mechanisms for intrusion detection, secure routing, authentication, and traffic obfuscation that can be integrated without significantly degrading latency or energy efficiency [50,51].

6.2 Standardized AI-Centric Benchmarks

With the proliferation of AI accelerators and dataflow architectures, NoC workloads are increasingly dominated by regular, bursty, and multicast traffic patterns arising from deep neural network (DNN) computations [52,53]. However, many existing evaluations still rely on synthetic or legacy benchmarks that do not accurately capture these characteristics. Community-agreed benchmarks, traffic traces, and evaluation methodologies for AI-centric NoCs would enable more realistic, reproducible, and comparable performance assessments across studies [45,46].

6.3 Integrated ML Frameworks for NoC

Machine learning techniques have been widely explored for adaptive routing, congestion prediction, dynamic voltage and frequency scaling (DVFS), and fault management in NoCs [38,54]. However, these approaches are often applied in an ad hoc manner, focusing on isolated optimization objectives. Integrated ML frameworks that support end-to-end training, deployment, and hardware mapping of ML models for NoC control would accelerate practical adoption. In particular, the co-design of ML algorithms and lightweight hardware accelerators for in-router inference remains an open research challenge [39,55].

6.4 Cross-Technology and Cross-Layer Co-Design

Emerging hybrid wired–wireless–optical NoCs in advanced 2.5D and 3D packages introduce a complex multi-physics design space involving signal integrity, thermal coupling, reliability, and manufacturing cost [34,35]. Addressing these challenges requires holistic co-optimization across circuit, architecture, and application layers. Future research should emphasize cross-layer design methodologies, formal modeling techniques, and scalable design space exploration tools capable of managing this growing complexity [36,56].

6.5 Sustainability and Lifetime Management

Energy-efficient NoC design plays a critical role in improving overall system sustainability and reducing operational carbon footprint [43]. Beyond energy, lifetime-aware routing, wear-leveling, and reliability management techniques can extend the usable lifetime of links, routers, and through-silicon vias (TSVs), particularly in deeply scaled and 3D-integrated technologies [41,42]. Such approaches contribute to reduced electronic waste and improved long-term system cost-effectiveness.

7 Conclusion

This paper has presented a systematic literature review of Network-on-Chip architectures spanning more than two decades of research. Using a structured

protocol, we synthesized contributions across topology design, routing algorithms, flow-control mechanisms, QoS support, reliability and fault tolerance, energy- and thermal-aware techniques, emerging NoC technologies, and ML-driven NoC optimization. We provided taxonomies, comparative tables, and conceptual timelines that highlight the evolution of the field. The identified research gaps and future directions particularly in security, AI-centric benchmarking, ML integration, and cross-technology co-design offer a roadmap for the next generation of NoC research and practice.

Acknowledgement. The authors gratefully acknowledge the support received from the FRGS-EC/1/2024/TK07/UNIMAS/03/1 Malaysia under grant number NAT/F02/FRGS-EC/86558/2024. Additionally, we would like to thank the Faculty of Engineering at Universiti Malaysia Sarawak for their support with the Antenna Laboratory facilities.

References

1. Dally, W.J., Towles, B.: Route packets, not wires: on-chip interconnection networks. In: Proceedings of the Design Automation Conference, pp. 684–689 (2001)
2. Benini, L., De Micheli, G.: Networks on chips: a new SoC paradigm. IEEE Comput. **35**(1), 70–78 (2002)
3. Ni, L.M., McKinley, P.K.: The architecture of interconnection networks. IEEE Comput. **26**(2), 62–76 (1993)
4. Kim, J., Dally, W.J., Scott, S., Abts, D.: Flattened butterfly: a cost-efficient topology for high-radix networks. In: International Symposium on Computer Architecture, pp. 126–137. ACM (2007)
5. Duato, J., Yalamanchili, S., Ni, L.: Interconnection networks: an engineering approach, morgan kaufmann (1997)
6. Puente, V.: Dynamic routing using virtual channels in NOCS. In: International Conference on High Performance Computing (2002)
7. Mutlu, O., Stark, J., Patt, Y.N.: Runahead execution: an alternative to very large instruction windows for out-of-order processors. In: High-Performance Computer Architecture, pp. 129–140. IEEE (2003)
8. Nicopoulos, D., Park, J., Kim, J., Vijaykrishnan, N., Irwin, M., Das, C.: Vichar: A dynamic virtual channel regulation scheme for network-on-chip routers. In: International Symposium on Microarchitecture, pp. 333–346 (2006)
9. Goossens, K., Dielissen, J., Radulescu, A.: Aethereal network on chip: concepts, architectures, and implementations. IEEE Design Test Comput. **22**(5), 414–421 (2005)
10. Srinivasan, S., Gupta, R.: Ras: a reliability-aware scheduler for multiprocessor systems-on-chip. In: Design Automation Conference (2004)
11. Satpathy, A.: FTR: a fault-tolerant router for network-on-chip. In: International Conference on VLSI Design, pp. 243–248. IEEE (2012)
12. Hu, J., Marculescu, R.: Design and exploration of on-chip crossbar networks. In: International Conference on Computer Design, pp. 524–529 (2005)
13. Mercier, P., et al.: Power-aware on-chip interconnect design. IEEE J. Emerg. Select. Topics Circuits Syst. **2**(2), 228–239 (2012)

14. Dang, Q., et al.: A power-efficient wireless network-on-chip architecture. In: International Symposium on Networks-on-Chip, pp. 25–32. IEEE (2011)
15. Pan, D., et al.: Firefly: illuminating future network-on-chip with nanophotonics. In: Design Automation Conference, pp. 694–699. ACM (2010)
16. Patti, R.: Three-dimensional integrated circuits and the future of system-on-chip designs. In: Proceedings of the IEEE, vol. 94, pp. 1214–1224. (2006)
17. Nicopoulos, D.: Data-centric thermal management in 3d multicore architectures. IEEE Trans. Very Large Scale Integr. VLSI Syst. **18**(10), 1432–1445 (2010)
18. Rahmani, A.-M., et al.: Applying machine learning for NOC design space exploration. In: Design, Automation and Test in Europe Conference, pp. 223–228 (2013)
19. Dai, Y.: A convolutional neural network approach for NOC routing. In: International Symposium on Networks-on-Chip (2018)
20. Bjerregaard, T., Mahadevan, S.: A survey of research and practices of networkon-chip. ACM Comput. Surv. **38**(1), 1–51 (2006)
21. Furber, S.: On-chip networks. Commun. ACM **49**(9), 43–47 (2006)
22. Chrysos, G., et al.: A survey on network-on-chip architectures and design methods. In: International Conference on Modern Circuits and Systems Technologies (2014)
23. Lee, S., et al.: A tree-based network-on-chip topology for high performance socs. In: Asia and South Pacific Design Automation Conference (2005)
24. Murali, S., De Micheli, G.: Mapping and physical planning of networks-on-chip architectures. In: Design Automation Conference, pp. 27–32 (2004)
25. Guan, N.L., et al.: Exploring wireless network-on-chip architectures using MMWAVE radios. In: International Symposium on Networks-on-Chip (2011)
26. Glass, C.J., Ni, L.M.: The turn model for adaptive routing. In: International Symposium on Computer Architecture, pp. 278–287 (1992)
27. Chien, A.A., Kim, J.: In: IEEE Transactions on Parallel and Distributed Systems, vol. 11, pp. 729–738 (2000)
28. Ababei, C., et al.: Routing with congestion awareness for networks-on-chip. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **28**(2), 237–250 (2009)
29. Wu, J.: LBDR: a fault-tolerant routing scheme for 3d NOCS. In: International Conference on ASIC (2011)
30. Moscibroda, T., Mutlu, O.: A case for bufferless routing in on-chip networks. ACM SIGARCH Comput. Archit. News **37**(3), 196–207 (2009)
31. Goossens, K., et al.: A reliable, real-time NOC for safety-critical systems. Integr. VLSI J. **46**(2), 154–170 (2013)
32. Shacham, A., Bergman, K., Carloni, L.P.: Photonic networks-on-chip for future chip multiprocessors. IEEE Trans. Comput. **57**(9), 1246–1260 (2008)
33. Abdelrazek, A., et al.: Machine learning for network-on-chip design space exploration. In: International Symposium on Networks-on-Chip (2017)
34. Jerger, N.E., Thottethodi, M., Kannan, R.: On-chip wireless communication. Proc. IEEE **106**(1), 54–73 (2017)
35. Shacham, A., Bergman, K., Carloni, L.P.: Photonic networks-on-chip for future generations of chip multiprocessors. IEEE Trans. Comput. **57**(9), 1246–1260 (2008)
36. Benini, L., De Micheli, G.: Multi-technology networks-on-chip. IEEE Design Test **34**(6), 10–17 (2017)
37. Wang, L., Zhang, Y., Qian, D.: Hybrid network-on-chip architectures: design and evaluation. IEEE Trans. Very Large Scale Integr. VLSI Syst. **29**(6), 1105–1118 (2021)
38. Shen, Y., Fu, Y., Zhang, L.: Deep reinforcement learning for adaptive routing in networks-on-chip. IEEE Trans. Parallel Distrib. Syst. **31**(8), 1937–1950 (2020)

39. Abdelhadi, A., Clancy, T.C.: Reinforcement learning for network-on-chip management: a survey. ACM Comput. Surv. **54**(5), 1–36 (2021)
40. Li, Z., Shen, Y., Qian, D.: A survey of machine learning techniques for networkonchip optimization. IEEE Access **10**, 39215–39235 (2022)
41. Kim, D., Lim, S.K.: Through-silicon-via aware interconnect design for 3d integrated circuits. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **18**(7), 1002–1014 (2010)
42. Yan, J., Yang, Y., Wang, Y.: Lifetime reliability-aware routing for networks-onchip. IEEE Trans. Very Large Scale Int. (VLSI) Syst. **22**(8), 1771–1784 (2014)
43. Murali, S., De Micheli, G.: Mapping and physical planning of networks-onchip architectures with quality-of-service guarantees. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **26**(11), 1943–1956 (2007)
44. Coskun, A.K., Rosing, T.S.: Temperature-aware system-level optimization for multicore architectures. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **38**(5), 878–891 (2019)
45. Reagen, B., Shao, U., Wei, G.-Y., Brooks, D.: MachSuite: benchmarks for accelerator design and customized architectures. In: Proceedings of the IEEE International Symposium on Workload Characterization (IISWC), pp. 110–119. IEEE (2016)
46. Parashar, A., et al.: Timeloop: a systematic approach to DNN accelerator evaluation. IEEE Micro **39**(3), 20–29 (2019)
47. Wang, H., Parashar, A., Emer, J.S.: AI-aware network-on-chip design for heterogeneous accelerators. IEEE Micro **43**(2), 34–44 (2023)
48. Fiorin, L., Palermo, G., Lukovic, S., Silvano, C.: Security issues in networks-onchip. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **27**(6), 1116–1129 (2008)
49. Rajendran, J., Karri, R., Sinanoglu, O.: Security analysis of integrated circuit layouts. IEEE Design Test **31**(3), 6–18 (2014)
50. Ancajas, D.M., Chakraborty, K., Tahoori, M.B.: Low-cost and scalable intrusion detection system for networks-on-chip. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **33**(8), 1220–1233 (2014)
51. Gebotys, C.H.: Hardware security and trust. IEEE Design Test **36**(2), 6–13 (2019)
52. Chen, Y.-H., Krishna, T., Emer, J., Sze, V.: Eyeriss: an energy-efficient reconfigurable accelerator for deep convolutional neural networks. IEEE J. Solid-State Circuits **52**(1), 127–138 (2016)
53. Sze, V., Chen, Y.-H., Yang, T.-J., Emer, J.S.: Efficient processing of deep neural networks: a tutorial and survey. Proc. IEEE **105**(12), 2295–2329 (2017)
54. Li, Z., Qian, D., Zhang, Y.: Machine learning-based routing and congestion control for networks-on-chip. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **38**(10), 1896–1909 (2019)
55. Farahnakian, F., Ashraf, A.J., Pahikkala, T., Plosila, J., Liljeberg, P., Porres, I.: Learning-based dynamic power management for multicore systems. ACM Trans. Embed. Comput. Syst. **16**(5), 1–20 (2017)
56. Pande, P.P., Grecu, C., Jones, M., Ivanov, A., Saleh, R.: Performance evaluation and design trade-offs for network-on-chip interconnect architectures. IEEE Trans. Comput. **54**(8), 1025–1040 (2005)

Geometric Modeling, Graphics and Visualization

Adaptive Hyperspectral Image Classification Using a CNN-MAOELM Hybrid Architecture

Oleksii Fesenko(✉), Robert Bieliakov(✉), Hryhorii Radzivilov(✉), Andrii Bernatskyi(✉), and Dmytro Koltovskov

Kruty Heroes Military Institute of Telecommunications and Information Technology, 45/1 Knyaziv Ostrozkyh Street, Kyiv 01011, Ukraine
{oleksij.fesenko,andrii.bernatskyi, dmytro.koltovskov}@viti.edu.ua, france417@gmail.com, grigoriy110371@gmail.com

Abstract. Hyperspectral image (HSI) classification on resource-constrained airborne platforms remains challenging because conventional Convolutional Neural Networks (CNN) require costly retraining when the input-stream distribution shifts due to atmospheric, illumination, or sensor-noise effects. The aim of this paper is to develop a method for synthesizing a hybrid neural-network architecture that combines deep spatial-spectral feature extraction with non-iterative streaming-mode adaptation under explicit memory and computation budgets. The proposed CNN-MAOELM (Convolutional Neural Network with Modified Adaptive Online Extreme Learning Machine) architecture couples a multi-scale convolutional feature extractor with an analytical MAOELM classifier whose output weights are recursively updated through least-squares estimation with adaptive forgetting, supplemented by controlled local correction of the convolutional subsystem when degradation is detected. Experimental verification on five benchmark hyperspectral datasets (Indian Pines, Pavia University, Salinas, Houston, HyRANK) shows competitive overall accuracy, more than twentyfold acceleration of online adaptation relative to CNN-AOELM, and substantially higher robustness under simulated concept drift, supporting onboard and near-real-time remote-sensing operation.

Keywords: hyperspectral image classification · convolutional neural network · extreme learning machine · adaptive online learning · resource-constrained inference

1 Introduction

Under contemporary conditions of military confrontation, remote sensing technologies play an important role in ensuring information superiority. Hyperspectral imagery (HSI) is especially valuable because it registers spectral signatures of objects in hundreds of narrow bands, enabling the detection of camouflaged equipment, identification of surface-cover materials, assessment of terrain passability, and detection of explosive hazards.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 393–412, 2026.
https://doi.org/10.1007/978-3-032-30491-9_25

Operational use of HSI requires classification algorithms capable of near-real-time decision-making, including onboard processing on unmanned aerial platforms with limited energy and computational resources. Convolutional Neural Networks (CNN) are widely used for HSI classification [1–4] and provide high accuracy on static benchmarks. However, their deployment in practical situational-awareness systems is limited by the high dimensionality of hyperspectral cubes, the scarcity of labeled samples, computational cost, and input-distribution shifts caused by weather, smoke, illumination changes, sensor noise, and camouflage variability. In addition, spectral uncertainty complicates classification because one class may exhibit different signatures under different acquisition conditions, while different objects may produce similar spectral responses.

Consequently, there is a need for hybrid classification methods that combine CNN capabilities in spatial-spectral feature extraction with the flexibility and speed of non-iterative learning algorithms. The developed paper addresses the stated problem by synthesizing the hybrid CNN with Modified Adaptive Online Extreme Learning Machine (CNN-MAOELM) architecture, which integrates a convolutional subsystem with the MAOELM classifier for streaming-mode operation.

2 Analysis of Recent Publications

A decisive stage in the development of computer vision is associated with the work of Krizhevsky *et al.* [1], where the AlexNet architecture was proposed. Subsequent studies devoted to VGGNet [2], ResNet [3], and DenseNet [4] demonstrated a stable tendency toward accuracy improvement with increasing network depth, although such growth is accompanied by a proportional increase in computational demand.

In the context of remote-sensing image analysis, Zhu *et al.* [5] showed the possibility of adapting residual architectures to multispectral data processing. Chen *et al.* [6] proposed deep residual networks for HSI classification and confirmed the effectiveness of deep models for representative benchmark tasks. At the same time, the structural complexity of such solutions demands substantial computational power and large labeled datasets.

In parallel, a line of fast non-iterative methods based on the Extreme Learning Machine (ELM) was formed. The basic ELM model [7] relies on random initialization of hidden-layer parameters and analytical computation of output weights, which considerably reduces training time. OS-ELM [8] extends the concept to sequential data arrival, while FOS-ELM [9] supplements it with a forgetting mechanism. Further development is represented by the kernel implementation for nonstationary series proposed by Wang and Han [10], as well as by the meta-cognitive sequential ELM for class imbalance and concept drift introduced by Mirza and Lin [11]. The aggregate of such results confirms the attractiveness of ELM-type approaches for streaming tasks; however, in their classical form they do not provide deep spatial-spectral representation learning needed for HSI.

For HSI classification, specialized spectral-spatial architectures have also been actively developed. SSRN [12] implements staged extraction of spectral and spatial representations within a 3D formulation. Deep recurrent networks for HSI [13] are focused on modeling inter-band dependence in spectral sequences. HybridSN [14] showed that

a combination of 3D and 2D convolutions makes it possible to reduce complexity compared with fully 3D networks while preserving high accuracy. Compact 3D-CNN solutions [15] confirmed the promise of lightweight architectures intended for reducing parametric space and computation budget.

Generalizing studies and contemporary architecture surveys also remain important. Li *et al.* [16] systematized early deep-learning approaches for HSI and described the transition from shallow classifiers to end-to-end models. A later survey by Bera *et al.* [17] detailed CNN-oriented solutions, whereas Kumar *et al.* [18] covered a broader spectrum of methods, including 3D-CNN, hybrid architectures, attention mechanisms, and transformers. Ashraf *et al.* [19] reflect the current direction of BERT-like spatial-spectral modeling of long-range dependencies between channels and neighboring pixels. Nevertheless, most such solutions remain oriented toward static benchmarks and offline retraining, which limits their suitability for streaming operation and for deployment on platforms with constrained computational resources.

Table 1. Comparative qualitative summary of representative HSI classification methods.

Method	Deep spatial-spectral	Non-iterative training	Online streaming adaptation	Resource-aware	Robust to concept drift
ResNet-50 [3]	✓	✗	✗	✗	✗
SSRN [12]	✓	✗	✗	✗	✗
HybridSN [14]	✓	✗	✗	◑	✗
Compact 3D-CNN [15]	✓	✗	✗	✓	✗
ELM [7]	✗	✓	✗	✓	✗
OS-ELM [8] / FOS-ELM [9]	✗	✓	✓	✓	◑
CNN-AOELM (baseline)	✓	✓	✓	◑	◑
CNN-MAOELM (proposed)	✓	✓	✓	✓	✓

In Table 1, ✓ denotes full support of the corresponding property, ✗ indicates its absence, and ◑ reflects partial support requiring retraining or external mechanisms. The comparison shows that existing reference architectures do not jointly provide deep spatial-spectral representation, non-iterative learning, online adaptation, resource awareness, and robustness to concept drift. This gap motivates the proposed CNN-MAOELM architecture, which coordinates recognition accuracy, training-time budget, and computational complexity through adaptive forgetting and automatic hyperparameter tuning. The purpose of the paper is to develop a method for synthesizing the CNN-MAOELM neural architecture for hyperspectral image classification. To reach the stated purpose, the following tasks are solved: (1) development of a hybrid architecture with a fast-adaptive

learning mechanism; (2) formalization of the mathematical model of the method; and (3) development of a three-phase learning algorithm with adaptation to newly arriving data in real time.

3 Mathematical Formalization of the Problem

3.1 Problem Statement

Consider a streaming sequence of hyperspectral images

$$\mathcal{S} = \{(X_t, y_t)\}_{t=1}^{T}, \tag{1}$$

where $X_t \in R^{H\times W\times C}$ denotes the input hyperspectral image tensor at time step t, with spatial dimensions $H \times W$ and C spectral channels, and $y_t \in \{1, \ldots, K\}$ is the corresponding class label, where K is the total number of classes.

The goal is to learn an adaptive classification operator

$$F_t : R^{H\times W\times C} \to \Delta^{K-1}, \tag{2}$$

defined as the composition of a convolutional feature extractor Φ_{CNN} and an extreme learning classifier Ψ_{ELM}:

$$\widehat{p_t} = F_t(X_t) = softmax(\Psi_{\text{ELM}}(\Phi_{\text{CNN}}(X_t; \theta_t); \beta_t)), \tag{3}$$

where $\widehat{p_t} = \left(\widehat{p_{t,1}}, \widehat{p_{t,2}}, \ldots, \widehat{p_{t,K}}\right) \in \Delta^{K-1}$ is the vector of posterior class probabilities, θ_t denotes the parameters of the CNN component adapted over time, and β_t is the output-weight matrix of the ELM component updated in the online mode.

The predicted class label is determined as

$$\widehat{y_t} = \arg\max_{k\in\{1,\ldots,K\}} \widehat{p_{t,k}}. \tag{4}$$

At each time step, the model first predicts $\widehat{y_t}$ for the incoming sample X_t, and after observing the true label y_t, updates the parameters θ_t and β_t in order to maintain classification accuracy under streaming data drift and constrained computational resources.

The optimization objective is formulated as

$$\min_{\{\theta_t,\beta_t\}_{t=1}^{T}} J = \frac{1}{T}\sum_{t=1}^{T}\left[L_{\text{CE}}\left(\widehat{p_t}, y_t\right) + \gamma_1 C(\theta_t) + \gamma_2 R_{\text{comp}}(\theta_t, \beta_t)\right], \tag{5}$$

where L_{CE} is the cross-entropy loss, $C(\theta_t)$ is a structural-complexity measure of the CNN architecture, $R_{\text{comp}}(\theta_t, \beta_t)$ denotes the normalized computational cost of model adaptation, and $\gamma_1, \gamma_2 \geq 0$ are weighting coefficients controlling the trade-off among classification accuracy, architectural complexity, and computational efficiency.

3.2 Constraints

Resource, structural, and operational constraints are expressed as a joint admissibility condition for the CNN-MAOELM model:

$$\begin{gathered} \mathrm{Mem}(\mathcal{F}_t) \le M_{max}, T_{proc} \le T_{arrival}, \\ N_{min} \le N_f^{(l)} \le N_{max}, H^{TH} + \lambda I \succ 0, \;\; \lambda > 0, \mathrm{Acc}(t) \ge \mathrm{Acc}_{\mathrm{threshold}} \end{gathered} \tag{6}$$

where $\mathrm{Mem}(\mathcal{F}_t)$ is the memory required by the model at time t; T_{proc} is the processing time of one batch; $N_f^{(l)}$ is the number of filters in the l-th convolutional layer; H^{TH}, guarantees positive definiteness of the regularized ELM matrix; and $\mathrm{Acc}(t) \ge \mathrm{Acc}_{\mathrm{threshold}}$ is the minimum admissible classification accuracy.

3.3 Assumptions

The developed formulation is based on four assumptions: 1) the condition of local stationarity, according to which the input data stream changes its statistical characteristics gradually, thus ensuring timely correction of the model weight parameters without the risk of catastrophic accuracy degradation; 2) representativeness of the initialization stage, assuming that the initial training sample contains a sufficient amount of structural features for correct formation of the base filters of the convolutional subsystem; 3) an assumption concerning the nature of disturbances, according to which spatial additive noise in hyperspectral images follows a Gaussian distribution and is statistically uncorrelated with the useful signal; and 4) hardware reliability of the system, guaranteeing fail-safe and continuous operation of the computing node throughout the entire iterative learning and adaptation cycle.

4 Proposed CNN-MAOELM Method

4.1 General Architecture

The proposed CNN-MAOELM architecture implements an incremental learning scheme in which spectral-spatial feature formation and online classifier adaptation are functionally separated. The convolutional subsystem forms compact discriminative embeddings from hyperspectral patches, whereas the MAOELM component performs non-iterative classification and recursive output-weight updating.

The algorithm consists of three functional blocks: Initial Training, Online Adaptation, and Training Cycle. The essence of the developed approach lies in the fact that the convolutional component does not perform continuous global retraining. Instead, it forms a compact feature vector that is supplied to the modified MAOELM classifier. Most of the operational adaptation is transferred from gradient procedures to a recursive analytical mechanism for updating output weights. When quality degradation signs arise, the method does not launch a global architecture search; it performs local correction of the parameters of the last convolutional block and of the ELM dimensionality.

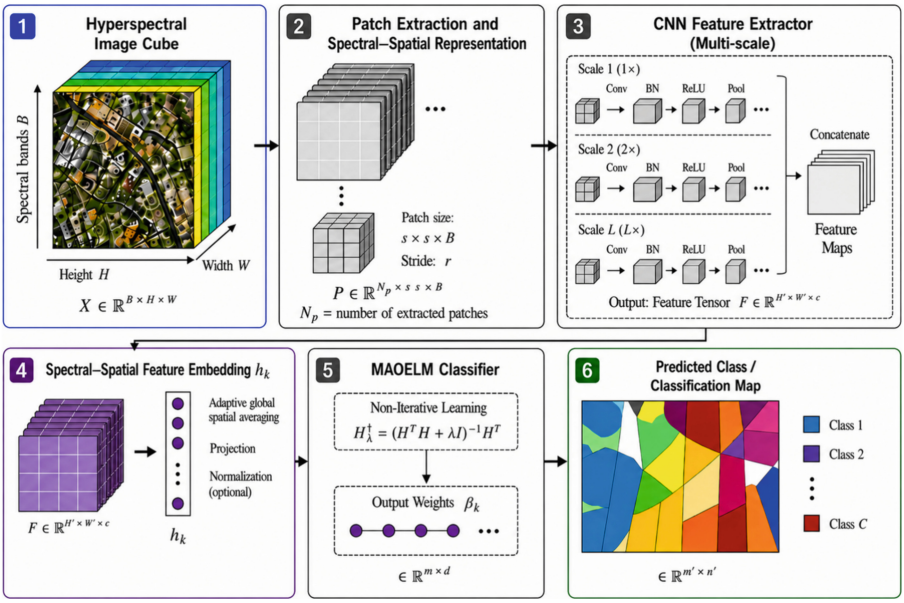


Fig. 1. General architecture of the proposed CNN-MAOELM method for adaptive hyperspectral image classification. The hyperspectral cube is partitioned into spectral-spatial patches, processed by a multi-scale CNN feature extractor, transformed into a compact embedding h_k, and classified by the MAOELM component to obtain the final classification map.

The functional flow of the proposed CNN-MAOELM architecture is shown in Fig. 1. The input hyperspectral cube is first divided into spatial-spectral patches, after which each patch is processed by a multi-scale convolutional extractor.

Let the input cube be denoted as $X \in R^{B\times H\times W}$, where B is the number of spectral bands and H, W are the spatial dimensions. The cube is partitioned into patches

$$p_k \in R^{s\times s\times B}, k = 1, 2, \ldots, N_p, \tag{7}$$

where $s \times s$ is the spatial window size. Each patch p_k is processed by the convolutional subsystem, and the obtained multi-scale feature tensor F_k is transformed into a compact embedding

$$h_k = N\left(W_f \cdot GAP(F_k) + b_f\right), h_k \in R^d, \tag{8}$$

where $GAP(\cdot)$ denotes global average pooling, W_f and b_f are projection parameters, $N(\cdot)$ is the normalization operator, and d is the embedding dimensionality.

The descriptor h_k is then supplied to the MAOELM classifier. The initial output-weight matrix is estimated analytically using the regularized Moore–Penrose solution

$$H_\lambda^\dagger = \left(H^T H + \lambda I\right)^{-1} H^T, \tag{9}$$

where λ is the Tikhonov regularization coefficient and I is the identity matrix. The class prediction for the k-th patch is obtained as

$$\hat{y}_k = softmax\left(H_{k\beta_k}\right). \tag{10}$$

The predicted class is assigned to the central pixel of the corresponding patch, forming the final classification map. Therefore, the general functional dependence of CNN-MAOELM can be written as

$$\hat{y}_k = MAOELM_{\beta_k}(\varphi_\theta(p_k)), \tag{11}$$

where $\varphi_{\theta(\cdot)}$ is the CNN-based spatial-spectral feature extractor and $MAOELM_{\beta_k}$ is the analytical classifier with the current output weights. The key feature of the architecture is the separation of computational roles: CNN performs deep feature extraction, whereas operational adaptation to streaming data is mainly carried out through recursive updating of the MAOELM output weights.

Thus, Fig. 1 reflects the data-flow organization of the developed method, while the internal organization of the corresponding training and adaptation procedures, together with the interaction between the three functional blocks. Detailed in the algorithmic scheme presented in Fig. 2 and elaborated in Sects. 4.2–4.4.

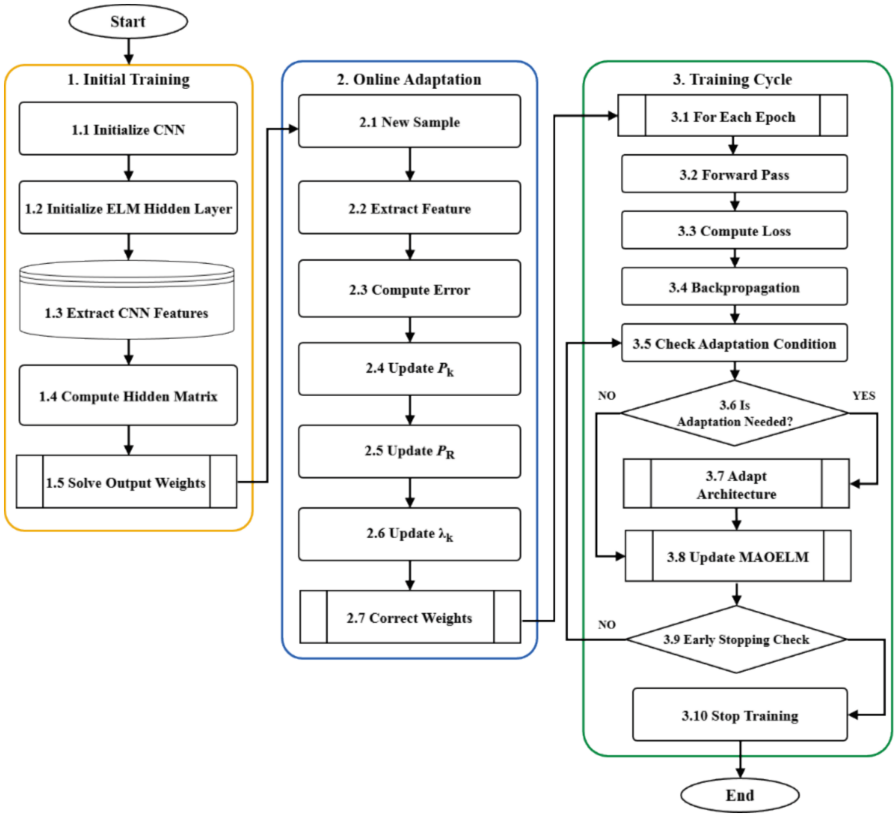


Fig. 2. Training and online adaptation procedure of the proposed CNN-MAOELM architecture.

4.2 Block 1. Initial Training

At the first step, the base parametric configuration of the hybrid CNN-MAOELM model is formed. This stage includes selection of the initial topology, initialization of weight coefficients and bias parameters, and specification of learning hyperparameters. After the spatial-spectral representation is formed by the convolutional subsystem, the output parameters of the ELM component are determined analytically, which yields the initial parametric state of the model. The obtained configuration acts as the basis for subsequent online updating, adaptive forgetting, and structural correction under arrival of new data.

For numerical stability of the initial stage, initialization of CNN weight coefficients is performed according to the He distribution:

$$\mathbf{W}_{\text{CNN}}^{(l)} \sim \mathcal{N}\left(0, \left(\sigma_{\text{CNN}}^{(l)}\right)^2\right), \sigma_{\text{CNN}}^{(l)} = \sqrt{\frac{2}{n_{\text{in}}^{(l)}}}, \tag{12}$$

where $n_{\text{in}}^{(l)}$ denotes the fan-in of the l-th convolutional layer, and $\mathcal{N}(\mu, \sigma^2)$ represents a normal distribution with mean μ and variance σ^2. Hidden-layer ELM weights are initialized randomly as

$$\mathbf{W}_{\text{ELM}} \sim \mathcal{N}\left(0, \sigma_W^2\right), \sigma_W = \sqrt{\frac{2}{n\text{in} + n_{\text{hidden}}}}, \mathbf{b}_{\text{ELM}} \sim \mathcal{N}(0, 1), \tag{13}$$

where $\mathbf{W}_{\text{ELM}}$ and $\mathbf{b}_{\text{ELM}}$ are the hidden-layer weight matrix and bias vector of the ELM component, respectively.

Forward propagation through the CNN subsystem is described, for an arbitrary layer l, by

$$z_{i,j,k}^{(l)} = \sum\nolimits_{m=0}^{M-1} \sum\nolimits_{n=0}^{N-1} \sum\nolimits_{c=0}^{C_{l-1}-1} w_{m,n,c,k}^{(l)} \cdot a_{i+m,j+n,c}^{(l-1)} + b_k^{(l)}, \tag{14}$$

where $w_{u,v,c,k}^{(l)}$ denotes the kernel coefficients of size $U \times V$ linking input channel c and output channel k, while $b_k^{(l)}$ is the bias of the k-th channel. Activation is implemented by ReLU,

$$a_{i,j,k}^{(l)} = \text{ReLU}\left(z_{i,j,k}^{(l)}\right) = \max\left(0,\ z_{i,j,k}^{(l)}\right). \tag{15}$$

To stabilize learning, batch normalization is applied:

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \in}},\ y_i = \gamma \cdot \hat{x}_i + \beta, \tag{16}$$

where μ_B and σ_B^2 are the batch mean and variance, and γ, β are the trainable scale and shift parameters.

After passing the convolutional stages, global average pooling forms a compact feature vector:

$$\mathbf{f} = \text{GlobalAvgPool}\left(a^{(L)}\right). \tag{17}$$

The hidden-layer activation matrix of the ELM component is then obtained as

$$\mathbf{H} = \sigma\left(\mathbf{F} \cdot W_{\text{ELM}} + 1_N \cdot b^T\right), \tag{18}$$

where $\mathbf{F}$ is the feature matrix for all training samples, $\sigma(\cdot)$ is the sigmoid activation, and 1 is the column vector of ones.

Analytical estimation of the ELM output weights is performed by the regularized Moore–Penrose solution

$$\beta = \left(\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I}\right)^{-1}\mathbf{H}^T\mathbf{T}, \tag{19}$$

where $\mathbf{T}$ is the one-hot target matrix and λ is the Tikhonov regularization coefficient. Simultaneously, the covariance matrix is initialized as

$$\mathbf{P}_0 = \left(\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I}\right)^{-1}. \tag{20}$$

Such a formulation provides a globally optimal solution for the output weights within the adopted regularized analytical setting and therefore eliminates the local-minimum risk characteristic of purely gradient-based optimization.

4.3 Block 2. Online Adaptation Based on MAOELM

The online-adaptation block constitutes the central component of the scientific novelty of the developed method. The modified adaptive online sequential extreme learning machine extends the ELM concept toward processing of streaming data arriving in successive portions. Within this approach, dynamic parameter updating is carried out according to newly observed samples without the need for retraining on the full dataset. Such organization of the adaptation mechanism preserves computational efficiency, reduces reconfiguration time, and maintains classification stability under variation of the input-stream structure.

For every new sample, after CNN-based feature extraction, the hidden-layer output vector is computed as

$$h_k = \sigma(\mathbf{W}_{\text{ELM}} \cdot \mathbf{f}_k + b), \tag{21}$$

where $\mathbf{f}_k$ is the feature vector formed by the CNN subsystem. The prediction error is then defined by

$$e_k = \mathbf{t}_k - \boldsymbol{h}_k^T\boldsymbol{\beta}_{k-1}, \tag{22}$$

where $\mathbf{t}_k$ is the one-hot target vector.

Recursive updating of the covariance matrix is carried out with the Woodbury identity:

$$\mathbf{P}_k = \frac{1}{\lambda_k}\left(\mathbf{P}_{k-1} - \frac{\mathbf{P}_{k-1}\boldsymbol{h}_k\boldsymbol{h}_k^T\mathbf{P}_{k-1}}{\lambda_k + \boldsymbol{h}_k^T\mathbf{P}_{k-1}\boldsymbol{h}_k}\right), \tag{23}$$

where λ_k is the adaptive forgetting coefficient. The output weights are corrected through the modified recursive least-squares rule

$$\boldsymbol{\beta}_k = \boldsymbol{\beta}_{k-1} + \mathbf{P}_k h_k \Big(t_k - h_k^T \boldsymbol{\beta}_{k-1} \Big). \tag{24}$$

Adaptive forgetting is regulated by the prediction-error norm:

$$\lambda_k = \lambda_{\min} + (\lambda_{\max} - \lambda_{\min}) \cdot \exp(-\gamma \| e_k \|_2), \tag{25}$$

where $\lambda_{\min}$ and $\lambda_{\max}$ define the admissible forgetting range and α is the sensitivity coefficient. When the error magnitude increases, λ_k decreases, which strengthens the contribution of the current observation and accelerates adaptation. When the error is small, λ_k approaches unity, corresponding to weak forgetting and retention of long-term information.

At the final stage of each iteration, inertial correction of the output weights is performed:

$$\boldsymbol{\beta}_k^{\text{final}} = \lambda_k \cdot \boldsymbol{\beta}_k + (1 - \lambda_k) \cdot \boldsymbol{\beta}_{k-1}. \tag{26}$$

where $\kappa \in (0, 1]$ is the smoothing coefficient. By functional meaning, such a procedure is equivalent to a low-pass filter operating in the parameter space and supports stable adaptation in the presence of local perturbations.

4.4 Block 3. Training Cycle

The training cycle provides iterative optimization of the convolutional subsystem together with adaptive updating of the ELM component. Within this cycle, mechanisms of dynamic architecture adaptation and early stopping are coordinated with multicriteria monitoring of performance metrics.

Step 3.1–3.2. Forward propagation. For each epoch $e \in \{1, \ldots, E_{max}\}$ and for each mini-batch, forward propagation through the convolutional and ELM components is carried out, yielding the prediction matrix $\hat{\mathrm{Y}}_{batch}$.

Step 3.3. Loss-function evaluation. At this stage, an integral target functional is formed, within which three key training criteria are coordinated: classification accuracy, parametric-complexity control, and computational-cost control. The general functional is given by

$$\mathcal{L}_{total} = \mathcal{L}_{\mathcal{CE}} + \alpha_{1\mathcal{L}_{\mathcal{L}2}} + \alpha_{2\mathcal{L}_{comp}}, \tag{27}$$

where $\mathcal{L}_{\mathcal{CE}}$ denotes the cross-entropy term, $\mathcal{L}_{\mathcal{L}2}$ is the regularization term for the convolutional weights, and $\mathcal{L}_{comp}$characterizes normalized computational complexity. Coefficients α_1 and α_2 set the relative contribution of the regularization and resource-aware components in the overall optimization procedure.

For the multiclass classification problem with K classes, the cross-entropy term is defined as

$$\mathcal{L}_{\mathcal{CE}} = -\frac{1}{B} \sum\nolimits_{i=1}^{B} \sum\nolimits_{j=1}^{K} y_{ij} \log\big(\widehat{y_{ij}}\big), \tag{28}$$

where B is the mini-batch size, $y_{ij} \in \{0, 1\}$ is the indicator of membership of the i-th sample in the j-th class, and $\hat{y}_{ij}$ is the predicted probability for class j. This component minimizes the discrepancy between the reference class structure and the model outputs.

The regularization term is represented as

$$\mathcal{L}_{\mathcal{L}2} = \sum\nolimits_{l=1}^{L} ||\mathbf{W}^{(l)}||_F^2, \tag{29}$$

where $\mathbf{W}^l$ is the weight matrix of the $l - th$ convolutional level, $||\cdot||_F$ is the Frobenius norm, and L is the number of convolutional levels. Introduction of this term constrains excessive parameter growth and preserves structural compactness of the architecture.

The component that reflects computational cost is defined as

$$\mathcal{L}_{comp} = \frac{\text{FLOPs}(\theta)}{\text{FLOPs}_{\text{target}}}, \tag{30}$$

where FLOPs (Floating-Point Operations) (θ) characterizes the actual computational complexity of the model with parameters θ, and $\text{FLOPs}_{\text{target}}$ denotes the target computational budget. Such normalization coordinates computational-efficiency requirements with the main classification term within a unified optimization functional.

Thus, minimization of $\mathcal{L}_{total}$ orients training toward a compromise between recognition quality, parametric compactness, and resource efficiency, which is essential for an adaptive model intended for operation under constrained computational resources.

Step 3.4. Backpropagation with CNN-parameter update. This stage updates the convolutional subsystem using gradients of the target functional. To preserve numerical stability and prevent uncontrolled gradient growth, gradient clipping is applied:

$$\nabla \mathbf{W}_{\text{clip}}^{(l)} = \begin{cases} \nabla \mathbf{W}^{(l)}, & \text{if } \|\nabla \mathbf{W}^{(l)}\| \le \tau, \\ \frac{\tau}{\|\nabla \mathbf{W}^{(l)}\|} \cdot \nabla \mathbf{W}^{(l)}, & \text{if } \|\nabla \mathbf{W}^{(l)}\| > \tau, \end{cases} \tag{31}$$

where $\tau > 0$ is the threshold value of the gradient norm. Weight updating is then performed by the Adam algorithm:

$$\begin{gathered} \boldsymbol{m}_t^{(l)} = \beta_1 \boldsymbol{m}_{t-1}^{(l)} + (1 - \beta_1) \nabla \mathbf{W}_{\text{clip}}^{(l)}, \\ \mathbf{v}_t^{(l)} = \beta_2 \mathbf{v}_{t-1}^{(l)} + (1 - \beta_2) \left(\nabla \mathbf{W}_{\text{clip}}^{(l)} \right)^{\odot 2}, \\ \mathbf{W}_t^{(l)} = \mathbf{W}_{t-1}^{(l)} - \eta \cdot \frac{\boldsymbol{m}_t^{(l)} / \left(1 - \beta_1^t\right)}{\sqrt{\mathbf{v}_t^{(l)} / \left(1 - \beta_2^t\right)} + \varepsilon}, \end{gathered} \tag{32}$$

where $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-8}$ are the Adam parameters.

Step 3.5–3.7. Architecture adaptation. The condition for architecture adaptation is checked on the basis of continuous monitoring of the loss and estimation of the validation-accuracy trend, formalized as

$$\text{trend}_{\text{acc}} = \frac{\text{Acc}_{\text{val}}^{(e)} - \text{Acc}_{\text{val}}^{(e-w)}}{w}, \tag{33}$$

where w is the size of the sliding analysis window. Adaptation is initiated when $\text{trend}_{\text{acc}} < \delta_{adapt}$, where δ_{adapt} is the threshold value.

For each level l, the importance metric is computed as the averaged $L2$ norm of the gradients:

$$\text{importance}^{(l)} = \frac{1}{B}\sum_{i=1}^{B}\left\|\frac{\partial \mathcal{L}_{\text{CE}}}{\partial a_i^{(l)}}\right\|_2. \tag{34}$$

The change in the number of filters is determined by the following rule system:

$$\Delta_{\text{filters}}^{(l)} = \begin{cases} +\alpha_{\text{expand}} \cdot N_f^{(l)}, & \text{if importance}^{(l)} > \bar{I} \text{ and util}^{(l)} > 0.7, \\ -\alpha_{\text{reduce}} \cdot N_f^{(l)}, & \text{if importance}^{(l)} < 0.5\bar{I} \text{ or util}^{(l)} < 0.3, \\ 0, & \text{otherwise}, \end{cases} \tag{35}$$

where $\bar{\text{I}}$ is the average importance across all levels, $util^{(l)}$ is the neuron-utilization coefficient, and $\alpha_{expand}, \alpha_{reduce} \in (0,1)$ are the adaptation-rate coefficients.

At the next stage, new filters are generated on the principle of parametric inheritance. Within this approach, newly added convolutional elements are initialized on the basis of parametric information accumulated inside the already formed architecture. This preserves continuity of the feature representation and supports stability of subsequent optimization. The corresponding expression is

$$\mathbf{W}_{\text{new}}^{(l)} = \mathbf{W}_{\text{best}}^{(l)} + \mathcal{N}\left(0, \sigma_{\text{inherit}}^2\right). \tag{36}$$

Step 3.8. MAOELM-parameter update. For a batch of new features $\mathbf{H}_{batch}$, recursive updating of the covariance matrix $\mathbf{P}_t$ and the output-weight matrix $\boldsymbol{\beta}_t$ is performed according to Eqs. (23) and (26). Equation (23) specifies correction of $\mathbf{P}_t$ with allowance for the information contribution of the new batch to the hidden feature space. Equation (26) describes refinement of $\boldsymbol{\beta}_t$ on the basis of the approximation error of the current batch, defined by the difference between the reference labels Y_{batch} and the prediction $\mathbf{H}_{batch}\boldsymbol{\beta}_{t-1}$. This scheme adapts MAOELM to new data without retraining on the full observation set and therefore increases computational efficiency in streaming mode.

$$\mathbf{P}_t = \mathbf{P}_{t-1} - \mathbf{H}_{\text{batch}}^T\left(I + \mathbf{H}_{\text{batch}}\mathbf{P}_{t-1}\mathbf{H}_{\text{batch}}^T\right)^{-1}\mathbf{H}_{\text{batch}}\mathbf{P}_{t-1}, \tag{37}$$

$$\boldsymbol{\beta}_t = \boldsymbol{\beta}_{t-1} + \mathbf{P}_t\mathbf{H}_{\text{batch}}^T(Y_{\text{batch}} - \mathbf{H}_{\text{batch}}\beta_{t-1}). \tag{38}$$

where $\mathbf{H}_{batch} \in R^{B\times M}$ is the hidden-layer activation matrix for the current batch of size B, M is the number of hidden neurons, $Y_{batch} \in R^{B\times C}$ is the target-output matrix, and $\boldsymbol{\beta}_t \in R^{M\times C}$ is the matrix of output weights for C classes.

In such a formulation, parameter updating is carried out in the space of compact feature representations, which increases the suitability of MAOELM for multidimensional streaming data.

Step 3.9. Early-stopping mechanism. Early termination of iterative training is implemented on the basis of a system of stopping criteria oriented toward convergence control,

attainment of the target quality level, and prevention of overfitting. The training cycle is stopped when the maximum number of epochs E_{max} reached, when the selected metric stagnates during $E_{patience}$ epochs, when validation accuracy reaches the target level $Acc^{e}_{val} \geq Acc_{target}$, or when the admissible overfitting threshold is exceeded. Diagnosis of the latter is based on the relative difference between training and validation accuracy, defined as

$$overfit = \frac{\mathrm{Acc}^{(e)}_{\mathrm{train}} - \mathrm{Acc}^{(e)}_{\mathrm{val}}}{\mathrm{Acc}^{(e)}_{\mathrm{train}}} > overfit_{\mathrm{threshold}}. \tag{39}$$

The above stopping scheme provides coordination between accuracy, stability, and computational expediency of the training procedure.

5 Experimental Study

To verify the proposed method, a benchmark suite of five hyperspectral image (HSI) datasets was formed (Table 2), covering urban, agricultural, and mixed scene types. In particular, the Indian Pines dataset [15] represents an agricultural scene with shelterbelts and provides a suitable test environment for evaluating the algorithm under mixed-pixel conditions. The Salinas dataset [16] is characterized by high spatial resolution, which enables the assessment of fine spectral discrimination between adjacent classes. The Pavia University dataset [17] is used to investigate urban areas and makes it possible to analyze classification accuracy at spatial object boundaries as well as during the recognition of small structures. Additional verification of model stability under the significant spectral complexity of urban development is provided by the Houston dataset [18]. The robustness of the method to the influence of complex background conditions is evaluated using the HyRANK dataset [19], which describes a natural landscape with increased intraclass variability of spectral features.

The selection of these specific datasets is motivated by the necessity to rigorously evaluate the algorithm's robustness against key destabilizing factors. These include spatial heterogeneity (dense urban environments in the Houston dataset and complex topography in HyRANK), varied spatial resolutions (1.3–30 m/pixel), the prevalence of mixed pixels (Indian Pines), and high inter-class spectral similarity (Salinas).

Table 2. Characteristics of the benchmark hyperspectral datasets.

Dataset	Sensor	Size	Bands	Classes	Spatial resolution
Houston	ITRES CASI	349 × 1905	144	15	2.5 m
HyRANK	Hyperion	1225 × 1220	176	14	30 m
Indian Pines	AVIRIS	145 × 145	220	16	20 m
Pavia Univ	ROSIS	610 × 340	103	9	1.3 m
Salinas	AVIRIS	512 × 217	224	16	3.7 m

For comparative analysis, the following methods were selected: a classical deep CNN (ResNet-50), baseline ELM with manually extracted features, hybrid CNN-ELM, adaptive CNN-AOELM, and the developed CNN-MAOELM. The experiments were implemented in Python 3.10 using PyTorch 2.1 [20] for neural-network modelling, NumPy 1.26 [21] and SciPy 1.11 [22] for numerical computations, scikit-learn 1.3 [23] for evaluation metrics, and Matplotlib 3.8 [24] for visualization. The development and execution environment was Visual Studio with Python; the computational platform comprised an Intel Core i7 CPU, 32 GB RAM, and an NVIDIA GeForce RTX 3060 GPU with CUDA 12.1.

For quantitative assessment of the effectiveness of the developed CNN-MAOELM method, three conventional quality metrics for hyperspectral-image classification were used, which provide a comprehensive analysis of model performance.

Overall Accuracy (OA) is defined as the ratio of the number of correctly classified pixels to the total number of test pixels and is calculated by the formula: $OA = \left(\frac{1}{N}\right) \times \Sigma\delta(\hat{y}_i, y_i)$, where N is the total number of test samples, $\hat{y}_i$ is the predicted class label, y_i is the true label, $\delta(\cdot, \cdot)$ is the Kronecker delta function. OA is a basic metric characterizing the overall effectiveness of the classifier; however, it may be biased under an imbalanced class distribution.

Cohen's Kappa (κ) takes into account the possibility of random agreement in classification and is defined as: $\kappa = \frac{(OA - p^e)}{(1 - p^e)}$, *where* $p^e = \left(\frac{1}{N^2}\right) \times \Sigma_k n_k \cdot m_k$—the probability of random agreement, n_k—the number of samples classified as class k, m_k—the actual number of samples of class k. The value $\kappa \in [-1, 1]$, where $\kappa = 1$ corresponds to perfect classification and $\kappa = 0$ corresponds to random guessing. The coefficient κ is more informative than OA for problems with imbalanced classes, which is typical of hyperspectral data.

Average Accuracy (AA) is calculated as the arithmetic mean of classification accuracies for each individual class: $AA = \left(\frac{1}{K}\right) \times \Sigma_k \left(\frac{n_{kk}}{n_k}\right)$, where K is the total number of classes, n_{kk}—the number of correctly classified samples of class k, n_k—the total number of samples of class k. AA provides a balanced assessment of classifier performance irrespective of the size of each class, which is important for tasks in which minority classes have high tactical significance.

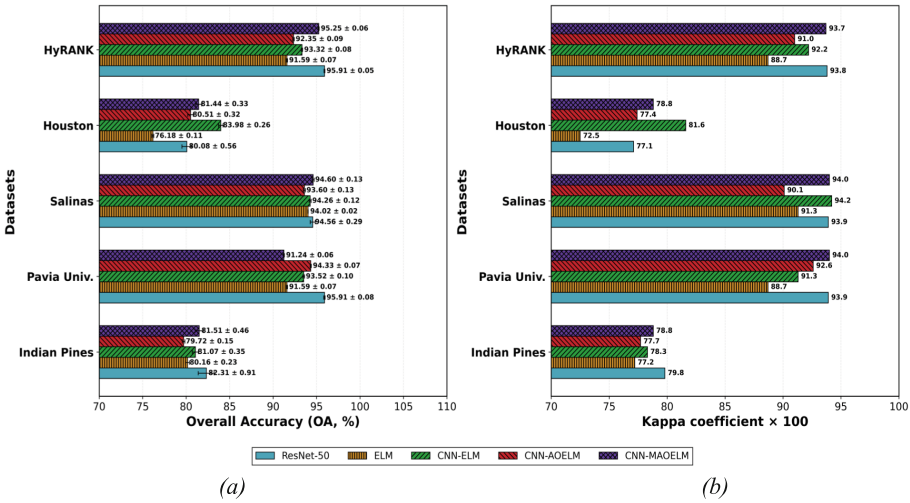
5.1 Classification Results

The OA metric reflects the share of correctly classified pixels relative to the total test volume and therefore provides the most direct integral view of classifier performance. As shown in the left panel of Fig. 3, CNN-MAOELM does not exhibit unconditional dominance across all five datasets, yet ensures stably high accuracy and a more favorable compromise between accuracy and adaptability.

The developed architecture shows its strongest behavior on HyRANK and Houston, where the adaptive mechanism yields improved resistance to spectral variability and urban heterogeneity.

The agreement coefficient κ reveals a similar tendency: CNN-MAOELM forms an averaged agreement level comparable with ResNet-50 while surpassing the baseline ELM and demonstrating a clear advantage on the Houston scene. Such behavior confirms that the proposed architecture preserves statistical stability of recognition while reducing dependence on heavy end-to-end retraining.

The AA metric is most sensitive to recognition quality for minority classes, because it is independent of the population size of each class. Figure 4 reveals a more subtle structure of inter-method differences in AA. Although ResNet-50 remains highly competitive on several datasets, CNN-MAOELM retains close average accuracy and offers a more rational compromise when the requirement of online adaptation is taken into account. Such a property is especially important for real-world HSI analysis, where appearance of new masking strategies or acquisition changes may force the classifier to react without full retraining.



(a) *(b)*

Fig. 3. Comparison of the algorithms across five hyperspectral datasets: *(a) Overall Accuracy (OA), %; (b) Cohen's kappa coefficient, $\kappa \times 100$.*

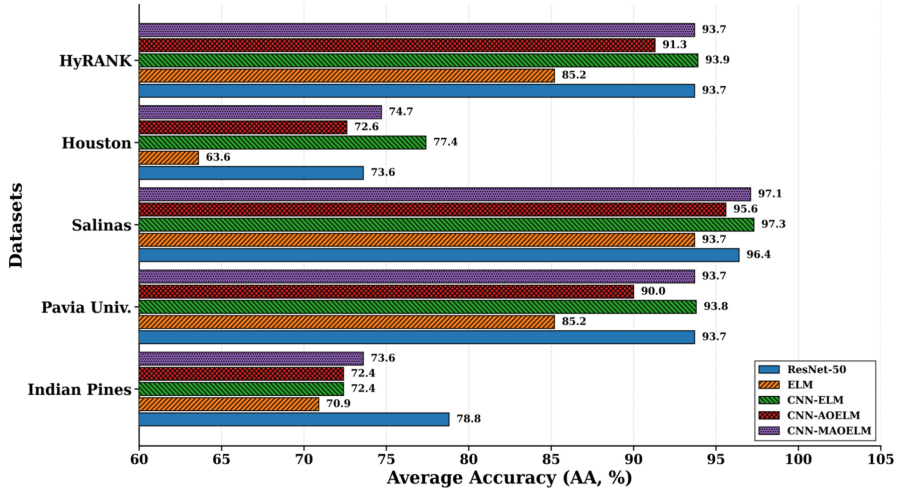


Fig. 4. Comparison of the algorithms by Average Accuracy (AA), %.

5.2 Analysis of Temporal Characteristics

Analysis of temporal characteristics confirms the substantial computational efficiency of the developed architecture. Compared with ResNet-50, initial training time is reduced approximately twofold: 40.34 min instead of 1 h 20 min. The main advantage of the method lies in its high speed of online adaptation to new streaming data. Owing to the modified recursive mechanism, adaptation to 1000 new samples requires only 7.45 s, whereas classical CNN-ELM and ResNet-50 solutions require computationally expensive full retraining. Even relative to CNN-AOELM, the proposed optimization accelerates adaptation by more than twenty times, which makes the hybrid neural network a rational solution for real-time hardware systems (Table 3).

Table 3. Comparison of training time and adaptation speed.

Method	Training time	Adaptation time (1000 samples)
ResNet-50	1 h 20 min	Full retraining / 1 h 20 min
ELM	1.25 min	Full retraining / 1 min
CNN-ELM	47.23 min	Full retraining / 47.25 min
CNN-AOELM	41.45 min	2.07 min
CNN-MAOELM	40.34 min	7.45 s

5.3 Robustness Under Input-Distribution Shift

To verify the key advantage of the developed CNN-MAOELM method with respect to the ability for online adaptation under changing characteristics of the input data, a series of experiments was carried out with emulation of distribution shift (concept drift). This experiment models realistic scenarios of hyperspectral-image quality degradation caused by atmospheric interference, changes in illumination conditions, smoke contamination, or hardware noise of sensors in the onboard UAV system.

The experimental methodology consists in gradual addition of additive Gaussian noise $\xi \sim \mathcal{N}(0, \sigma^2 I)$ to the normalized spectral vectors of the test subset $\widetilde{X}_t = X_t + \xi$, where the intensity parameter $\sigma \in \{0.0, 0.01, 0.05, 0.1, 0.2, 0.3, 0.5\}$ varies from the absence of perturbation to substantial distortion of the input data. The choice of this grid is dictated by physically meaningful operating regimes typical of hyperspectral acquisition: $\sigma \in [0.01,\ 0.05]$ reproduces low-level sensor and quantization noise; $\sigma \in [0.1,\ 0.2]$ models moderate atmospheric scattering and acquisition-related distortions; while $\sigma \in [0.3,\ 0.5]$ emulates severe observation conditions, including degraded illumination and strong instrumental interference.

The stated protocol reproduces a typical scenario of practical application in which a model pre-trained under controlled conditions operates in an operational environment with a changed statistical structure of input data. To ensure correctness of comparison, all investigated methods received the same noisy input data within the full set of spectral channels.

Additionally, for CNN-MAOELM an assessment of the effectiveness of the online-adaptation mechanism (Block 2 of the developed algorithm) was carried out. The adaptation procedure consisted of the following: a subset of $N_{\text{adapt}} = 300$ noisy samples with known class labels was selected from the test subset, on which recursive updating of the output weights β was performed using the modified RLS algorithm with the adaptive forgetting coefficient λ_k. After completion of adaptation, the model was evaluated on the remaining part of the test subset.

The experimental results are shown in Fig. 5 and indicate that, at a low noise level $\sigma \leq 0.05$, all methods retain high classification accuracy. However, as the noise intensity increases to $\sigma > 0.1$, substantial performance degradation is observed for all static models and for one method with an online-adaptation mechanism, which confirms the hypothesis about the inability of fixed classifiers to operate effectively under distribution shift.

At the same time, CNN-MAOELM with the activated online-adaptation mechanism demonstrates significantly higher robustness: at $\sigma = 0.2$, the model preserves accuracy at the level of 60–73% depending on the dataset, whereas static CNN methods degrade to 20–35%. This gain is caused by the ability of the adaptive forgetting coefficient λ_k to dynamically regulate the balance between retention of previous knowledge and integration of new information about the changed distribution.

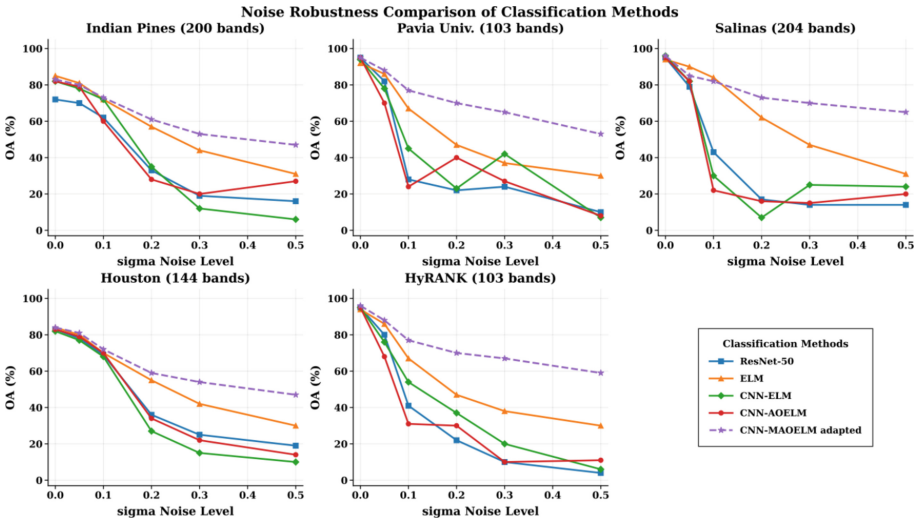


Fig. 5. Robustness analysis under increasing input noise.

6 Scientific Novelty of the Obtained Results

First, a method for synthesizing the hybrid CNN-MAOELM neural architecture for hyperspectral image classification is developed. The distinguishing feature of the method lies in a three-phase operating algorithm that integrates initial analytical identification of the ELM component, recursive online adaptation of output weights, and controlled architecture correction of the convolutional subsystem. Owing to such organization, the developed solution combines deep spatial-spectral feature extraction with high-speed streaming adaptation.

Second, the mechanism of adaptive online learning for streaming data is improved. The essential difference lies in the use of recursive least squares with adaptive forgetting and inertial smoothing of the output-weight matrix. Such a combination makes it possible to preserve numerical stability, suppress abrupt parameter oscillations, and increase classifier robustness under concept drift.

Third, the method of automatic neural-architecture correction during learning receives further development. The developed formulation uses continuous monitoring of performance metrics and gradient-based importance analysis of convolutional layers. As a result, architecture correction is performed not through a costly global search but through local structural adaptation, which is substantially more compatible with resource-constrained operation.

7 Conclusions

The paper develops a method for synthesizing the hybrid CNN-MAOELM neural architecture intended for classification of hyperspectral images under constrained computational resources. The developed approach combines a convolutional subsystem for

spatial-spectral feature extraction with a modified adaptive online extreme learning machine, which makes it possible to preserve competitive classification accuracy while considerably reducing training and adaptation time.

Practical value of the developed method is manifested most distinctly under input-distribution shift. Experimental results with gradually increasing Gaussian noise confirm that the method maintains substantially higher robustness than static baselines and classical hybrid solutions. For 1000 new samples, online updating is performed in 7.45 s, which supports applicability of the architecture to onboard and near-real-time systems.

7.1 Future Work

Future work will address the integration of spatial-spectral attention mechanisms, uncertainty-aware adaptive forgetting in the MAOELM update procedure, and automated local correction of the CNN subsystem. Additional research will focus on embedded implementation on onboard platforms such as NVIDIA Jetson Orin, Xilinx Zynq UltraScale +, and Google Edge TPU, as well as on extending the architecture to multisensor fusion of hyperspectral, SAR, and thermal data for all-weather unmanned aerial remote sensing.

Acknowledgments. The authors thank the Kruty Heroes Military Institute of Telecommunications and Information Technology for institutional support and access to computational facilities. The authors are grateful to the anonymous reviewers of ICCSA 2026 for constructive comments that contributed to the improvement of the manuscript.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105 (2012)
2. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015). https://doi.org/10.48550/arXiv.1409.1556
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings IEEE CVPR, pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
4. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings IEEE CVPR, pp. 4700–4708 (2017). https://doi.org/10.1109/CVPR.2017.243
5. Zhu, X.X., et al.: Deep learning in remote sensing: a comprehensive review and list of resources. IEEE Geosci. Remote Sens. Mag. **5**(4), 8–36 (2017). https://doi.org/10.1109/MGRS.2017.27623071
6. Chen, Y., Jiang, H., Li, C., Jia, X., Ghamisi, P.: Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. IEEE Trans. Geosci. Remote Sens. **54**(10), 6232–6251 (2016). https://doi.org/10.1109/TGRS.2016.2584107
7. Huang, G.-B., Zhu, Q.-Y., Siew, C.-K.: Extreme learning machine: theory and applications. Neurocomputing **70**(1–3), 489–501 (2006). https://doi.org/10.1016/j.neucom.2005.12.126

8. Liang, N.-Y., Huang, G.-B., Saratchandran, P., Sundararajan, N.: A fast and accurate online sequential learning algorithm for feedforward networks. IEEE Trans. Neural Netw. **17**(6), 1411–1423 (2006). https://doi.org/10.1109/TNN.2006.880583
9. Zhao, J., Wang, Z., Park, D.S.: Online sequential extreme learning machine with forgetting mechanism. Neurocomputing **87**, 79–89 (2012). https://doi.org/10.1016/j.neucom.2012.01.027
10. Wang, X., Han, M.: Online sequential extreme learning machine with kernels for nonstationary time series prediction. Neurocomputing **145**, 90–97 (2014). https://doi.org/10.1016/j.neucom.2014.05.003
11. Mirza, B., Lin, Z.: Meta-cognitive online sequential extreme learning machine for imbalanced and concept-drifting data classification. Neural Netw. **80**, 79–94 (2016). https://doi.org/10.1016/j.neunet.2016.04.009
12. Zhong, Z., Li, J., Luo, Z., Chapman, M.: Spectral-spatial residual network for hyperspectral image classification: a 3-D deep learning framework. IEEE Trans. Geosci. Remote Sens. **56**(2), 847–858 (2018). https://doi.org/10.1109/TGRS.2017.2755542
13. Mou, L., Ghamisi, P., Zhu, X.X.: Deep recurrent neural networks for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. **55**(7), 3639–3655 (2017). https://doi.org/10.1109/TGRS.2016.2636241
14. Roy, S.K., Krishna, G., Dubey, S.R., Chaudhuri, B.B.: HybridSN: exploring 3D–2D CNN feature hierarchy for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. **17**(2), 277–281 (2020). https://doi.org/10.1109/LGRS.2019.2918719
15. Ahmad, M., Khan, A.M., Mazzara, M., Distefano, S., Ali, M., Sarfraz, M.S.: A fast and compact 3-D CNN for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. **19**, 1–5 (2022). https://doi.org/10.1109/LGRS.2020.3043712
16. Li, S., Song, W., Fang, L., Chen, Y., Ghamisi, P., Benediktsson, J.A.: Deep learning for hyperspectral image classification: an overview. IEEE Trans. Geosci. Remote Sens. **57**(9), 6690–6709 (2019). https://doi.org/10.1109/TGRS.2019.2907932
17. Bera, S., Shrivastava, V.K., Satapathy, S.C.: Advances in hyperspectral image classification based on convolutional neural networks: a review. Comput. Model. Eng. Sci. **137**(1), 205–240 (2023). https://doi.org/10.32604/cmes.2023.028868
18. Kumar, V., Singh, R.S., Rambabu, M., Dua, Y.: Deep learning for hyperspectral image classification: a survey. Comput. Sci. Rev. **53**, 100633 (2024). https://doi.org/10.1016/j.cosrev.2024.100633
19. Ashraf, M., Zhou, X., Vivone, G., Chen, L.: Spatial-spectral BERT for hyperspectral image classification. Remote Sens. **16**(3), 539 (2024). https://doi.org/10.3390/rs16030539
20. Paszke, A., et al.: PyTorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, vol. 32, pp. 8024–8035. Curran Associates (2019). https://doi.org/10.48550/arXiv.1912.01703
21. Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., et al.: Array programming with NumPy. Nature **585**, 357–362 (2020). https://doi.org/10.1038/s41586-020-2649-2
22. Virtanen, P., et al.: SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat. Methods **17**, 261–272 (2020). https://doi.org/10.1038/s41592-019-0686-2
23. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
24. Hunter, J.D.: Matplotlib: a 2D graphics environment. Comput. Sci. Eng. **9**(3), 90–95 (2007). https://doi.org/10.1109/MCSE.2007.55

A Re-parameterized Multi-scale and Global Context Framework for Waste Detection in Vietnamese Street Scenes

Phat Hoang Ho[1,2], An Hoang Nguyen[1,2,3], Kien Trang[1,2,3], and Bao Quoc Vuong[1,2(✉)]

[1] School of Electrical Engineering, International University, Ho Chi Minh City 700000, Vietnam
vqbao@hcmiu.edu.vn
[2] Vietnam National University, Ho Chi Minh City 700000, Vietnam
[3] School of Information Technology, Monash University, Malaysia Campus, Subang Jaya 47500, Malaysia

Abstract. Roadside waste detection is important for smart environmental monitoring, but it remains difficult in Vietnamese street scenes because waste objects often appear with challenges such as high traffic density, diverse object scales, lighting conditions, and complex backgrounds. In this study, we propose a VGD-YOLO architecture specifically designed for complex Vietnam street scenarios. The proposed model integrates an enhanced backbone with RepHMS blocks to improve feature representation while maintaining computational efficiency, along with a refined detection head augmented with RepViT modules to strengthen global feature modeling and contextual understanding. By combining efficient structural re-parameterization and transformer-inspired representation learning, the proposed architecture enhances both localization accuracy and classification robustness. Experiments were conducted on a real-world Vietnamese roadside waste dataset of about 2,000 images covering four categories: plastic bottles, plastic bags, paper, and other waste. Results show that the proposed model outperforms several YOLO baselines. The final configuration achieves 0.487 precision, 0.447 recall, 0.433 mAP@50, and 0.290 mAP@50–95, demonstrating robust performance in complex roadside environments.

Keywords: YOLO · deep learning · ViT · street waste detection

1 Introduction

The plastic waste, often referred to as "white pollution," is one of the main threats to the global environment [1]. The severity of plastic waste is especially serious in Vietnam, driven by rapid industrialization, economic expansion, and rising consumerism [2]. According to the technical report from the World Bank [3], Vietnam discharged about 2.8 to 3.1 million tons of plastic waste annually on land, while discharging approximately 0.28 to 0.73 million tons of plastic waste to the oceans, resulting in it being one

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 413–427, 2026.
https://doi.org/10.1007/978-3-032-30491-9_26

of the five highest nations polluting the ocean [4]. In addition, approximately 10–15% of plastic waste is uncollected in urban areas and 45–60% in rural areas, while only 10% is properly managed or recycled [5]. The critical issue of plastic pollution in Vietnam requires multiple approaches to address the problem, including enforcing laws and decrees, increasing awareness, promoting recycling, and sorting and managing plastic waste through innovative technologies, etc.

In the context of Vietnam's urban and suburban roads, where waste objects vary significantly in size, shape, and appearance and are often partially occluded by traffic or surrounding structures, robust real-time detection is essential. The rapid development of Deep Learning (DL) research has enabled the deployment of DL models for plastic waste detection, enabling more efficient and scalable approaches in modern waste management and sustainability systems [6, 7]. On the other hand, You Only Look Once (YOLO)-based models have been widely applied to environmental monitoring and management systems, particularly for vision-based waste detection and classification [8–10]. In general, YOLO is an end-to-end object detection framework designed to achieve high accuracy while maintaining real-time performance. Its ability to accurately localize and classify objects in complex outdoor environments makes it well-suited for roadside trash analysis using street-level imagery.

Motivated by these advances, this study proposes an enhanced YOLO-based architecture, VGD-YOLO, specifically designed for roadside trash detection and classification in complex Vietnamese street environments, where objects often exhibit large-scale variations, irregular shapes, dense distributions, and cluttered backgrounds. The main contributions of this work are summarized as follows:

- We designed an improved YOLO backbone incorporating RepHMS modules that leverage structural reparameterization to enhance feature extraction while maintaining computational efficiency. This design improves robustness when dealing with visually complex roadside scenes.
- We enhance the detection head by integrating RepViT modules, enabling more effective global context modeling and long-range dependency capture through transformer-inspired mechanisms. This improvement is particularly beneficial for detecting small, low-contrast, and partially occluded waste objects.
- We evaluate the performance of the proposed model on an original Vietnam roadside trash dataset, against other baseline YOLO models. The experiment results demonstrate that the proposed architecture achieves superior performance across Precision, Recall, mAP 50, and mAP 50–95 metrics.

1. Related Work

Conventional approaches to the plastic waste detection problem rely on Convolutional Neural Network (CNN)-based models. Several models, including Xception, MobileNet, DenseNet, and InceptionResnet, were reported in [11] to classify the TrashNet dataset [12], with DenseNet121 and DenseNet169 achieving 95% accuracy after hyperparameter tuning. In another study [13], which implements the conventional TrashNet and ScrapNet datasets with additional plastic waste images, ResNet and EfficientNet were reported to achieve high accuracy up to 98%. Although the experimental

results from the above models are good, they are challenging for real-time applications, particularly in terms of inference time and computational resources.

In comparison to the CNN-based models, YOLO-based methods also achieve notable results in plastic waste detection and classification problems. In [9], the YOLOv7 framework is implemented for the real-time waste detection problem. The YOLOv7 architecture achieved 80.1% Precision, 86.8% mAP 50, and 61.8% mAP 50–95 on an in-house dataset. In another study [14], the proposed YOLO-VG model is based on the YOLOv8 framework for waste detection and classification. The YOLO-VG model replaces the conventional CNN architecture with the GSConv block to reduce redundant information, while introducing the ODConv module and the VoV-Ghost structure at the neck network to improve the model's adaptability and generalization. The YOLO-VG model scored 88.4% in mAP 50, achieving 24.6% better computational efficiency and reducing the architecture size by 20.4% compared to the original YOLOv8 structure. In study [8], the authors implement a YOLOv12-based framework for waste detection and classification on an in-house dataset. The YOLOv12-based model achieved 73% Precision and 78% mAP 50 after 100 training epochs, confirming its capability for waste management challenges. In study [10], the authors propose the WCD-YOLO model, based on the YOLOv10 backbone, with an additional MCA module to broaden the model's perceptual range and a FNC2f-BiFPN feature pyramid network to enhance detection of plastic waste targets. The WCD-YOLO models achieved 95.8% in mAP 50 and 74% in mAP 50–95 on an in-house dataset, with an extremely light architecture of 7.2 M parameters and GFLOPs of 8.5G.

From the above studies, it is clear that YOLO-based architectures have the potential to solve the waste detection problem, even in extremely complex street scenes in Vietnam, especially in later frameworks with more robust feature extraction and broader attention mechanisms. In this study, we propose the VGD-YOLO model, an YOLOv11-based framework for the roadsides waste detection problem. The experimental results indicate that the proposed model has outperformed conventional YOLO models and has advanced research in the field of waste detection and classification.

2 Methods

2.1 VGD-YOLO Architecture

In this study, the VGD-YOLO model is proposed for Vietnam roadsides trash detection, which has its own custom architecture, as depicted in Fig. 1. Similar to other YOLO versions, the proposed model consists of three main components: backbone, neck, and head. Based on the YOLOv11 framework, Re-parameterized Hierarchical Multi-Scale module (RepHMS) modules are introduced into the backbone to enhance multi-scale feature extraction while maintaining computational efficiency. Additionally, SPPF(Spatial Pyramid Pooling - Fast) and C2PSA (Convolutional block with Parallel Spatial Attention) modules are integrated at the end of the backbone to strengthen contextual representation and spatial attention. In the neck, feature fusion is performed using upsampling and concatenation operations combined with C3K2 (Cross Stage Partial with kernel size 2) blocks to obtain more effective multi-scale features. Furthermore, the Re-parameterized Vision Transformer (RepViT) module is incorporated before the

detection head to enhance global feature representation at deeper detection scales. The architecture of the proposed model is shown in the figure below.

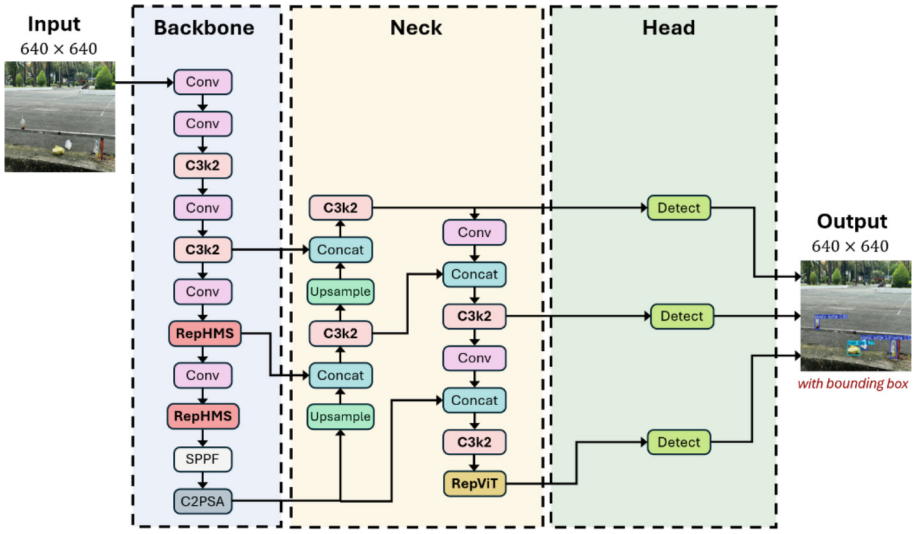


Fig. 1. Our proposed VGD-YOLO architecture

2.2 Re-parameterized Hierarchical Multi-scale Module

Framework in the proposed custom architecture, standard convolutional and C3K2 blocks are first employed to extract low- and mid-level features. To further enhance the high-level semantic representation before feature fusion, the RepHMS module is introduced at deeper stages of the backbone, as shown in Fig. 2.

RepHMS adopts a hierarchical multi-branch structure to jointly model shallow and deep features. The input feature map is first processed by a 1×1 convolution for channel adjustment and then split into multiple branches. Early branches preserve fine-grained spatial information, while deeper branches progressively extract richer semantic representations. Within each branch, feature transformation is performed using RepHDW (Re-parameterized Heterogeneous Depth-wise Convolution), which enables multi-scale spatial modeling through parallel depth-wise convolutions with heterogeneous kernel sizes. This design significantly enlarges the effective receptive field while maintaining computational efficiency.

To address the representational limitations of conventional depth-wise operations, RepHDW leverages a multi-branch training strategy that is analytically re-parameterized into a single equivalent depth-wise convolution during inference, ensuring no additional runtime overhead. The outputs of all hierarchical branches are then aggregated through feature concatenation and fused by a final 1×1 convolution, which restores channel interactions and scales the output channels to produce a compact yet expressive representation.

In the proposed architecture, RepHMS is strategically placed before the SPPF and C2PSA modules, ensuring that enriched, multi-scale, and semantically enhanced features are passed to both the spatial pyramid pooling and attention refinement stages. The overall structure of the RepHMS module is illustrated in Fig. 2, while the detailed formulation and re-parameterization mechanism of RepHDW are described in the following section.

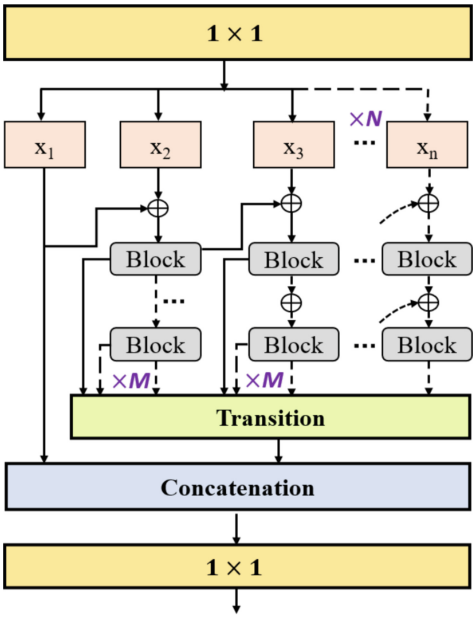


Fig. 2. RepHMS block

To be more specific, RepHDWConv is adopted as the fundamental convolutional operator within each hierarchical branch of the RepHMS module. This is achieved through a two-stage re-parameterization process. During training, it runs multiple depth-wise convolution branches in parallel, each with a different kernel size (e.g., 3×3, 5×5, 7×7). Each depth-wise convolution branch is followed by a batch normalization (BN) layer. Before merging kernels, BN must be absorbed into the convolution weights. The BN transformation is represented as Eq. (1):

$$BN\left(RepHDW\left(x\right)\right) = \gamma \cdot \frac{\left(RepHDW\left(x\right) - \mu\right)}{\sqrt{\sigma^2 + \varepsilon}} + \beta \tag{1}$$

where x denotes the input feature map of the RepHDWConv block, RepHDW(x) represents the output feature map produced by the depth-wise convolution branch before batch normalization. μ and σ indicate the accumulated mean and standard deviation of the batch normalization layer, respectively, while γ and β correspond to the learned scaling factor and bias. ϵ is a small constant introduced to ensure numerical stability.

Let W_{fused} and B_{fused}, which are Eqs. (2) and (3), respectively, denote the fused convolution weight and bias after absorbing the batch normalization parameters into the

depth-wise convolution kernel. We will get a formula as

$$W_{fused} = \gamma \cdot \frac{RepHDW}{\sqrt{\sigma^2+\varepsilon}} \tag{2}$$

$$B_{fused} = \gamma \cdot \frac{(-\mu)}{\sqrt{\sigma^2+\varepsilon}} + \beta \tag{3}$$

After fusing with the BN layers, the convolution will be represented as Eq. (4):

$$BN(RepHDW(x)) = W_{fused(x)} + B_{fused} \tag{4}$$

In the second phase, the heterogeneous depth-wise convolutions must be merged into a single convolution, resulting in a feature map output. The formula of the feature map will be represented as Eq. (5):

$$O = I \otimes \left(K_{2n-1} + \sum_{i=1}^{m} K_{2n-(2i+1)} \right) + \left(B_{2n-1} + \sum_{i=1}^{m} B_{2n-(2i+1)} \right) \tag{5}$$

where I represents the input feature map of the re-parameterized convolution during inference, and O denotes the corresponding output feature map. K_n and B_n refer to the weight and bias of a depth-wise convolution with an $n \times n$ kernel, respectively, and $\otimes$ indicates the depth-wise convolution operation. Thus, n $\geq$ 3 corresponds to the largest kernel size used in the heterogeneous depth-wise convolution, while m is the largest integer satisfying $2n - (2m + 1) \geq 3$.

2.3 RepViT Block for Improving Neck Structure

RepViT is a re-parameterized vision Transformer architecture originally designed as a hierarchical feature extractor, where an initial convolutional stem reduces spatial resolution, followed by multiple stages that progressively extract higher-level semantic features through Transformer-based blocks, as shown in Fig. 3. Across these stages, RepViT leverages self-attention to model long-range dependencies while employing structural re-parameterization to significantly improve backbone and memory overhead during inference.

In the RepViT architecture, feature extraction begins with a 3 $\times$ 3 convolutional stem that reduces the spatial resolution of the input image. The network then proceeds through multiple hierarchical stages (Stages 1–4), each consisting of one or more RepViT blocks. Downsampling operations are applied between consecutive stages to progressively reduce spatial dimensions while increasing channel depth, allowing the model to concentrate on higher-level semantic features while keeping computational complexity moderate.

By applying RepViT in this manner, the model benefits from Transformer-style global dependency modeling and enhanced feature representation, while preserving the lightweight and end-to-end detection efficiency of YOLO models. This partial integration strategy allows RepViT to strengthen high-level feature interactions before the detection head, improving robustness in complex scenes without altering the original backbone structure.

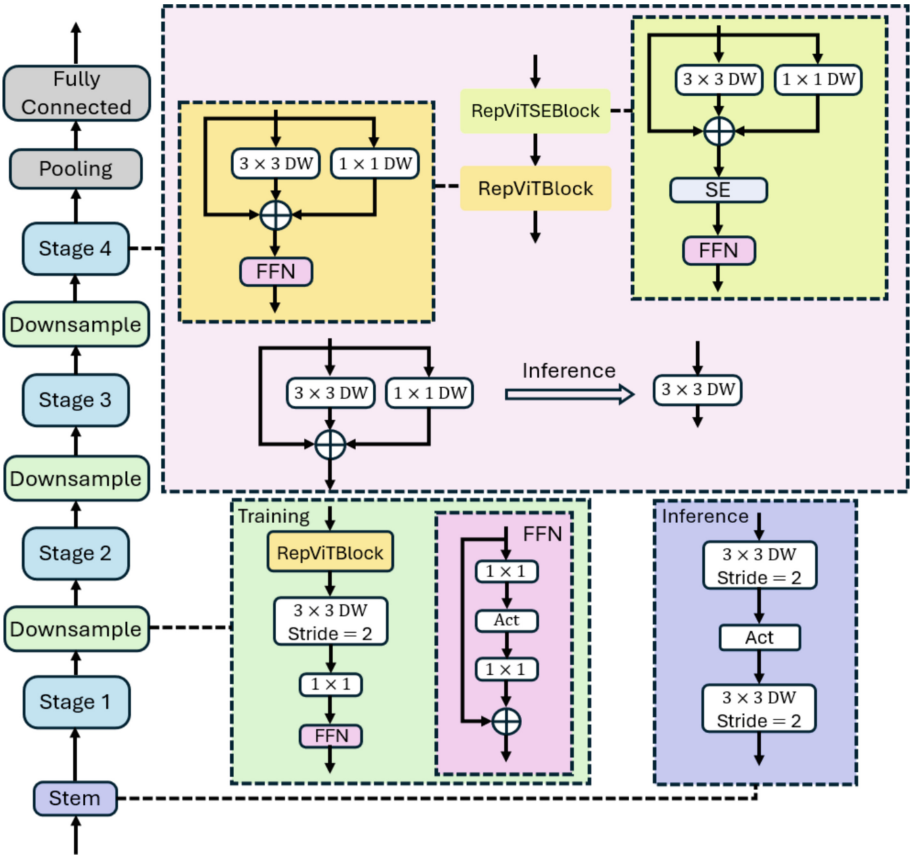


Fig. 3. RepViT architecture

3 Experiments and Results

3.1 Dataset

The dataset consists of approximately 2,000 waste images collected from real-world roadside environments in Vietnam. The images were captured across multiple locations, including Ho Chi Minh City and several other provinces, ensuring diversity in background, rain conditions, and scene complexity. The dataset contains 4 types of waste commonly found in roadside scenarios: plastic bottles, plastic bags, paper, and other types (fruit, masks, etc.). All classes were manually annotated to provide accurate ground-truth labels for model training and evaluation. This diverse data collection supports robust learning and improves the generalization capability of the proposed YOLO-based detection model. In each class, plastic bottles have 1606 instances, plastic bags have 1705, paper has 580, and others have 905, respectively.

3.2 Implementation Setting

All experiments were conducted in a Python 3.10.19 environment with PyTorch 2.5.1, accelerated by CUDA 12.1. The model was trained on a computer equipped with an NVIDIA GeForce RTX 3060 GPU with 12 GB of VRAM. The input images were resized to 640 × 640 pixels before training. The training epoch and batch size are set at 250 and 12, respectively. In addition, the ratio of the training and validation sets is 80:20, respectively.

3.3 Assessment Metrics

To comprehensively evaluate the performance of the proposed framework, several standard object detection metrics are used, including precision, recall, mAP@50, and mAP@50–95. Among these, mAP@50 and mAP@50–95 are considered the primary metrics for comparison with baseline models and for conducting ablation studies.

Precision measures the proportion of correctly predicted positive samples among all predicted positives and is defined as:

$$\text{Precision} = \frac{TP}{TP+FP} \tag{6}$$

where *TP* denotes true positive detections and *FP* represents false positive detections.

Recall evaluates the model's ability to identify all relevant objects in the dataset and is expressed as:

$$\text{Recall} = \frac{TP}{TP+FN} \tag{7}$$

where *FN* refers to false negatives, i.e., objects that are present but not detected by the model.

The metric mAP@50 represents the mean Average Precision computed at an Intersection-over-Union (IoU) threshold of 0.5, where a detection is considered correct if the IoU between the predicted and ground-truth bounding boxes is at least 50%. In contrast, mAP@50–95 is calculated by averaging the AP values over multiple IoU thresholds ranging from 0.5 to 0.95.

$$\mathbf{mAP_{50-95}} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{K} \sum_{i=1}^{N} Ai, j \tag{8}$$

$$\mathbf{mAP_{50}} = \frac{1}{N} \sum_{i=1}^{N} AP_{50}, i \tag{9}$$

where N denotes the total number of object categories, and $AP_{50,\,i}$ represents the Average Precision of the i-th category evaluated at an IoU threshold of 0.50; K indicates the number of IoU thresholds used for evaluation, which is set to 10 and ranges from 0.50 to 0.95 with a step size of 0.05. $AP_{i,\,j}$ denotes the Average Precision of the i-th category computed at the j-th IoU threshold. The Average Precision is obtained from the precision–recall curve, which is determined by the number of true positives (TP), false positives (FP), and false negatives (FN) at different confidence levels.

3.4 Quantitative Results

Figure 6 shows the training progress with four evaluation metrics of our model. Almost all values reached the saturation point and fluctuated less after 100 epochs. Only mAP 50–95 reaches around 0.25, while the other metrics climb above 0.4. To evaluate the effectiveness of the proposed architecture, the VGD-YOLO model achieves the best overall detection performance among the compared methods, particularly in mAP@50 and mAP@50–95, reaching 0.433 and 0.290, respectively. These improvements aim to enhance localization accuracy and robustness across multiple IoU thresholds by using Re-parameterized Hierarchical Multi-Scale (RepHMS) modules in the backbone and the RepViT block in the neck.

In addition, RepViT is a lightweight transformer-based attention, enabling the model to capture long-range contextual dependencies. The figure below shows the comparison of the models' performance.

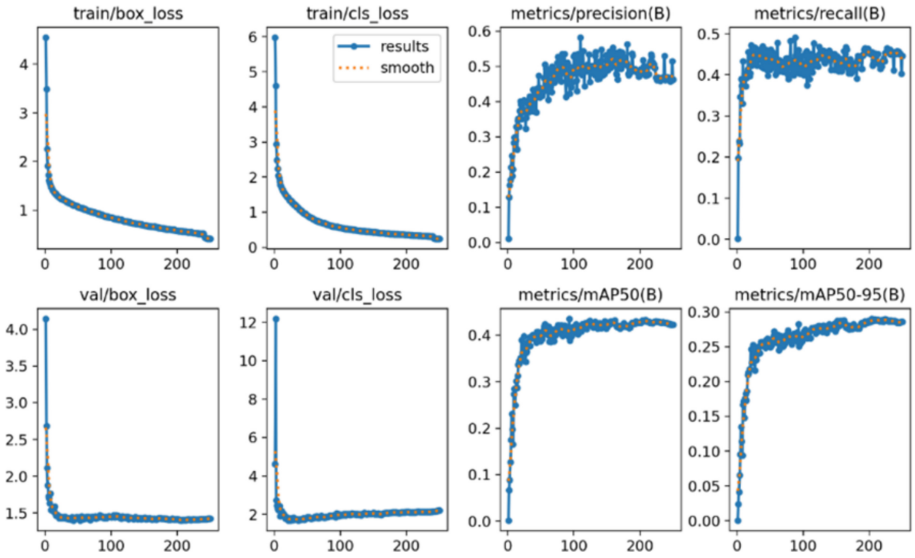


Fig. 4. The training progress of the proposed model.

Table 1. The comparison of performance between YOLO baseline and VGD

Model	Precision	Recall	mAP@50	mAP@50–95
YOLO11 [15]	0.474	0.432	0.417	0.275
YOLOv10 [16]	**0.492**	0.415	0.415	0.281
YOLOv9 [17]	0.462	**0.455**	0.422	0.280
YOLOv8 [18]	0.486	0.432	0.419	0.276
VGD-YOLO (ours)	0.487	0.447	**0.433**	**0.290**

3.5 Qualitative Results

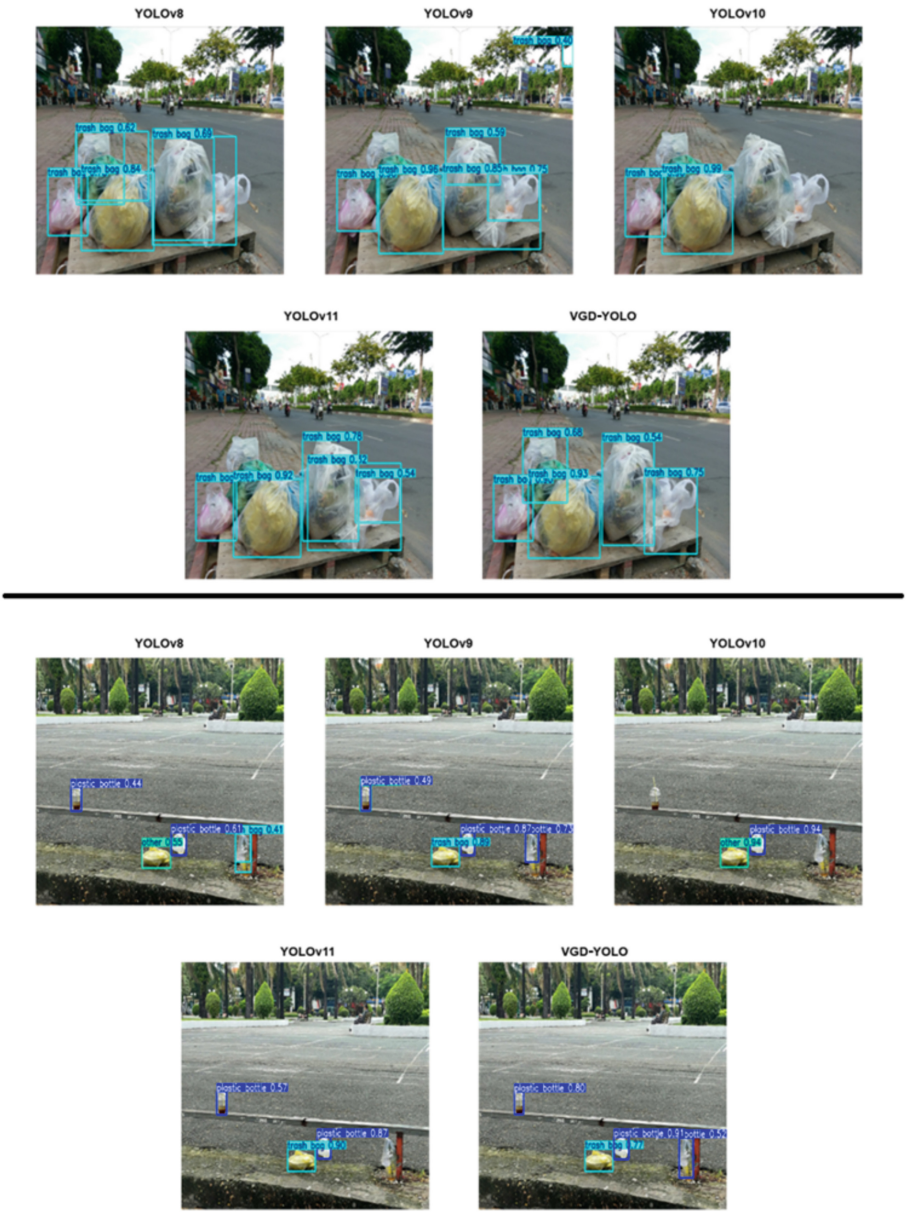


Fig. 5. The comparison of visualization between different models in daytime ideal conditions

In addition to the quantitative evaluation, we also perform detection experiments using other models to qualitatively compare their performance with that of the proposed framework. The visualization result of the proposed model in Fig. 5 shows that the custom model successfully detected all objects in the images. However, the variants of YOLO models missed the plastic bottles, which were shown in YOLOv10 and YOLOv11. Moreover, the YOLOv8 detected incorrect classes with duplicated problems and has relatively low confidence in true positive results. Similarly, in the trash bags class detection, the custom model shows superior detection capabilities than the YOLO variant model in terms of roadsides scenarios, whereas YOLO variants still struggle with the duplicated problems and missing objects in the ideal daytime conditions.

In the nighttime scenario, the proposed model still shows better performance, with a high confidence score in Fig. 6. According to the visualization results, all YOLO-based variants incorrectly detect plastic bottles, whereas VGD achieves high accuracy with a 0.76 confidence score.

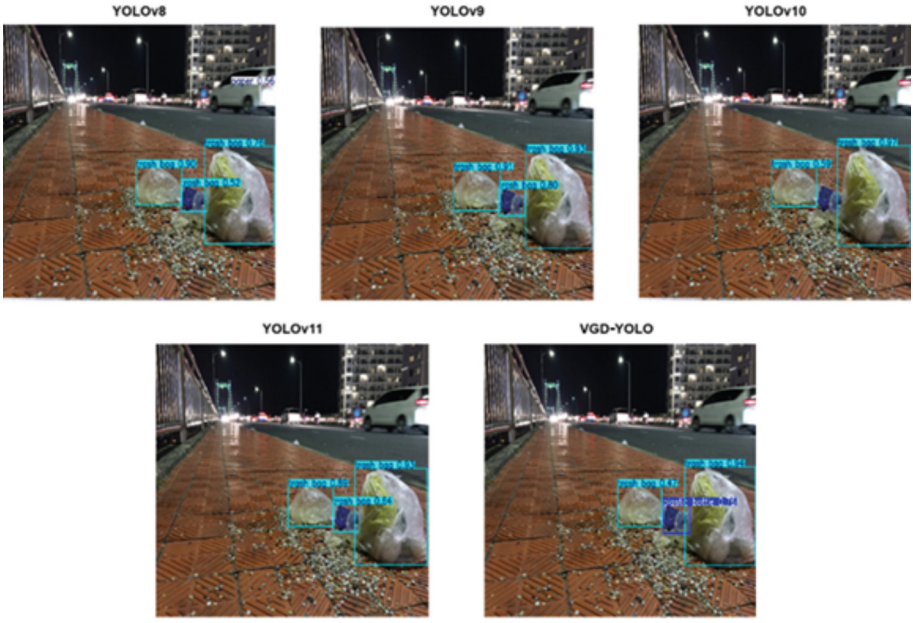


Fig. 6. The comparison of visualization between different models in nighttime ideal conditions

From the visual results of daytime non-ideal conditions, YOLOv11 can detect paper objects more correctly than the other model versions in Fig. 7. However, all models of YOLO detect the large trash bag object incorrectly instead of papers due to its white color. Overall, the proposed YOLO custom model can still detect small objects better than the other YOLO versions.

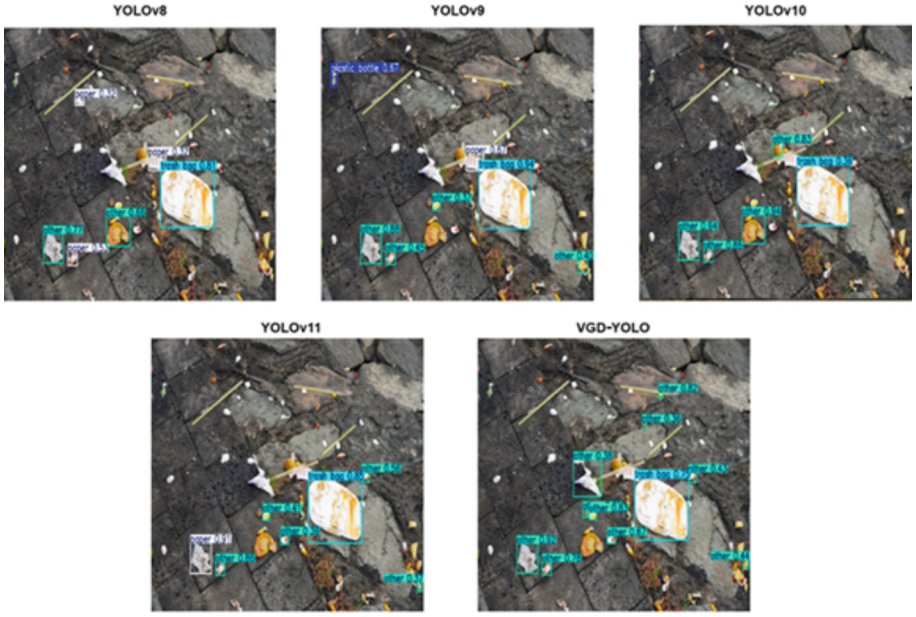


Fig. 7. The comparison of visualization between different models in daytime non-ideal conditions

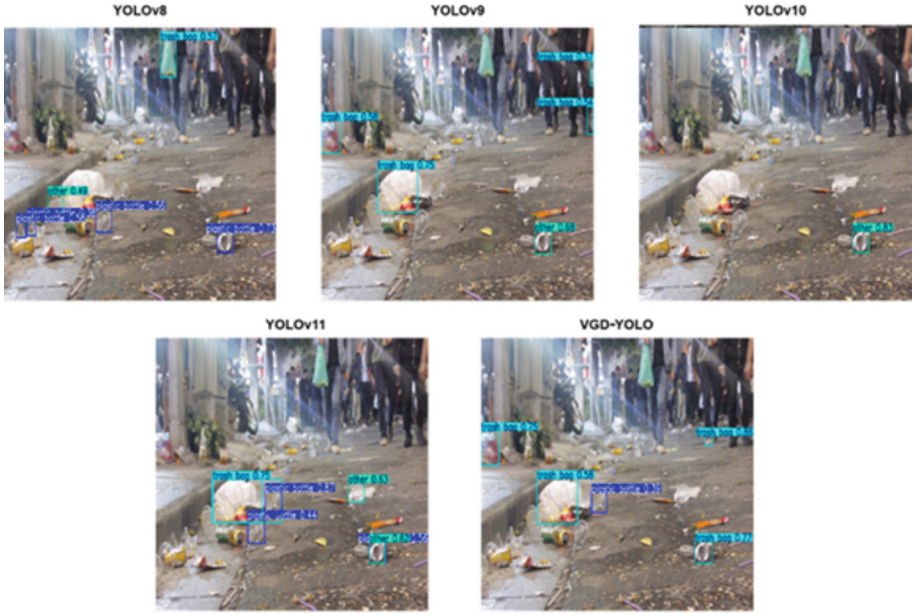


Fig. 8. The comparison of visualization between different models in nighttime non-ideal conditions

From the visual results of nighttime non-ideal conditions, YOLOv11 has superior detection in terms of plastic class. However, when detecting small objects, the VGD model shows better performance compared to YOLO based line, which is shown in Fig. 8.

3.6 Ablation Studies

In order to provide the comparative effects of implemented settings, we conduct several sets of ablation studies.

Effect of Feature Extraction Module
Based on the results shown in Table 2, the RepHMS module was selected for integration into the proposed framework. The table presents a performance comparison between several backbone block variants, including C3k2, C2f/C2fCIB, RepNCSPELAN4, and C2f. Among these alternatives, RepHMS achieves the most balanced and competitive performance across the evaluation metrics. Specifically, RepHMS obtains the highest mAP@50 of 0.432 and the best mAP@50–95 score of 0.284, indicating improved detection capability across different IoU thresholds. In terms of precision, RepHMS reaches 0.487, which is comparable to the highest precision achieved by C2f/C2fCIB (0.492), while maintaining a strong recall of 0.428. This balance between precision and recall contributes to the improved overall detection accuracy reflected in the mAP scores.

Table 2. The comparison of backbone methods between different blocks

Block	Precision	Recall	mAP@50	mAP@50–95
C3k2	0.474	0.432	0.417	0.275
C2f/C2fCIB	0.492	0.415	0.415	0.281
RepNCSPELAN4	0.462	0.455	0.422	0.280
C2f	0.486	0.432	0.419	0.276
RepHMS	**0.487**	**0.428**	**0.432**	**0.284**

Effect of Attention Mechanisms
Both proposed models benefit from the introduction of RepHMS, while Model B, which is our final proposed model, further improves performance by integrating RepViT on top of RepHMS. Model A, which uses only RepHMS, already achieves strong detection accuracy with an mAP@50 of 0.432 and an mAP@50–95 of 0.284, confirming that RepHMS effectively enhances local feature extraction and multi-scale representation. By adding RepViT, Model B achieves the best overall performance, reaching an mAP@50 of 0.433 and an mAP@50–95 of 0.290. Although the increase in mAP@50 appears modest, the mAP@50–95 shows a clear improvement, indicating that RepViT significantly improves precise bounding box localization, especially under stricter IoU thresholds. Both models maintain the same precision of 0.487, but Model B improves recall from 0.428 to 0.447, meaning that the RepViT-enhanced architecture detects more

waste objects without increasing false positives. This confirms that RepViT contributes effective global context modeling and long-range feature interaction, complementing RepHMS's strong local and multi-scale representations.

Table 3. The comparison of performance between model A **(Only RepHMS)** and model B **(RepHMS + RepVit)**

Model	Precision	Recall	mAP@50	mAP@50–95
Only RepHMS	0.487	0.428	**0.432**	**0.284**
RepHMS + RepVit	**0.487**	**0.447**	**0.433**	**0.290**

4 Conclusion

This paper presented the VGD-YOLO framework for roadside waste detection in complex Vietnamese street environments. The proposed model integrates RepHMS into the backbone to strengthen efficient multi-scale feature extraction and incorporates RepViT into the neck to enhance global contextual modeling and long-range feature interaction. Experimental results on a real-world Vietnamese roadside waste dataset show that VGD-YOLO achieves the best overall performance compared with several YOLO baselines. In particular, the final model reaches 0.487 in precision, 0.447 in recall, 0.433 in mAP@50, and 0.290 in mAP@50–95, demonstrating improved robustness and localization accuracy in challenging roadside scenes. Thus, the proposed framework shows promising potential for practical roadside waste monitoring and intelligent environmental management in Vietnam.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this paper.

References

1. Andronicеanu, A., Dragulanescu, I.-V.: A survey on the buyers' eco-responsibility and the urban white pollution. Environ. Eng. Manag. J. **15**, 481–487 (2016). https://doi.org/10.30638/eemj.2016.051
2. Salhofer, S., Jandric, A., Soudachanh, S., Le Xuan, T., Tran, T.D.: Plastic recycling practices in Vietnam and related hazards for health and the environment. Int. J. Environ. Res. Public Health. **18**, 4203 (2021)
3. World Bank: Vietnam: Plastic Pollution Diagnostics. World Bank, Washington, DC (2022)
4. Jambeck, J.R. et al.: Plastic waste inputs from land into the ocean. Science. **347**(6223), 768–771 (2015). https://doi.org/10.1126/science.1260352
5. World Bank: Toward a National Single-use Plastics Roadmap in Vietnam: Strategic Options for Reducing Priority Single-Use Plastics. World Bank, Washington, DC (2022)
6. Nahiduzzaman, M., et al.: An automated waste classification system using deep learning techniques: toward efficient waste recycling and environmental sustainability. Knowl.-Based Syst. **310**, 113028 (2025). https://doi.org/10.1016/j.knosys.2025.113028

7. Bobulski, J., Kubanek, M.: deep learning for plastic waste classification system. Appl. Comput. Intell. Soft Comput. **2021**, 6626948 (2021). https://doi.org/10.1155/2021/6626948
8. Dipo, M.H. et al.: Real-time waste detection and classification using YOLOv12-based deep learning model. Digital. **5**(2), 19 (2025). https://doi.org/10.3390/digital5020019
9. Rahmatulloh, A., Darmawan, I., Aldya, A.P.: Firmansyah Maulana Sugiartana Nursuwars.: WasteInNet: deep learning model for real-time identification of various types of waste. Cleaner Waste Systems. **10**, 100198 (2025). https://doi.org/10.1016/j.clwas.2024.100198
10. Ling, L., Chen, Y.: Zhiwu li.: WCD-YOLO: a waste classification detection model. J. Environ. Manag. **398**, 128601 (2026). https://doi.org/10.1016/j.jenvman.2026.128601
11. Aral, R.A., Keskin, Ş.R., Kaya, M., Hacıömeroğlu, M.: Classification of TrashNet Dataset Based on Deep Learning Models. In: 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, pp. 2058–2062, IEEE (2018), doi:10.1109/BigData.2018.8622212
12. Thung, G., Yang, M.: Classification of Trash for Recyclability Status (2016).
13. Masand, A., Chauhan, S., Jangid, M., Kumar, R., Roy, S.: Scrapnet: an efficient approach to trash classification. IEEE Access. **9** (2021). https://doi.org/10.1109/ACCESS.2021.3111230
14. Song, L., Yu, H., Yang, Y., et al.: YOLO-VG: an efficient real-time recyclable waste detection network. J. Real-Time Image Proc. **22**, 79 (2025). https://doi.org/10.1007/s11554-025-01655-5
15. Glenn, J.: YOLOv11. GitHub. (2024) https://github.com/ultralytics/ultralytics/releases/tag/v8.3.50
16. A. Wang et al.: YOLOv10: Real-Time End-to-End Object Detection. arXiv (2024). https://arxiv.org/abs/2405.14458
17. C.-Y. Wang, I.-H. Yeh, H.-Y. M. Liao.: YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. arXiv (2024). https://arxiv.org/abs/2402.13616
18. Jocher, G., Qiu, J., Chaurasia, A.: YOLOv8. Github (2023) https://github.com/ultralytics/ultralytics

LCRO-GAN: A Multi-stage GAN Framework for Image-Based Reconstruction of Historical Monuments

Pranav Jigalur(✉), Likhith Gowda, Kripa S Rai, Praveen Koppal, and Sindhu R Pai

Department of Computer Science and Engineering, PES University, Bengaluru, India
pranav.jigalur@gmail.com, sindhurpai@pes.edu

Abstract. The virtual reconstruction and preservation of historical monuments is a critical problem in digital archaeology, especially when cultural heritage images contain damaged or missing regions. Existing image inpainting methods often struggle to restore such regions with both photorealistic quality and structural fidelity, limiting their usefulness for heritage reconstruction. This paper presents an efficient AI-based approach for achieving structural consistency in the reconstruction of damaged monument images. Specifically, a sequential multi-stage generative adversarial network (GAN) framework is proposed to restore missing regions while maintaining architectural structure. Furthermore, a novel masking strategy, called Largest Contour-based Regional Occlusion (LCRO), is introduced to generate architecturally realistic damage patterns for training and evaluation. Accordingly, the proposed framework, termed LCRO-GAN, is designed as a robust three-stage GAN pipeline for structurally consistent monument image reconstruction. The pipeline performs coarse reconstruction in the first stage, followed by high-resolution inpainting to recover fine architectural details, and a final refinement stage to enhance visual consistency and generate seamless outputs. Experimental results show that LCRO-GAN achieves a PSNR of 28.7 dB, an SSIM of 0.827, and an FID score of 11.5, indicating the effectiveness of the multistage approach for photorealistic image restoration. The findings further demonstrate that the combination of structured masking and multi-stage adversarial learning improves both structural consistency and reconstruction accuracy. Overall, the proposed framework offers a scalable approach for digital heritage restoration and serves as a useful tool for monument preservation projects.

Keywords: Generative Adversarial Networks (GAN) · LCRO Masking (Largest Contour-based Regional Occlusion) · Multi-stage GAN Framework · Peak Signal-to-Noise Ratio (PSNR) · Fréchet Inception Distance (FID) · Structural Similarity Index (SSIM)

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 428–444, 2026.
https://doi.org/10.1007/978-3-032-30491-9_27

1 Introduction

Historic monuments showcase concrete connections to our cultural heritage and they represent centuries of architectural achievements, artistry, and historical significance [1,2]. Consequently, digital reconstruction and conservation have become increasingly significant in India, as many ancient temples, forts, and archaeological monuments continue to deteriorate due to natural disasters, environmental pollutants, and human interference [3,4]. In normal images, local texture patterns are adequate for realistic image inpainting; however, for architectural reconstruction, it is necessary to follow strict patterns of structure, symmetry, and cultural design motifs [5]. Conventional image inpainting methods often fail to deliver photorealistic reconstruction and preserve structural consistency.

Significant improvements in the performance of deep learning techniques have been reported in the field of image synthesis and reconstruction with the help of Generative Adversarial Networks (GANs) [6]. Single-image restoration solutions that comprise multi-scale attention mechanisms and deep image priors have also shown promising results [7,8]. However, the performance of single-stage GAN models seems to suffer from difficulties in preserving architectural details in the reconstructed images. Single-stage GAN models typically struggle to preserve fine-grained architectural details and often introduce artifacts when applied to complex structural imagery [9]. In addition to this, existing masking strategies are simplistic, thus failing to generate realistic deterioration patterns observed in damaged monuments, which can limit their effectiveness in real-world restoration approaches [10]. Masking plays a crucial role in image reconstruction and inpainting tasks, as it defines the regions of missing or corrupted information that the model must restore. These techniques lack structural and semantic awareness, removing regions without considering underlying spatial structure or architectural importance of the scene. Such indiscriminate masking strategies can obscure essential structural elements, thereby producing reconstructions that are visually inconsistent and architecturally unrealistic. Existing studies on occlusion handling in deep models also reveal that these simple masking strategies have several limitations on theory and real-world applications [11]. These limitations highlight the need for more structure-aware masking techniques, since the final restoration depends largely on the masking approach.

This research attempts to bridge the gap in the digital reconstruction of ancient monuments and sculptures by proposing *LCRO-GAN*, a novel multi-stage Generative Adversarial Network framework designed specifically for Dravidian-style monument reconstruction. In contrast to conventional inpainting methods that use generic free-form masking, the proposed framework incorporates a *Largest Contour-based Regional Occlusion (LCRO)* masking technique to generate realistic, context-aware damage patterns that more closely resemble natural architectural deterioration. The LCRO-GAN pipeline follows a progressive restoration strategy. The initial stage creates a coarse structural foundation, followed by intermediate texture refinement for structural consistency, and a final stage that enhances high-resolution details by coordinating the use of multiple generative models.

In order to test and validate the proposed framework, a dataset of more than 20,000 high-resolution images of Indian temples and monuments were collected. The dataset was compiled using images obtained from multiple open-access, royalty-free image repositories. The images were further refined through systematic preprocessing and augmentation. Experimental evaluation shows that the proposed multi-stage framework achieves competitive PSNR, SSIM, and FID scores, highlighting its capability to produce photorealistic and structurally consistent restorations. Overall, the proposed approach provides a robust and scalable solution for the digital preservation of historical monuments and establishes an effective data-driven framework for cultural heritage reconstruction.

The remainder of this paper is structured as follows: Sect. 2 reviews related work on image inpainting and GAN-based reconstruction approaches. Section 3 describes the process of dataset creation. The detailed description of the proposed LCRO masking strategy is discussed in Sect. 4. The LCRO-GAN architecture and training methodology are presented in Sect. 5. Experimental results and comparative analysis are covered in Sect. 6. Finally, Sect. 7 concludes the paper and outlines directions for future research.

2 Related Work

The researches related to the domain of image processing and inpainting using generative AI has seen significant shift from standalone conditional GANs towards advanced multi-stage pipelines that decomposes the image reconstruction process into manageable pieces [6]. This improvement serves the requirements of architectural restoration, where the global structure, local texture, and semantic consistency needs to be maintained simultaneously. Techniques that combines pre-processing or multimodal strategies can assist in enhancing restoration and analysis [12].

Pix2Pix conditional GAN model has been used widely for image-to-image reconstruction tasks, especially for creating binary damage masks and guided inpainting. Rakhimol and Uma Maheswari demonstrated that through the application of conditional GANs for automatic mask generation and partial convolution networks for temple murals restoration, improving PSNR, MSE, and SSIM values can be made by greater extents [13].

The variants of DCGAN are simple models for texture synthesis in image restoration tasks. Recently, the application of attention and structural priors to DCGAN architecture has been done to enhance the restoration model's capacity to handle geometric complexity of architectural structures.

CycleGAN has also been utilized for style transfer and colorization in heritage restoration, enabling unpaired learning scenarios where the ground truth of the restored images is not available [14]. This is beneficial for the Indian temple restoration task because of the scarcity of information about the structures before damage.

Devalla and Arya proposed a GAN architecture using gated convolution and contextual attention layers that were trained on Indian temple and sculpture

images. Their method obtained an L1 loss of 0.29 after prolonged training, showing improved free-form missing region handling [15].

For temple murals of Tamil Nadu, Sumathi and Uma Devi proposed an edge- and line-guided diffusion patch GAN (WSFN + SCN + DIMD) that could restore large degraded regions. Their approach recorded stunning metrics: SSIM 0.8853, PSNR ∼29.88, MSE 0.0021, and LPIPS 0.0426 on proprietary degraded mural datasets.

Multiple works have blended modified U-Net damage detection with style transfer for priors in texture and PartialConv inpainting to produce perceptually coherent completions on Rajasthani painting images [16].

Previous research was dedicated to automatic mask generation blended with exemplar-based region-filling algorithms for both structure and texture reconstruction in Indian temple murals, with assessment using PSNR, MSE, and SSIM compared to conventional techniques [10].

Triple GAN frameworks have more recently been suggested whereby there are distinct networks dealing with varying aspects of restoration (structure, texture, refinement), and each step utilizes the output of the preceding step. Several works employed coarse-to-refinement pipelines where a coarse network produces crude completions, which are then fine-tuned by expert networks dedicated to texture and detail refinement. Sophisticated pipelines distinguish structural reconstruction from content creation, applying structure-specific networks to ensure architectural consistency while content networks address texture synthesis. The researchers have employed multiple discriminators operating at different scales (global + local/patch), effectively addressing both the macro-structural and micro-textural features in detail, to make training more stable and realistic [1].

The literature details significant advancements in GAN-based in-painting of Indian temples using techniques like structure-content decomposition, multi-discriminator architectures, and gated convolutions. However, there are still significant issues with standardization, expert verification, practical implementation, and cross-cultural transfer. To ascertain historical authenticity, future research will need expert-guided evaluation standards and high-quality labeled datasets from Indian temples.

3 Dataset Creation

This section describes the methodology followed for building the dataset used in this research, including data collection, preprocessing, deduplication, and measures taken to enhance data quality and diversity.

3.1 Data Acquisition

Data Sourcing and Retrieval. The dataset employed in this paper was assembled by the authors using web scraping and API-based retrieval, followed by manual curation from publicly available, copyright-free image repositories, including Pexels, Unsplash, and Pixabay, that support royalty-free usage for academic

research [17,18]. Libraries like `requests` and `BeautifulSoup` has been used to navigate web pages and identify image URLs. To obtain a broad range of images, keywords for specific temple names, common terms like "Dravidian style", "Indian temple architecture" and certain kingdom names like "The Vijayanagara Kingdom", "The Rastrakutas" has been used. This method gathered a diverse set of images exhibiting the diversity of Indian cultural heritage.

Initial Data Preprocessing. The image URLs were gathered and stored in a traceable JSON format. The subsequent phase included downloading and verification of each of the images. A python script was created for downloading the images, in which multithreading was implemented for network latency management as well as increased efficiency. Each image that was downloaded went through two validation steps:

- *Automated Checks:* The integrity of the files was checked by scripts to ensure that all the images were correct and maintained in their original files. This also optimized the image format in order to be successfully integrated into the training process.
- *Manual Filtering:* The manual screening was done to eliminate unnecessary or poor images. The images of modern buildings, irrelevant objects, low-resolution images that did not contribute to the learning process in the model were eliminated. This step was necessary to make the dataset of quality and integrity.

3.2 Dataset Refinement and Augmentation

Deduplication. To reduce the risk of model overfitting and potential dataset bias, a thorough deduplication process has been carried out before training. This has been developed through a hash-based file comparison technique to detect and eliminate similar images, regardless of file name or origin. Automated methods may not always capture near-duplicates or slight variations. Therefore, the dataset was also reviewed manually to identify and remove such cases. After completing this cleaning process, the dataset consisted of 19,998 distinct images.

Data Augmentation. Even though the amount of original images was also considerable, the diversity of the architecture styles was in the nature of the dataset which made it difficult to train an efficient model. This issue was addressed by significantly expanding the dataset through data augmentation techniques, which not only increased the dataset size but also enhanced the model's ability to generalize. These augmentations created modified versions of the existing images, artificially increasing the dataset size six-fold to a total of 119,936 images. The types of augmentations applied are:

- *Geometric Transformations:* Horizontal and vertical flips and rotations of 90°, 180°, and 270° has been performed. This brought in different perspectives and orientation to the model in such a way that it would be more resistant to different orientations.

– *Photometric Adjustments:* Images' brightness and contrast were changed slightly (within a ±10% range). This mimicked a range of illumination conditions and ensured the model that was trained was more robust against variations in environmental conditions.

Prior to feeding the images into our model, one final preprocessing step had to be performed. All images were resized to a standard resolution of 500×500 pixels. This was important for computational efficiency and consistent model training. Free-form masks were applied to the images subsequently. This is simulating damage and forming the "broken" images which our model has been trained to fix. The free-form topology of the masks avoids overfitting the model to the shape of damage and requires it to learn to complete missing information from context. All images used in this work were obtained from open-access platforms that allow free use for research and academic purposes. No copyrighted or restricted images were included in the dataset. The final curated dataset was generated solely for experimental evaluation of the proposed LCRO-GAN framework.

4 Data Preprocessing

The performance of a deep learning model, in this case, a Generative Adversarial Network (GAN) for an image-to-image translation task such as architectural reconstruction, rests to a large extent on the realism and quality of its training dataset. Towards this end, having a paired dataset of images of undamaged, original monuments and their synthetically damaged counterparts is crucial. Although there are many ways to create image masks, generic approaches are not adequate for domain specific architectural reconstruction.

4.1 Rationale for a Custom Masking Strategy: Limitations of Conventional Methods

Throughout the image inpainting domain in general, there are a number of masking techniques that are widely utilized. They can be generally classified as:

– *Geometric Masks:* These are simple, regular shapes like rectangles or circles used to mask out portions of an image. Although easy to produce, they are structure-naive and fail to model realistic patterns of damage on intricate structures such as monuments (Fig. 1a).
– *Irregular (Free-form) Masks:* These techniques, typically involving random walks or "brush stroke" generation, form masks with irregular edges. While they produce more detail than geometric masks, they are nonetheless image content agnostic. The occlusions they produce are random and most unlikely to be a whole architectural element (e.g., a dome, a spire, or the whole facade) (Fig. 1b).
– *Noise-Based Masks:* Methods such as introducing random pixel noise (e.g., salt-and-pepper) emulate corruption to the signal instead of the global physical deterioration typical of deteriorating monuments (Fig. 1c).

The major drawback of such traditional methods is that they are semantically oblivious. The structural deterioration of monuments is seldom random; it is usually in large, connected areas that relate to specific architectural features. A fallen roof, a ruined wall, or a weathered facade represents a region-based and structurally meaningful loss of information.

A model learned from images of easily dissectible rectangular holes might be very good at inpainting textures and lines but would probably not convincingly reconstruct complicated, non-rectilinear features having distinctive architectural grammar. The mask itself needs to tell the model what kind of damage it is supposed to fix. Hence, to learn a model for authentic architectural restoration, a damage simulation strategy is needed that emulates damage in semantically and structurally meaningful ways. A new unsupervised machine learning solution is specifically designed for this task.

(a) Geometric Mask (b) Free-form Mask (c) Noise-based Mask

Fig. 1. Comparison of conventional masking techniques.

4.2 The Proposed Technique: Largest Contour-Based Regional Occlusion (LCRO)

To alleviate the above-mentioned drawbacks, the Largest Contour-based Regional Occlusion (LCRO) technique has been introduced. The main goal of LCRO is to detect and completely mask the most salient object in an image—presumed to be the monument—thus mimicking extensive, realistic structure damage instead of superficial, haphazard holes. This unsupervised method utilizes a pipeline of conventional computer vision operations. The result of this process is shown in Fig. 2.

Edge Detection for Structural Boundary Identification. The process begins by converting the input BGR image (I) into a single-channel grayscale representation, I_{gray}. This reduces computational complexity and narrows the analysis to luminance gradients, which are the major indicators of structural form. After this, the Canny edge detection algorithm is applied. This is a critical step for distinguishing the monument's silhouette and primary structural

lines from the background and internal textures. The thresholds of the algorithm, $\theta_{\text{low}} = 50$ and $\theta_{\text{high}} = 150$, were selected empirically to strongly map these boundaries over our dataset. The outcome is a binary edge map, E.

Dominant Object Segmentation via Contour Analysis. With the use of the edge map E, a set C containing all unique outer boundaries will be obtained, with every boundary c forming the boundary of a shape detected.

$$C = \{c_1, c_2, ..., c_n\} = \text{findContours}(E) \tag{1}$$

The LCRO algorithm uses the central heuristics for segmentation. It is supposed that, in a well-composed photograph of a monument, the visual salient subject is the monument. Thus, it will be framed by the largest area with the profile. This maximum contour, $c_{\max}$, was found by enumerating the set C and choosing the one with largest area.

$$c_{\max} = \underset{c \in C}{\text{argmax}}(\text{Area}(c)) \tag{2}$$

This simple intuitive method allows us to do sound, unsupervised foreground segmentation that is application-specific without the need to use computationally expensive deep learning segmentation models.

Mask Generation and Synthetic Occlusion. The main contour $c_{\max}$ has been split off to form a binary mask M. This is accomplished by initializing a zero-(black) image and drawing in white (pixel value 255) the filled contour over it. Thus a neat clean mask will be obtained that covers the principal architectural building. And lastly, This mask will be used to produce the "damaged" image, I_{damaged}. The mask M will be utilized as an index, replace every pixel in the original color image I which is in the region of the mask with a block white color $[255, 255, 255]$. This is to mimic a total and absolute occlusion of the monument.

$$I_{\text{damaged}}(p) = \begin{cases} [255, 255, 255] & \text{if } M(p) = 255 \\ I(p) & \text{if } M(p) = 0 \end{cases} \tag{3}$$

4.3 Automated Pipeline for Dataset Generation

To generalise this process to our entire data of monument images, an automated batch processing pipeline was used. This script manipulates the entire pipeline: it loads raw images, executes the LCRO method on each of them, and saves outputs neatly. The pipeline produces:

1. The damaged image (I_{damaged}) with the white occlusion.
2. The binary mask (M) employed to generate the occlusion.

In addition, the pipeline constructs a necessary manifest file (`image_pairs.txt`), monitoring the file paths for every triplet (original, degraded, mask). Our data loader directly consumes this manifest file during GAN training, which guarantees a clean and reproducible pipeline from data preparation to training the model.

Fig. 2. LCRO masking.

5 Proposed Methodology

This section provides an overview of the LCRO-GAN framework, including the architecture of each component, the associated loss functions, the training methodology, and the evaluation metrics.

5.1 The LCRO-GAN Framework

The proposed methodology, LCRO-GAN, introduces a multi-stage generative adversarial network architecture designed to achieve high-fidelity image reconstruction of damaged historical structures. The overall architecture of the framework is illustrated in Fig. 3. The core of the architecture is a sequential, three-stage pipeline designed to address the inherent limitations of image inpainting. Image reconstruction, particularly for objects with intricate geometric structures and complex textures such as historical structures, involves addressing multiple interrelated sub-problems such as creating global structural consistency, reconstructing realistic and consistent textures, and restoring fine-grained, high-frequency features. To effectively address these problems, the framework decomposes the reconstruction process into multiple stages, where each stage addresses distinct challenges associated with restoration.

The pipeline begins with the first generator, denoted as G_1, which utilizes an efficient U-Net architecture based on Deep Convolutional GAN design. This stage generates a coarse but structurally coherent completion of the missing regions. The output from this stage is subsequently passed to a second generator in the sequence, G_2, which uses a CycleGAN-based architecture with an efficient ResNet architecture. This stage focuses on refining the coarse reconstruction by generating realistic textures and ensuring domain consistency. The final stage uses a third generator, G_3, based on the Pix2Pix conditional GAN architecture with an efficient ResNet design to enhance the refined image with sharper edges and high-frequency details, producing the final, photorealistic output. This progressive refinement procedure, from coarse global shapes to fine local details, can

be interpreted as a form of implicit curriculum learning. Rather than attempting to tackle the extremely challenging and ill-posed high-resolution inpainting problem in a single end-to-end operation, a process fraught with training instability and mode collapse, the framework instead steps through the learning process in multiple stages.

5.2 Stage 1: Coarse Structural Inpainting (G_1)

The first stage of the LCRO-GAN architecture is dedicated to create the foundational structure of the inpainted region. This is achieved by the generator G_1, which receives the masked input image I_{damaged} and produces a coarse reconstruction I_{coarse}. The primary objective of this stage is to provide a structurally realistic foundation for subsequent refinement steps. To achieve this, the design of G_1 is optimized for learning global, low-frequency information efficiently.

Generator Architecture (G_1). The generator G_1 is based on an encoder-decoder architecture inspired by Deep Convolutional GANs (DCGANs). The encoder decimates the input image with strided convolutions to obtain a low-dimensional latent vector without pooling. Leaky ReLU activation functions are applied to promote stable gradients and avoid the "dying ReLU" risk. The decoder then upsamples the latent vector using transposed convolutions to generate the inpainted image as output. It uses ReLU activations, with a final Tanh layer scaling pixel values to the range $[-1, 1]$. To improve training stability, Batch Normalization is applied following each convolution and transposed convolution layer (excluding the generator output and discriminator input). This normalization minimizes internal covariate shift and prevents issues such as vanishing or exploding gradients.

Loss Function (L_{G_1}). The generator G_1 is trained using a composite loss function with competing adversarial and reconstruction terms:

$$L_{G_1} = L_{\text{adv}}^{(1)} + \lambda_1 L_{L1}^{(1)} \tag{4}$$

where:

- L_{G_1}: Total loss of the Stage-1 generator (G_1).
- $L_{\text{adv}}^{(1)}$: Adversarial loss corresponding to the discriminator supervision for G_1.
- $L_{L1}^{(1)}$: L_1 reconstruction loss measuring the mean absolute error between generated and ground-truth images.
- λ_1: Weighting coefficient that controls the contribution of the L_1 loss term.

The loss function used to train the first-stage generator is defined in Eq. (4), which combines adversarial supervision with pixel-wise reconstruction loss. In Eq. (4), λ_1 balances the weight of the L1 reconstruction loss. The adversarial loss ($L_{\text{adv}}^{(1)}$) encourages G_1 to generate images that are indistinguishable from real ones, creating realism. The L1 loss ($L_{L1}^{(1)}$) computes the mean absolute pixel difference between the generated and corresponding ground-truth images. L1

loss is preferred over L2 loss because it produces sharper, less blurry outputs. At this point, the aim is to ensure structural accuracy rather than complete photorealism. Therefore, intricate losses such as perceptual loss are not included to ensure that the optimization process focuses on coarse geometric reconstruction.

5.3 Stage 2: Texture and Consistency Refinement (G_2)

The second stage of the framework focuses on texture synthesis and structural improvement. The generator G_2 takes the coarse structural map I_{coarse} from the previous stage as input and generates a refined image, I_{refined}, by generating contextually consistent textures within the reconstructed regions.

Generator Architecture (G_2). The generator G_2 adopts a CycleGAN-style architecture, which is optimized for image-to-image translation tasks. The core building block of this network is Residual Blocks (ResBlocks), each consisting of convolutional layers with skip connections that add the input of the block to its output. This residual learning framework preserves the low-level features learned from the encoder, supports deep network architectures, and facilitates efficient gradient propagation. Since Instance Normalization operates independently across each sample in the batch, G_2 uses it instead of Batch Normalization. This allows the model to preserve image-specific textures and styles. U-Net skip connections link the encoder and decoder layers to preserve global structural information and fine details from I_{coarse}, while reflection padding is used to minimize edge artifacts.

Loss Function (L_{G_2}). Optimization of G_2 is driven by an aggregate loss:

$$L_{G_2} = L_{\text{adv}}^{(2)} + \lambda_2 L_{L1}^{(2)} + \lambda_3 L_{\text{cycle}} \tag{5}$$

where:

- L_{G_2}: Total loss of the Stage-2 generator (G_2).
- $L_{\text{adv}}^{(2)}$: Adversarial loss for the second-stage generator.
- $L_{L1}^{(2)}$: Pixel-wise L_1 reconstruction loss.
- L_{cycle}: Cycle-consistency loss enforcing structural consistency between forward and inverse mappings.
- λ_2, λ_3: Weighting parameters controlling the influence of reconstruction and cycle-consistency losses, respectively.

The optimization objective for the second-stage generator is defined in Eq. (5), which incorporates adversarial, reconstruction, and cycle-consistency constraints. Here, λ_2 and λ_3 regulate the term balance. The adversarial and L1 reconstruction losses enforce realism and pixel-wise accuracy, similar to Stage 1. In addition, cycle-consistency loss (L_{cycle}) is introduced to ensure that an image translated from one domain to another and back remains consistent with the original image. This regularizes G_2, enforcing structural consistency and preventing the generation of unrealistic or inconsistent textures.

5.4 Stage 3: High-Fidelity Detail Enhancement (G_3)

The final stage of the LCRO-GAN pipeline includes the final refinement of the reconstructed image, aimed at the recovery of high-frequency details, sharp structural boundaries, and overall photorealism. In this stage, a more sophisticated generator, G_3, is utilized, based on the Pix2Pix conditional GAN architecture which takes as input the refined image I_{refined} and generates the final output I_{final}.

Generator Architecture (G_3). The generator G_3 uses a coarse-to-fine multi-scale framework to generate high-resolution images. The global generator initially produces a low-resolution output to create semantic structure, while local enhancer networks refine high-resolution feature. The outputs from all scales are combined to generate the final reconstruction. This design captures both global coherence and fine textures effectively. Hierarchical residual blocks further refine feature representations at each level, enabling stable training of deep, high-capacity models.

Loss Function (L_{G_3}). The overall objective for G_3 emphasizes perceptual quality and is defined as:

$$L_{G_3} = L_{\text{adv}}^{(3)} + \lambda_4 L_{L1}^{(3)} + \lambda_5 L_{\text{perceptual}} + \lambda_6 L_{\text{feature_matching}} \tag{6}$$

where:

- L_{G_3}: Total loss of the Stage-3 generator (G_3).
- $L_{\text{adv}}^{(3)}$: Adversarial loss for the final-stage generator.
- $L_{L1}^{(3)}$: Pixel-wise L_1 reconstruction loss.
- $L_{\text{perceptual}}$: Perceptual loss computed using feature representations extracted from a pretrained VGG-19 network.
- L_{FM}: Feature matching loss based on intermediate discriminator feature maps.
- $\lambda_4, \lambda_5, \lambda_6$: Weighting coefficients for reconstruction, perceptual, and feature matching losses, respectively.

Here in Eq. (6), λ_4, λ_5 and λ_6 balance the contributions of each loss component. The adversarial and $L1$ losses enforce realism and pixel fidelity, respectively, while the perceptual loss computed using a pre-trained VGG-19 network enforces semantic and textural accuracy. The feature matching loss aligns intermediate discriminator features for real and generated images, thereby stabilizing adversarial training and improving the visual realism of the generated images. This set of losses steers G_3 to produce images that are structurally accurate and, more importantly, indistinguishable from real ones.

5.5 Adversarial Supervision and Training Protocol

The effective coordination of the three separate generator modules is achieved through a unified discriminator and a multi-stage training protocol. This design

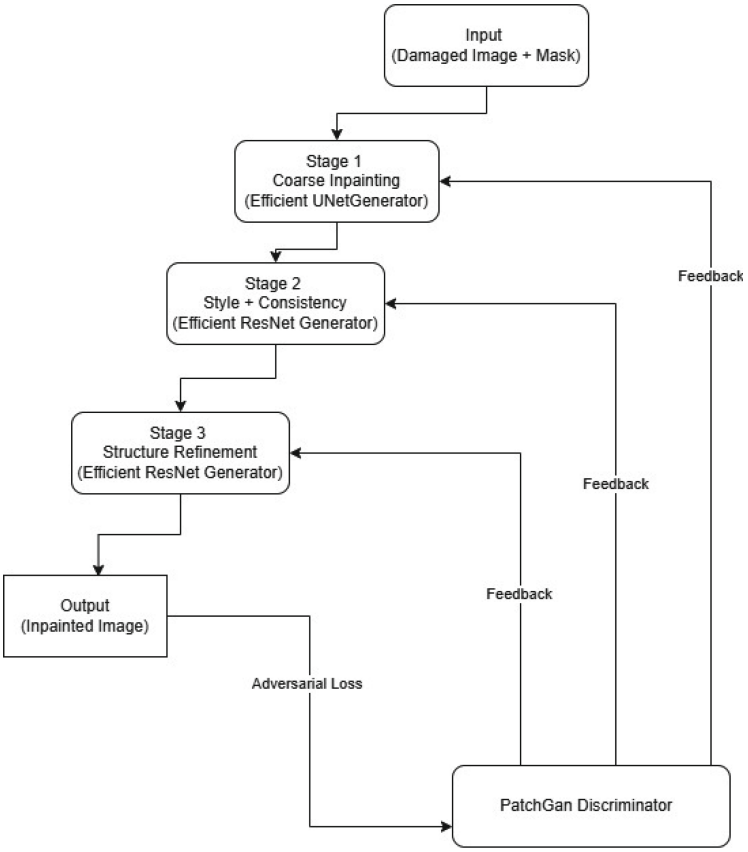


Fig. 3. High Level View of Proposed Methodology.

enables each stage to build upon the outputs of the preceding stage, resulting in a coherent, high-performance inpainting pipeline.

Unified Discriminator Design (D). A single discriminator network D provides adversarial supervision for all three generators. This discriminator uses the architecture of PatchGAN, which has been proven to be effective in various tasks such as image-to-image translation or image synthesis. Unlike a classic discriminator, which would return a single scalar value that classifies an entire image as real or fake, a PatchGAN is a fully convolutional network returning a 2D map of probabilities. Each value corresponds to the "realism" of an overlapping N $\times$ N patch of the input image (e.g., 70 $\times$ 70 patches). This patch-level assessment encourages the generator to produce realistic high-frequency details throughout the image. Moreover, this is more efficient than a full-image discriminator because it requires fewer parameters. A key characteristic of the discriminator is its ability to accept inputs at multiple scales, enabling it to effectively supervise

the outputs of all three generators, which may exhibit varying levels of detail or resolution.

Comprehensive Training Strategy. The model is optimized using a staged and joint optimization approach. Considering that the Adam optimizer assumes a learning rate that decreases in time, the training proceeds over four stages: (1) G_1 and the discriminator are trained for 100 epochs to produce coarse structural completions; (2) with G_1 frozen, G_2 is trained for 100 epochs to add realistic textures; (3) freezing both G_1 and G_2, G_3 is trained for 100 epochs to enhance fine details and high-resolution quality; and (4) Joint Fine-Tuning: all generators are unfrozen for an end-to-end training with a reduced learning rate. This hybrid training combines the stability of progressive learning and the comprehensive improvement of joint optimization. It enables earlier generators to modify their outputs to facilitate more seamless stage transitions, ultimately creating a final coherent and high-fidelity reconstruction.

5.6 Evaluation Metrics

Quantitative performance evaluation is conducted by aggregating a set of metrics capable of capturing pixel-level accuracy, perceptual realism, and distributional similarity in particular. PSNR measures pixel-wise reconstruction fidelity but often does not correlate well with human perception; SSIM tackles this aspect by comparing luminance, contrast, and structure between the generated and real images for a more perceptually correct measure. To further assess the perceptual aspect, LPIPS computes feature-space distances between deep network-processed generated and real images, leveraging VGG or AlexNet, demonstrating

(a) Masked Input (b) Restored Output

Fig. 4. Qualitative comparison of masked input images and corresponding reconstructions generated by the proposed LCRO-GAN framework.

strong correlation with human judgment. Finally, FID quantifies both the quality and diversity of generated images by comparing feature distributions of real and generated images using an InceptionV3 network, where lower scores indicate higher realism. Collectively, these metrics provide a comprehensive and rigorous evaluation of the model's reconstruction fidelity, perceptual quality, and generative performance.

6 Results and Discussion

The model was tested and trained on a high-performance computing environment. All tests were run in an NVIDIA Quadro RTX 5000 with 16 GB of its own VRAM as well as CUDA version 12.6. Every generator stage of the LCRO-GAN pipeline was trained with 100 epochs, G1 took about 15 minutes per epoch, G2 and G3 took about 30 minutes per epoch, as they are more complex to architecture. After the three-stage training, further 100 epochs of joint end to end fine-tuning were done again at a rate of 30 minutes per epoch to allow the training to become stable and the high-frequency photorealistic details and low-frequency structural priors to be co-learned on the entire model. The reconstruction results of the pipeline is illustrated in Fig. 4. This demonstrates the progression from coarse structural completion to high-fidelity refinement.

Table 1. Reconstruction Metrics Across LCRO-GAN Stages.

Stage	PSNR (dB)	SSIM	LPIPS	FID
Stage 1 – Coarse	24.8	0.712	0.073	48.2
Stage 2 – Refined	26.5	0.764	0.058	31.7
Stage 3 – Details	27.9	0.803	0.045	18.4
Final (Joint Fine-Tune)	**28.7**	**0.827**	**0.038**	**11.5**

Table 1 illustrates the reconstruction quality of the proposed multi-stage GAN model. The framework had been shown to be pixel-level accurate, with the final Joint Fine-Tune achieving a PSNR of 28.7 dB and an SSIM of 0.827. More importantly, the gradual training approach leads to extensive perceptual realism improvements. FID, which is a measure of the quality and distribution of images, dropped significantly by 76.1 percent since the starting coarse stage score of 48.2 scored lower in the last stage of 11.5. On that note, the LPIPS improved immensely and reduced to its current final score of 0.038, which is outstanding. These findings as a whole therefore suggest that the progressive optimization achieved the task of moving the framework out of producing a plausible structural completion into producing a high-fidelity, photorealistic reconstruction, thereby justifying the design of the framework.

7 Conclusion

The proposed LCRO-GAN model exhibits strong reconstruction ability and structural consistency of the dataset, which contains the images of various Dravidian-style temple architectures and sculptures. The model was able to reconstruct structurally coherent and realistic results from degraded inputs. Furthermore, the combination of a multi-stage GAN framework with the novel LCRO masking technique enables the model's ability to identify the overall architectural layout as well as the minute fine-grained details of temples and sculptures, thus demonstrating their effectiveness in the digital preservation and restoration of historical monuments.

Future work will focus on expanding the dataset to include wide variety of architectural styles across different regions and cultures, improving the model's generalization ability. Additionally, exploring context-aware image reconstruction techniques would eliminate the need of explicit damage masks. We aim to develop an end-to-end application with automated preprocessing, inference, visualization, and export tools, making the model usable by researchers, historians, and conservation experts for heritage restoration and digital preservation. These future advancements together position LCRO-GAN as a scalable foundation for next-generation digital heritage restoration systems.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Deng, X., Yu, Y.: Ancient mural inpainting via structure information guided two-branch model. Heritage Sci. **11**(1), 1–16 (2023). https://doi.org/10.1186/s40494-023-00972-x
2. Basu, A., et al.: Digital restoration of cultural heritage with data-driven computing: a survey. IEEE Access (2023). https://doi.org/10.1109/ACCESS.2023.3280639
3. Altaweel, M., Khelifi, A., Zafar, M.H.: Using generative AI for reconstructing cultural artifacts: examples using Roman coins. J. Comput. Appl. Archaeol. **7**(1), 301–315 (2024). https://doi.org/10.5334/jcaa.146
4. Gaber, J.A., Youssef, S.M., Fathalla, K.M.: The role of artificial intelligence and machine learning in preserving cultural heritage and art works via virtual restoration, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. X-1/W1-2023, pp. 185–192 (2023). https://doi.org/10.5194/isprs-annals-X-1-W1-2023-185-2023
5. Nogales, A., et al.: ARQGAN: an evaluation of generative adversarial network approaches for automatic virtual inpainting restoration of greek temples. Expert Syst. Appl. **180**, 115092 (2021). https://doi.org/10.1016/j.eswa.2021.115092
6. Guan, J., et al.: Progressive generative mural image restoration based on adversarial structure learning, NPJ Heritage Sci. **13**, 309 (2025). https://doi.org/10.1038/s40494-025-01883-9
7. Arumugam, C.D., Banothu, B.: Integrating multi-scale attention network with deep image prior for single image restoration. Int. J. Inf. Technol. **17**, 555–565 (2024)

8. Malagi, V.P., Naresh, E., Mithra, C., Suresh, B.V.N.V.K.: Adaptive residual convolutional neural network for moiré image restoration. Int. J. Inf. Technol. **17**(2), 755–765 (2024/2025)
9. RSINet: Inpainting remotely sensed images using triple GAN framework. In: Proceedings of the IEEE IGARSS (2022). https://doi.org/10.1109/igarss46834.2022.9884330
10. Mol, V.R., Maheswari, P.U.: The digital reconstruction of degraded ancient temple murals using dynamic mask generation and an extended exemplar-based region-filling algorithm. Heritage Sci. **9**(1), 1–18 (2021). https://doi.org/10.1186/S40494-021-00604-2
11. Chavan, S.A., Chaudhari, N.M., Pawar, U.B.: Mathematical analysis behind occlusion handling in image with deep learning. Int. J. Inf. Technol. **15**(7), 3727–3739 (2023)
12. Wajid, M.A.: A deep learning approach for image and text classification using neutrosophic fuzzy sets. Int. J. Inf. Technol. (2024)
13. Rakhimol, V., Maheswari, P.U.: Restoration of ancient temple murals using CGAN and PCONV networks. Comput. Graph. **109**, 83–92 (2022). https://doi.org/10.1016/j.cag.2022.11.001
14. Image restoration technology of Tang Dynasty tomb murals based on adversarial edge learning. Research Square (2023). https://doi.org/10.21203/rs.3.rs-2765366/v1
15. Devalla, V., Arya, A.: Temples restoration using gated convolution and contextual attention in generative adversarial networks. In: Proceedings of the IEEE International Conference on Smart Computing and Communication (ICSCCC) (2023). https://doi.org/10.1109/ICSCCC58608.2023.10176941
16. Singh, U., Maiti, S., Saini, A.K.: Ancient Indian murals digital restoration through image inpainting. IEEE SPIN (2023). https://doi.org/10.1109/SPIN57001.2023.10116111
17. Pexels, Free stock photos (2024). https://www.pexels.com
18. Pixabay, Royalty-free images (2025). https://pixabay.com

Run-Time Assurance for Onboard COTS-Based AI in Safety-Critical LEO Small-Satellite Systems

Omer Khalid[1,2], Jose-Carlos Gamazo-Real[1,2(✉)], and Elena Roibás-Millán[1]

[1] Universidad Politécnica de Madrid, Instituto Universitario "Ignacio Da Riva" (IDR-UPM), 28040 Madrid, Spain
{omer.hkhalid,josecarlos.gamazo,elena.roibas}@upm.es

[2] Universidad Politécnica de Madrid, ETSI de Sistemas Informáticos, 28031 Madrid, Spain

Abstract. Safety-critical systems are increasingly applying Artificial Intelligence (AI) to enhance perception, autonomy, and decision-making. In parallel, the widespread adoption of Commercial Off-The-Shelf (COTS) hardware offers substantial benefits in terms of cost, computational performance, and development speed. However, combining unpredictable, nondeterministic AI models with heterogeneous and uncertified COTS devices introduces significant challenges for guaranteeing reliability, predictability, and certifiability, which are essential requirements in systems where failures can be catastrophic. This article examines these emerging challenges and proposes a layered Run-Time Assurance (RTA) methodology tailored for AI-enabled functions deployed on COTS platforms for safety-critical systems. The approach introduces four complementary monitoring layers, such as input, inference, output, and system-level behavioural supervision, which are designed to detect anomalies, timing overruns, semantic inconsistencies, and elevated mission risks before they escalate into unsafe states. A detailed case study of a Low Earth Orbit (LEO) microsatellite, using the UPMSat-2 from the IDR-UPM research institute as the reference platform, is presented to demonstrate how the proposed RTA framework can maintain safe operation during perception uncertainties, hardware-induced degradations, and collision-risk assessments. The proposed methodology aims to support future certifiable integration of onboard AI pipelines, enabling reliable autonomy in resource-constrained spacecraft and advancing the safe adoption of high-performance COTS computing within New-Space systems.

Keywords: Run-Time Assurance · Artificial Intelligence · Commercial Off-The-Shelf · Safety Monitor · Safety-Critical System · LEO Small Satellite · New Space

1 Introduction

Safety-critical systems are entering a new phase of evolution, driven by the growing demand for higher levels of autonomy onboard in domains such as aerospace, transportation, and industrial robotics [1]. As these systems increasingly operate with reduced

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 445–460, 2026.
https://doi.org/10.1007/978-3-032-30491-9_28

human supervision, they must be able to perceive complex environments, adapt to dynamic conditions, and make reliable decisions in real time, capabilities for which Artificial Intelligence (AI) has emerged as a key enabler [2]. Moreover, the adoption of Commercial Off-The-Shelf (COTS) hardware is gaining popularity across multiple safety-critical domains, because they offer clear advantages such as low cost, high computational performance, shorter development cycles, and faster technological evolution compared to certified or radiation-hardened systems [3, 4]. These benefits are particularly appealing in the New Space sector, where small satellites require faster and less costly development, and testing methodologies apart from coverage, parameters such as availability, mean time between failures, and reliability [5].

Nonetheless, integrating AI-based functionalities in COTS hardware into safety-critical systems also brings extra challenges. AI models executed on COTS processors behave unpredictably and its nondeterministic components whose internal decision mechanisms cannot be fully traced or formally verified [2, 6]. Their performance and outputs could differ based on input, hardware load, thermal conditions, or minor implementation details that make them inherently difficult to certify within traditional assurance frameworks [7]. At the same time, safety-critical systems are built upon principles of deterministic, explainable, and verifiable behaviour, where every operational state must remain within predefined safety margins, and timing guarantees must be respected under all conditions [2, 8]. This creates a mismatch considering that AI-enabled COTS devices offer exceptional computational capability but lack the predictability and certifiability required for systems whose failure modes could bring serious losses. Bridging this gap requires new architectural strategies capable of monitoring the behaviour of AI-driven COTS components while maintaining the strict safety guarantees that these domains require.

To bridge this gap between advanced AI capabilities in COTS devices and strict safety requirements, researchers have drifted towards Run-Time Assurance (RTA) methods. RTA provides continuous online supervision of untrusted components which aren't present in static verification methods. They can supervise AI models, while guaranteeing that safety boundaries are not violated during operation [9, 10]. In its classical form, RTA pairs an unverified high-performance controller with a verified safety controller and employs a real-time monitor to decide when intervention is required. This paradigm is emerging as a practical and scalable method for integration of AI in safety-critical systems, particularly when full formal verification is infeasible or when systems must operate on heterogeneous and potentially unreliable COTS hardware. Preliminary works have explored modular and multilayer RTA structures, including frameworks with multiple controllers and monitoring layers [11], as well as safety-filtering architectures that separate controller, monitor and override layers [12]. However, none introduces a unified four-layers model tailored to AI perception on heterogeneous COTS hardware, making the architecture proposed here distinct for safety-critical space systems. The novelty of the present work lies in proposing a unified four-layer RTA model tailored to onboard AI perception on heterogeneous COTS platforms, integrating different behavioural monitoring within a single safety-assurance architecture.

This article proposes an approach for safely integrating AI-based functionalities into safety-critical systems operating on COTS hardware. It proposes a layered RTA

architecture and safety monitors that supervise AI components through input, inference, output, and system-level behavioural monitors. The motivation of this study is the need to exploit AI and COTS hardware in resource-constrained New Space missions without compromising safety. The approach ensures that mission safety is preserved even when AI models exhibit uncertainty, nondeterminism, or timing violations associated with heterogeneous COTS platforms. A detailed Low Earth Orbit (LEO) microsatellite case study, using the UPMSat-2 from the IDR-UPM (Instituto Universitario "Ignacio Da Riva") as the reference platform [13], demonstrates how the proposed approach maintains safe operation during perception anomalies and hardware-induced degradations. The work contributes to a practical path towards certifiable AI deployment in resource-restricted and critical domains such as aerospace. The proposed framework is analytical and architectural in scope; no dedicated analysis software was used at this stage. Instead, it was developed through traditional engineering analysis.

The remainder of this article is structured as follows. Sect. 2 introduces the motivation for contextualising the increasing adoption of AI in safety-critical systems, beside the outline of the challenges faced when using resource-constrained COTS devices rather than certified ones. Then, in Sect. 3, a description of RTA systems and their operations is presented. In Sect. 4, the RTA layered methodology is presented and described with a detailed case study involving a scientific LEO microsatellite equipped with an AI-based perception pipeline demonstrating how this approach can be instantiated in a real mission scenario. Finally, Sect. 5 presents the conclusions and outlines future directions of research.

2 Motivation Scenario: Integration of COTS Devices in Safety-Critical Applications

Safety-critical systems like as avionics controllers, spacecrafts, self-driving vehicles, industrial systems, and medical equipment increasingly rely on complex software as they integrate advanced sensing, perception, and decision-making capabilities while running on tightly constrained ad-hoc hardware. These must meet strict real-time and reliability requirements while operating with limited Central Processing Unit (CPU) capacity, memory, and energy. However, with the rapidly increasing in processing capabilities, the possibility of upgrading the hardware to meet these demands is infeasible due to weight constraints, higher certification burdens, increased energy consumption, and cost considerations. This creates a pressing need for techniques to optimise resource usage without compromising safety guarantees.

2.1 Safety Considerations in Hardware and Software Architectures

The emergence of edge-computing in safety-critical systems introduces new demands on how hardware and software architectures ensure safety. Unlike traditional centralised architectures, edge systems must perform computations under strict latency, energy, and reliability constraints. Hardware platforms deployed at the edge often combine different processing units, such as multicore CPUs, Field Programmable Gate Arrays (FPGA),

Graphics Processing Units (GPU), and Deep Learning (DL) accelerators, each with different timing behaviour and resource-sharing patterns [14]. This heterogeneity enables high performance; however, it complicates predictability and affects robustness and reliability in safety-critical contexts. As a result, determining how current hardware architectures provide deterministic behaviour and ensure overall reliability becomes a central question in safety-critical systems. On the software side, modern software systems rely on complex middleware, virtualisation, containerisation, and distributed computing [15]. These architectures must coordinate tasks with different levels of criticality while sharing limited hardware resources in real time. It introduces additional layers of nondeterminism and raises questions about whether they can maintain the safety guarantees required in safety-critical systems. Moreover, the growing adoption of AI adds further difficulties, as it black-box behaviour and high computational demands, can produce unexpected outputs, and complicate the certification process, which sacrifices and affects the system reliability [16].

2.2 COTS Devices Integration for Safety-Critical Edge Computing

The increasing interest in using of COTS hardware in safety-critical edge systems in the edge is influenced by its obvious advantages: lower cost, higher performance, shorter development timelines, and rapid progression of computing technologies [3]. These benefits are especially attractive in domains such as aerospace, where traditional hardware is expensive, slow to certify, and often lags cutting-edge commercial processors with respect to computational capabilities [17]. However, COTS components adoption into this sector introduces significant challenges for systems that must meet strict safety, reliability, and robustness requirements [18]. COTS hardware is typically not designed with determinism or long-term reliability in mind. its limitations can become even more exposed when deployed at the edge in harsh environments with strong thermal, radiation, or vibration constraints. For aerospace systems in particular, disadvantages such as radiation susceptibility and uncertainty in long-term component behaviour raise concerns about how safety can be assured when using COTS hardware [19]. Figure 1 summarizes the key concepts of safety-critical edge AI-based systems, as it includes domains such as autonomous vehicles, aerospace, and space technologies, all of which require high-safety considerations to ensure reliable operation. AI introduces additional challenges due to its behaviour, especially when deployed on resource-constrained edge devices.

3 Run-Time Assurance and Safety Monitoring Methodology

3.1 RTA Methodology

Safety-critical systems operate under strict safety constraints, where failures may have severe consequences including death, environmental damage, or significant economic losses; therefore, they demand high guarantees of correct and reliable behaviour. Meeting these requirements has become increasingly challenging with the integration of advanced AI components, who's opaque (black-box) nature and limited interpretability make it difficult to fully verify their operation under all operational conditions. This challenge is

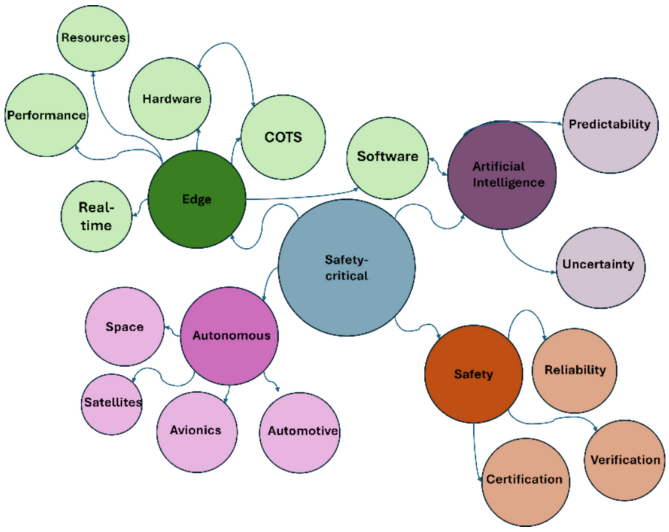


Fig. 1. Relationships among key concepts in safety-critical systems and AI-COTS edge computing.

further magnified by the widespread adoption of COTS hardware and software, which, although cost-effective and high-performance, are not certified for the timing, reliability, or functional constraints demanded by safety-critical systems. Their use may cause performance variability and unnoticed interactions that may alter or degrade the behaviour of AI algorithms deployed in such environments.

As shown in Fig. 2, the architecture of an RTA framework typically comprise three main components: an untrusted or experimental controller that carries out the primary mission objectives; a safety controller that is simpler, verified, and capable of keeping the system within a guaranteed safe region; and a safety monitor that decides when to shift authority between the two. This monitor evaluates real-time measurements, derived state estimates, and predicted trajectories to determine whether the system is approaching unsafe regions of its operational space. If a violation of safety constraints is imminent, the monitor ensures a deterministic low-latency switch to the safety controller, which maintains the autonomous system within predefined boundaries until normal operation can resume safely.

RTA provides a promising approach to mitigating these risks, particularly those arising from performance variability and hidden interactions associated with the use of COTS technology, while increasing the level of assurance in such systems. It is traditionally used when an untrusted or experimental controller is deployed within an autonomous platform, enabling a seamless transition to a conventional or certified safety controller when necessary. In this context, the RTA mechanism continuously monitors the operational state of the system, ensuring that AI-driven actions and their interactions with COTS components remain within a predefined safety margin. When potential safety violations are detected, the control is immediately transferred to a verified backup controller or a safe operational mode, thus preserving overall system safety despite the presence

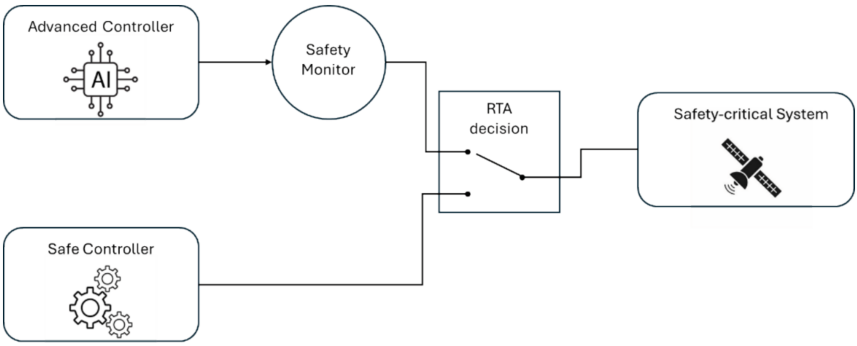


Fig. 2. Conventional architecture of an RTA framework.

of opaque AI components and uncertified COTS elements. The main objective in RTA is to design a switching policy that determines, at each decision point, whether control should reside with the verified safe controller or the untrusted one. This strategy must guarantee the continued safety of the system while simultaneously striving to optimise the relevant performance criterion.

3.2 RTA Variations

Several variations can be considered in the RTA methodology, but they all follow the core idea of combining an untrusted controller with a verified safe controller under the supervision of a run-time monitor. In the academic literature, RTA architectures are typically classified into the following main types:

- Simple-switching RTA (Simplex): A monitor switches from an untrusted controller to a certified safe controller whenever the system approaches unsafe conditions, ensuring immediate protection with minimal architectural complexity [20].
- Shielding-Based RTA: A safety shield filters or corrects unsafe control actions instead of switching controllers, allowing continuous operation while preventing unsafe behaviour in real time [12, 21].
- Predictive or Look-ahead RTA: The monitor predicts future trajectories under proposed actions and intervenes early if a safety violation is likely, enabling anticipatory rather than reactive assurance [12].
- Recovery-Based RTA: The verified safe controller gradually guides the system back to safe conditions, supporting smoother transitions and stable operation, as opposed to sudden switching [22].

3.3 Deployment of RTA and Safety Monitors for Safety Critical System

RTA has great potential as a practical approach to enable the deployment of advanced software-intensive functionalities, particularly those that rely on AI in safety-critical systems. Unlike offline verification techniques, which attempt to exhaustively prove correctness before deployment, RTA operates continuously during execution. It supervises system behaviour in real time, detecting deviations from safe operating boundaries and

intervening before unsafe states can propagate. This characteristic makes RTA particularly attractive for safety-critical systems, and especially for small satellites and other New-Space platforms. In these environments, budget constraints, compressed development schedules, and the increasing adoption of commercial processors introduce a non-negligible likelihood of unexpected behaviours once the system is in orbit.

Designing an RTA begins with representing the system in terms that allow the safety monitor to predict how future behaviour may evolve under different controllers. The operating conditions of the system are divided into a collection of discrete quantised states, and each controller defines how these states change over time. An RTA design must then specify how to choose between the two controllers at every moment, so the system remains within its safe operating region while still attempting to achieve mission objectives. To do this, the designer establishes a set of unsafe states that the system must never enter and defines a monitoring logic that can detect when the untrusted controller is likely to drive the system toward those unsafe regions. The switching logic then replaces the untrusted controller with the safe controller whenever necessary. Because different choices of controller affect mission performance, such as efficiency, data quality, or energy consumption, the design may also incorporate a way to evaluate how desirable a given trajectory is and adjust how aggressively the untrusted controller is used. Ultimately, the goal is to construct a switching strategy that keeps the system safe while allowing the highest mission value possible.

The deployment of RTA in safety-critical systems introduces additional considerations. Due to the limited resources of the autonomous onboard computers, they should be capable of handling the computational overhead associated with continuous monitoring, and the monitor itself must be immune to the same faults it has been designed to detect. Integrating RTA with COTS hardware also requires ensuring that timing determinism can be maintained when running on heterogeneous processing architectures such as GPUs or multicore systems. These challenges highlight the importance of lightweight and certifiable RTA methodologies that can function in a reliable manner within the constraints of small-satellite platforms. Table 1 shows examples of RTA methods developed in different safety-critical domains.

Table 1. RTA methods developed in different safety-critical domains.

Name	Description	Use case/Domain implementation	Limitation with respect to the presented work	Ref.
Simplex RTA	Trusted safety controller supervises untrusted controller; monitor switches control when safety is threatened.	Automatic vehicle braking system	Does not explicitly monitor AI perception nor COTS effects	[20]

(*continued*)

Table 1. (*continued*)

Name	Description	Use case/Domain implementation	Limitation with respect to the presented work	Ref.
Safety filtering RTA	Online monitor filters or corrects unsafe commands in real time, ensuring safety with untrusted controllers.	Practical method for safety-critical Cyber-Physical Systems (CPS)	Primarily focused on control and command	[12]
Automatic-GCAS (Ground Collision Avoidance System)	Real deployed RTA system that monitors aircraft state and automatically overrides pilot control to avoid ground collision.	Aviation systems	No AI, no perception uncertainty, certified hardware	[23]
ULGEN Framework	RTA modules with multiple safe and advanced controllers plus a decision module	Robotic CPS platforms, such as case studies with real robot implementation	Does not explicitly monitor AI perception nor COTS effects	[11]

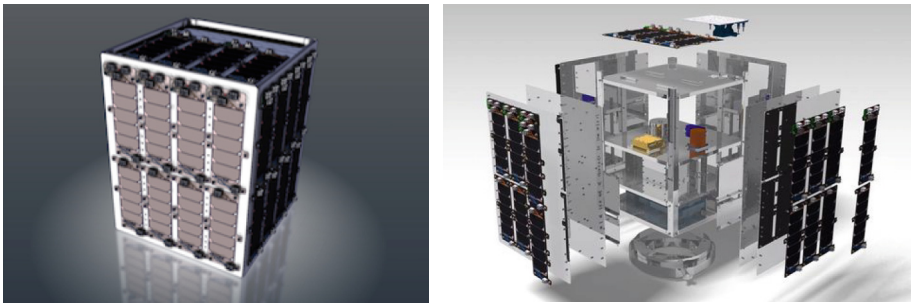
4 Case Study: RTA-Enabled Safety Monitoring in Scientific LEO Small Satellites

4.1 System Overview of LEO Small Satellites

LEO satellites typically operate between 160 and 1,500 Km above the earth surface [24], making it one of the most rapidly evolving sectors in modern space technology. Their proximity to earth allows low-latency communication [25], along with their lower cost, smaller mass, and higher scalability compared to both Medium Earth Orbit (MEO) and geostationary Orbit (GEO) satellites [26], making them highly suitable for modern missions. These features are well-suited for applications requiring hard real-time connectivity, for instance, broadband internet, IoT, mobility services and earth observation [27]. In recent years, LEO small-satellite constellations have gained more interest as there are some projects of large-scale satellite constellation including Airbus, OneWeb, and SpaceX Starlink. These initiatives demonstrate not only the feasibility of deploying mass-produced satellites, but also the potential of integrating AI capabilities on board at scale. However, LEO platforms also introduce a set of technical challenges that AI systems on board must help address, such as shorter operational lifetimes, constrained energy resources, and strict real-time pro-cessing requirements. Consequently, AI models designed for LEO small satellites must be compact, energy-efficient, and operationally robust, while simultaneously ensuring safety objectives of the mission.

4.2 UPMSat-2 Microsatellite as a Reference Platform

UPMSat-2 is a microsatellite developed by the IDR-UPM research institute of the Universidad Politécnica de Madrid [13]. Its approximate mass is 50 kg, with dimensions of 0.5x0.5x0.6 m, and a power output of 15 W, as illustrated in Fig. 3. It was put into orbit on September 3, 2020, thanks to the In-Orbit Demonstration and Validation (IOD/IOV) program from the European Commission. UPMSat-2 operates in a polar orbit at an altitude of approximately 600 km, within the LEO regime, and has an estimated operational lifespan of two years. However, after more than five years in orbit, it successfully completed its mission after re-entering Earth's atmosphere on October 11, 2025.



(a) (b)

Fig. 3. Images of the envelope of the UPMSat-2 microsatellite (a) and the deployment of the electro-mechanical elements that constitute the spacecraft (b).

The satellite is designed as a technology demonstrator, carrying a payload of several experiments from research groups and industry, mainly focused on assessing equipment performance in space. The system architecture incorporates a magnetic attitude control system and various sensors, such as voltage, intensity, and temperature, which are used for platform monitoring, experimentation, and over-all spacecraft operations. Communications with the ground station are established through two radio links using the UHF 400 MHz band, and the satellite includes an onboard mission computer responsible for all data handling, control functions, and experiment management. The computer utilises a Gaisler LEON3 processor and incorporates peripherals such as memories and analogue/digital interfaces. Its primary functions include attitude determination and control (ADCS), platform monitoring, data storage, telemetry and telecommand (TMTC) management, and payload operation. The TMTC subsystem plays an important role in maintaining communication with the ground station through the telecommunications hardware, and the importance of the ADCS in ensuring the correct orientation of the satellite relative to the Earth. The RTA and safety monitoring approach proposed in this article is intended for integration into future iterations of UPMSat-2, which is considered the reference platform for the case study.

4.3 Proposed Application of RTA and Safety Monitors

This case study demonstrates how a layered RTA and safety monitor can be instantiated for a LEO small satellite that uses AI-based image processing on board. This functionality can be implemented on a COTS device to support conjunction assessment and autonomous collision avoidance.

The proposed system consists of a scientific LEO small satellite equipped with an AI-based image processing algorithm on board capable of tracking debris and other space objects. This algorithm estimates their relative motion to predict the probability of an intersection with the satellite's trajectory and to prevent possible collisions. To enable these functions, the satellite is equipped with (i) a wide field of view (FoV) optical camera to capture the surrounding space environment, serving as the raw input for the AI model, (ii) an ADCS to determine the spacecraft orientation that ensure images are correctly orientated, (iii) data from receivers of the Global Navigation Satellite System (GNSS), inertial sensors, and ground-based tracking to provide position, velocity, and absolute time (PVT), and (iv) a COTS System-on-a-Chip (SoC) module executing an AI perception pipeline. The processed perception output can provide the onboard mission computer (OBC) with an estimate of the probability of collision (*Pc*), enabling the satellite to remain within its safety envelope (*Pc* below a mission-defined threshold) and to execute a minimal-risk avoidance burn if the risk increases or the AI becomes untrustworthy at runtime. Fig. 4 demonstrating the high-level architecture of the proposed system, showing how sensor inputs, mission information and AI perception outputs are combined.

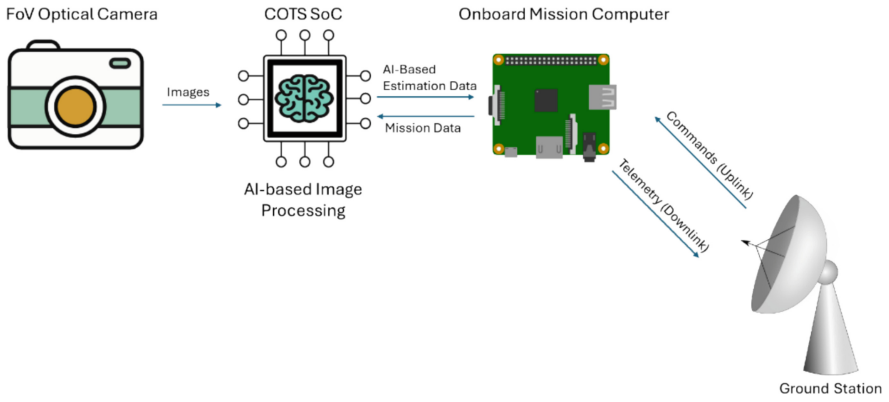


Fig. 4. Proposed onboard architecture of the AI-based perception system.

Given the safety-critical nature of satellites, their operation must remain safe in all circumstances. However, the behaviour of AI-based systems, characterised by black-box properties and inherent uncertainties, can make them difficult to fully trust. In addition, employing COTS components in harsh environments introduces additional reliability concerns. As summarised in Table 2, the failure modes associated with the use of black-box AI models and COTS devices are mapped to their corresponding system hazards. Both failure categories have the potential to jeopardise the hard real-time deadlines

associated with perception and guidance tasks. RTA and mode switching are therefore essential to maintaining safety without requiring modifications to the black-box AI models.

Table 2. Analysis of failure modes and reliability hazards of the proposed system.

Failure mode (source)	Description	Hazard/effect
Missed detection (AI)	Undetected object from image	Underestimated *Pc*, causing collision risk
False-positive detection (AI)	Unreliable collision prediction	Unreliable prediction
Latency overruns (AI/COTS)	Inference after passing the deadline	Degraded *Pc* accuracy
Input corruption (camera)	Blur, saturation and other effects	False negative, positive predictions
Compute degradation (COTS)	GPU overheating, memory pressure	Monitor dropout, loss of determinism

4.4 RTA System Proposal

Proposed Layered RTA Architecture. The proposed RTA and safety monitor for LEO small satellites, shown in Fig. 5, is currently under consideration for implementation. It shows that different monitors cooperate to detect anomalies before they propagate and lead to unsafe decisions for the spacecraft. It is designed to serve as a safety mechanism capable of maintaining satellite operation despite failures in the perception and estimation system. It is composed of four monitoring layers (input, inference, semantic, and behaviour layers) and their associated signals, which feed the safety system. The design can follow established taxonomies of perception-safety monitoring and the RTA literature [28].

Fig. 5. Schematic of the proposed layered RTA architecture.

The four monitoring layers that constitute the proposed system are described in detail below:

- Layer 1: Sensors or input monitors. This layer is used to validate the optical inputs before the AI inference, providing an input health output $h_{in} \in [0,1]$ and event flags.

- Layer 2: Inference monitors. This layer is in charge of ensuring the COTS pipeline meets compute and timing constraints, providing an output of compute health score h_{cmp} and timing OK/Overrun, thermal and memory warnings.
- Layer 3: Output monitors. This layer is responsible for enforcing physics and geometry-based checks on AI tracking, assessing motion, size, orbital behaviour, and cross-sensor consistency, producing a vector and count of contract violations.
- Layer 4: System-behaviour monitors. This layer is responsible for checking safety envelopes at mission level, providing a risk score assessment R_{sys} and recommending safe actions (e.g., "Hold" action to maintain current state of AI navigation and "ManeuverNow" action to transfer control to the safe controller).

The closed-loop system, comprising the satellite, perception module, and guidance controller, can be modelled as a finite and discrete-time plant as follows, taking into account that the finite-state and discrete-time abstractions are a common way to pose RTA switching as a computational problem over a plant [28].

$$M = (Q, q_0, A, D) \tag{1}$$

where Q is a large and finite set of quantized states, $q_0 \in Q$ is the initial state bundles orbital/relative motion and monitor summaries (input health h_{in}, compute health h_{cmp}, semantic-violation count v_{sem}, and system risk R_{sys}), A is a finite action set, and $D{:}Q \times A \to Q$ is a deterministic transition function over a fixed sampling period to specify the next state based on the current state Q and the action A taken. This representation allows the RTA switching problem to be formulated in terms of quantized system states and discrete controller actions.

With respect to the actions (A), the representation can be performed using the two-action RTA formulation:

$$A = \{U, S\} \tag{2}$$

where U selects the untrusted (AI-dependent) controller and S selects the safe classical controller, defining the two possible RTA actions: maintaining the AI-dependent controller or switching to the verified safe controller

For the switching policy π, a memoryless policy can be adopted, enabled by the layered monitors so that switching depends only on the current quantized state:

$$\pi : Q \to \{U, S\} \tag{3}$$

Operation Modes. The proposed RTA system can enable switching among three modes, such as nominal, suspicious, and safe, as described below:

- Nominal (AI-trusted): This is the nominal mode where the AI inference is generated in trustful manner, and all safety contracts are satisfied and no hard violations; during this mode it uses the AI-enhanced fully navigation. It operates when the following conditions are satisfied:

$$h_{in} \geq \theta_{in} \tag{4}$$

$$h_{cmp} \geq \theta_{cmp} \tag{5}$$

$$v_{sem} \leq \alpha \tag{6}$$

$$Pc \leq Pc_{low} \tag{7}$$

where θ_{in} is the threshold for a good image input, θ_{cmp} is the threshold for compute reliability, v_{sem} is the semantic violation count produced by Layer 3 of monitor layers, α is the allowed number of soft violations, and Pc_{low} is the threshold used to decide if the system remains safe.

- Suspicious (degraded trust): It happens with soft anomalies, such as $h_{cmp} < \theta_{cmp}$, repeated soft safety contract violations, or Pc trending upward. During this mode, AI outputs are down weighted and request ground confirmation.
- Safe (safe verified): It occurs at hard violations, such as timing greater than L_{max} (deadline miss), thermal limit exceeds, persistent semantic violations, or $Pc \leq Pc_{high}$, where Pc_{high} is the upper bound on acceptable collision probability. During this mode the AI influences are disabled and the control will be handled back to the verified controller.

5 Conclusion and Future Work

This article considers the growing reliance on AI and COTS hardware in safety-critical systems and the resulting challenges in guaranteeing reliability, determinism, and certifiability. To address these issues, a system based on layered RTA and safety monitors is proposed, incorporating multilevel safety monitors to maintain safe operation even when deploying computationally demanding and opaque AI models on heterogeneous COTS platforms. The effectiveness of multi-layer monitoring combined with a structured switching policy is shown through a LEO microsatellite case study, using the UPMSat-2 from the IDR-UPM as the reference platform, hence demonstrating safe behaviour under nominal, degraded, and unsafe conditions. The findings demonstrate that RTA offers a practical approach for integrating advanced image processing pipelines into resource-constrained space systems without requiring full formal verification.

Future work will be centred on the practical implementation of the proposed layered RTA architecture and multi-level safety monitoring within an onboard AI image-processing payload for LEO small satellites. This effort aims to prove that mission safety can be preserved even when relying on computational components that exhibit black-box behaviour and limited interpretability. Through the deployment of RTA-supervised AI modules on heterogeneous COTS platforms in orbit, future developments intend to provide practical and thorough evidence that advanced perception algorithms can operate safely under real mission conditions despite timing variability, hardware uncertainty, or inference anomalies. On a broader horizon, this research direction contributes to the development of certification pathways capable of integrating AI-driven subsystems in safety-critical space missions.

Acknowledgments. This work was partially funded by the IDR-UPM (Instituto Universitario "Ignacio Da Riva") research institute of the Universidad Politécnica de Madrid (UPM) under the project "MLPLUS-CM: Consolidación de una Plataforma de Microsatélite y Tecnologías para Investigación Científica Espacial Avanzada" (Ref.: TEC-2024/TEC-263) of the Spanish regional call for proposals 2024 titled "Grants for R&D projects carried out in collaboration between research groups belonging to universities and research organisms of the Community of Madrid" in the modality of R&D Activity Programs in Technologies. The research work was also partially funded by the "Programa propio de investigación de la Escuela Técnica Superior de Ingeniería de Sistemas Informáticos (ETSISI)" within the call "Convocatoria de ayudas para la presentación de trabajos en congresos de investigación y reuniones científico-técnicas 2026" of the ETSISI of the UPM. The authors also acknowledge the collaboration of the ETSISI through its doctoral program in Computer Science and Technologies for Smart Cities.

References

1. Perez-Cerrolaza, J. et al.: Artificial intelligence for safety-critical Systems in Industrial and Transportation Domains: a survey. ACM Comput. Surv. **56** (2024). https://doi.org/10.1145/3626314
2. Wang, Y., Chung, S.H.: Artificial intelligence in safety-critical systems: a systematic review. Ind. Manag. Data Syst. **122**, 442–470 (2021). https://doi.org/10.1108/IMDS-07-2021-0419
3. Budroweit, J., Patscheider, H.: Risk Assessment for the Use of COTS Devices in Space Systems under Consideration of Radiation Effects. Electronics **10**, 1008 (2021). https://doi.org/10.3390/electronics10091008.
4. Schwierz, A., Forsberg, H.: Assurance case to structure COTS hardware component assurance for safety-critical avionics. In: IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), pp. 1-10. Institute of Electrical and Electronics Engineers Inc. (2018). https://doi.org/10.1109/DASC.2018.8569774.
5. Gamazo-Real, J.C., Zamorano-Flores, J.R., Sanz-Andrés, Á.: Integration of COTS processing architectures in small satellites for onboard computing using fault injection testing methodology. In: Software Engineering and Formal Methods (SEFM 2021 Collocated Workshops), pp. 333–347. Springer (2022). https://doi.org/10.1007/978-3-031-12429-7_24
6. Perez-Cerrolaza, J., Abella, J., Kosmidis, L., Calderon, A.J., Cazorla, F., Flores, J.L.: GPU devices for safety-critical systems: a survey. ACM Comput. Surv. **55**, 147 (2023). https://doi.org/10.1145/3549526
7. Lee, S., Kang, W., Bertogna, M., Chwa, H.S., Lee, J.: Timing guarantees for inference of AI models in embedded systems. In: Real-Time Systems, pp. 259–267. Springer (2025). https://doi.org/10.1007/s11241-025-09445-9
8. Raj, C.A.: Model-based approaches in safety-critical embedded system design. European J. Comput. Sci. Inform. Technol. **13**, 30–41 (2025) https://doi.org/10.37745/ejcsit.2013/vol13n203041
9. Rahman, Q.M., Corke, P., Dayoub, F.: Run-time monitoring of machine learning for robotic perception: a survey of emerging trends. IEEE Access. **9**, 20067–20075 (2021). https://doi.org/10.1109/ACCESS.2021.3055015
10. Neto, A.V.S., Camargo, J.B., Almeida, J.R., Cugnasca, P.S.: Safety Assurance of Artificial Intelligence-Based Systems: a systematic literature review on the state of the art and guidelines for future work. IEEE Access. **10**, 130733–130770 (2022). https://doi.org/10.1109/ACCESS.2022.3229233

11. Yalcinkaya, B., Torfah, H., Desai, A., Seshia, S.A.: Ulgen: a runtime assurance framework for programming safe cyber-physical systems. IEEE Trans. Comput.-Aided Design Integr. Circ. Syst. **42**, 3679–3692 (2023). https://doi.org/10.1109/TCAD.2023.3246386
12. Hobbs, K.L., Mote, M.L., Abate, M.C.L., Coogan, S.D., Feron, E.M.: Runtime Assurance for Safety-Critical Systems: an introduction to safety filtering approaches for complex control systems. IEEE Control. Syst. **43**, 28–65 (2023). https://doi.org/10.1109/MCS.2023.3234380
13. Instituto Universitario "Ignacio Da Riva" (IDR-UPM) of the Universidad Politécnica de Madrid: UPMSat-2 Microsatellite, https://www.idr.upm.es/en/upm-sat2. Accessed 19 Mar 2026
14. Barbirotta, M. et al.: Fault-Tolerant Hardware Acceleration for High-Performance Edge-Computing Nodes. Electronics. **12**, 3574 (2023) https://doi.org/10.3390/electronics12173574
15. Urblik, L., Kajati, E., Papcun, P., Zolotová, I.: Containerization in Edge Intelligence: A Review. Electronics **13**, 1335 (2024). https://doi.org/10.3390/electronics13071335.
16. Biondi, A., Nesti, F., Cicero, G., Casini, D., Buttazzo, G.: A safe, secure, and predictable software architecture for deep learning in safety-critical systems. IEEE Embed. Syst. Lett. **12**, 78–82 (2020). https://doi.org/10.1109/LES.2019.2953253
17. Furano, G. et al.: Towards the use of artificial intelligence on the edge in space systems: challenges and opportunities. IEEE Aerosp. Electron. Syst. Mag. **35**, 44–56 (2020). https://doi.org/10.1109/MAES.2020.3008468
18. Ma, K. et al.: Radiation-induced degradation analysis and reliability modeling of COTS ADCs for space-borne miniature fiber-optic gyroscopes. IEEE Trans. Instrum. Meas. **70** (2021). https://doi.org/10.1109/TIM.2021.3054419
19. Brunetti, G., Campiti, G., Tagliente, M., Ciminelli, C.: COTS devices for space missions in LEO. IEEE Access. **12**, 76478–76514 (2024). https://doi.org/10.1109/ACCESS.2024.3405373
20. Slagel, J.T., White, L.M., Dutle, A., Muñoz, C.A., Crespo, N.: A formal verification framework for runtime assurance. In: NASA Formal Methods (NFM), pp. 322–328. Springer (2024). https://doi.org/10.1007/978-3-031-60698-4_19
21. Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., Topcu, U.: Safe reinforcement learning via shielding. In: 32nd AAAI Conference on Artificial Intelligence, AAAI 2018. pp. 2669–2678. AAAI press (2018). https://doi.org/10.1609/aaai.v32i1.11797.
22. Nigam, V., Talcott, C.: Automating recoverability proofs for cyber-physical systems with runtime assurance architectures. In: Theoretical Aspects of Software Engineering (TASE), pp. 1–19. Springer (2023). https://doi.org/10.1007/978-3-031-35257-7_1
23. Swihart, D.E., Griffin, E.: Automatic ground collision avoidance system (AutoGCAS). In: Proceedings of the 13th WSEAS international conference on Systems (2009).
24. Angirasa, D., Ayyaswamy, P.S.: Review of evaluation methodologies for satellite exterior materials in low earth orbit. J. Spacecr. Rocket. **51**, 750–761 (2014). https://doi.org/10.2514/1.A32742
25. Wang, Q., Chen, X., Qi, Q., Li, M., Gerstacker, W.: Multiple-satellite cooperative information communication and location sensing in LEO satellite constellations. IEEE Trans. Wirel. Commun. **24**, 3346–3361 (2025). https://doi.org/10.1109/TWC.2025.3530083
26. Osoro, O.B., Oughton, E.J.: A techno-economic framework for satellite networks applied to low earth orbit constellations: assessing Starlink, OneWeb and Kuiper. IEEE Access. **9**, 141611–141625 (2021). https://doi.org/10.1109/ACCESS.2021.3119634
27. Wei, Y., Li, H., Du, X.: An efficient LEO global navigation constellation design based on walker constellation. In: 2020 IEEE Computing, Communications and IoT Applications (ComComAp), pp. 1–6. Institute of Electrical and Electronics Engineers Inc. (2020). https://doi.org/10.1109/ComComAp51192.2020.9398888

28. Miller, K. et al.: Optimal runtime assurance via reinforcement learning. In: ACM/IEEE 15th International Conference on Cyber-Physical Systems (ICCPS), pp. 67–76. Institute of Electrical and Electronics Engineers Inc. (2024). https://doi.org/10.1109/ICCPS61052.2024.00013

Surgical Instrument Tracking: a Lightweight YOLO Approach Across Diverse Domains

Nathalia Frederico[1(✉)], Rafael Miranda[1], Saulo Silva[2], and Jefferson Morais[1]

[1] Federal University of Pará (UFPA), Belém, Brazil
almeidadifre@gmail.com, rafaeldoriamm@gmail.com, jmorais@ufpa.br
[2] Instituto Federal de Goiás (IFG), Luziânia, Brazil
saulo.silva@ifg.edu.br

Abstract. Artificial intelligence has the potential to improve robotic and laparoscopic surgery through accurate surgical instrument tracking. However, widespread clinical implementation is currently hindered by complex visual interferences, such as bleeding and smoke, and the prohibitive computational costs of state-of-the-art models. Therefore, this paper presents a framework to unify three public surgical datasets (Cholec, Robust, and Badilla) by standardizing annotations (bounding boxes, keypoints, and semantic masks) to the YOLO format. Using the unified dataset, a lightweight YOLOv11-s architecture was trained and evaluated. This integration aims to enhance the generalization of the model in various surgical procedures and clinical domains, mitigating the limitations of single-dataset training. The model achieved a mean Average Precision (mAP@0.5) of 90.2%, and the inference speed was 5.1 ms per frame. The system demonstrated resilience to severe class imbalances that are natural in surgical scenarios, maintaining high accuracy even for underrepresented instruments. The experimental analysis indicates computational viability in low-resource environments, as validated by retrospective data. Further research can employ data augmentation in minority classes to achieve robustness on a larger dataset and deployment in real clinical scenes, focusing on interoperability under restricted operational settings.

Keywords: Surgical dataset integration · Yolov11 · Low-resource tracking

1 Introduction

Robotic surgery is a constantly growing trend in medicine. Encouraged by its potential to improve surgical results in terms of predictability, decision-making, precision, safety, and time optimization [14].

Solutions based on artificial intelligence (AI), which are still in the early stages of clinical integration, have a promising future in robotic surgery. However, concerns about reliability exist among the medical community [51].

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 461–477, 2026.
https://doi.org/10.1007/978-3-032-30491-9_29

These concerns converge on one of the most demanding areas of surgical practice: the precise tracking of surgical instruments, given the dynamic and complex context that involves various types of instruments and the challenging visual conditions involved in their handling, such as bleeding, smoke, and variable lighting [44].

Patient interest in robotic surgery continues to increase, resulting in better results and a shorter hospital stay [46]. To achieve an interface between surgeons and robots, the current paradigm suggests the need for integrating AI and its new ways of interpreting modern surgery [41].

The history of robotic surgery dates back to the 1980s, with various models designed for this purpose, from the first, the Programmable Universal Machine for Assembly (PUMA), to the most established and widely used model worldwide, the Da Vinci Surgical System [28]. The implementation of robotic systems in healthcare is characterized by exponential growth [31].

Currently, AI is transforming robotic surgery by integrating machine learning (ML) and computer vision (CV) algorithms to refine accuracy, assist in preoperative planning, and, in the future, expand the level of medical and care autonomy in surgical procedures [1].

Controlled studies have demonstrated that the exhaustive training of surgeons through extensive simulator training shortens the learning curve for skill and precision, potentially reducing the frequency of accidents and complications [12]. The ethical priority in the surgical context should be considered: "never the first time on the patient" [7].

In this context, the challenge itself lies in the transition from computer lab models to simulation centers, considering that expensive computational costs and a lack of metrics for educational utility are the main barriers to implementation in educational institutions.

To address these challenges, this study evaluates the YOLOv11-s architecture to optimize the trade-off between latency and precision. By unifying specialized public datasets, we propose a robust methodology that covers real laparoscopic, robotic, and simulated scenarios.

The remainder of this paper is organized as follows: Sect. 2 reviews related works; Sect. 3 details the experimental methodology; Sect. 4 discusses the results; Sect. 5 addresses limitations; Sect. 6 concludes the paper; Sect. 7 presents an ethics statement; and finally, Sect. 8 displays the availability of data and code.

2 Related Works

2.1 Systematic Search

The exponential growth of the field motivated a literature review on the recent period (2021–2025). The strings that involved four dimensions (application, problem, technique and resource) were subsequently combined: ("Surgical Instrument*" OR "Surgical tool*" OR "Surgical Simulation*") AND ("Clutter*" OR "Bin-picking" OR "Occlusion*") AND ("Machine Learning" OR "Deep Learning") AND ("Dataset*" OR "Data collection" OR "Training set*").

Focusing on ML and CV models applied to healthcare issues, the literature searches deploy important databases in the fields of Engineering and Computing, as follows: IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink, Web of Science, PubMed, and Google Scholar.

PRISMA (Preferred Reporting items for Systematic Reviews and Meta-Ana- lyses) guidelines constitute a standard protocol for identifying and screening studies [40]. Its selection justified the goal of presenting transparency and methodological rigor in the process of searching the bibliographic bases of this study.

Through the steps of Identify, Screen, Elect, and Synthesize, the protocol allows comparison with the following previously established criteria. Four inclusion criteria were used: published in English; recent (2021–2025); open access full-text; and consistent with the objectives of this work. Five exclusion criteria were used: secondary literature, such as literature reviews, systematic reviews, meta-analyses, editorials, or surveys; incomplete access; duplicates; non related studies; and gray literature.

From an initial pool of 219 screened records, 48 articles were ultimately included for qualitative synthesis after a strict application of predefined inclusion and exclusion criteria.

After the systematic screening of 219 studies, 48 were ultimately included after their eligibility was assessed on the basis of predefined inclusion and exclusion criteria. The following section begins the analysis of state-of-the-art (SOTA) methods.

2.2 SOTA Models for Surgical Instrument Tracking

Surgical instrument tracking has evolved from feature-based manual methods to deep learning architectures capable of navigating the complexities of the intraoperative environment. The following subsections provide a comparative synthesis of the most relevant studies from the past five years to provide a thorough overview of the current state of the art.

The technical literature was organized along three main lines: the Task performed (from basic detection to Surgical Workflow Recognition), Data Availability, and Model Architecture.

The Education purpose topic also looked at studies that were specifically designed for medical training, skill testing, or feedback from a simulator. We chose to contextualize different approaches, ranging from real-time assisted methodologies to designs with purely pedagogical purposes.

Considering the following performance metrics grouped by methodological function, studies were compared using this general framework:

- **Classification:** Accuracy (Acc), Precision (Prec), Recall (Rec), and F1-Score are calculated to measure the level of ability to find patterns (categories or identities).
- **Overlap:** Sørensen–Dice Coefficient and Intersection over Union (IoU) are often used to point the match of segmentation boundaries.

- **Error:** Statistical Errors demonstrate changes in position or orientation of the objects by the Mean Squared Error (MSE), Root Mean Square Error (RMSE), and Pose Error.
- **Flow:** Detection and tracking tasks consist on verify spatial localization and temporal coherence, it may uses the mean Average Precision (mAP), Higher Order Tracking Accuracy (HOTA), and Multi-Object Tracking Accuracy (MOTA).
- **Speed:** At last, to indicate computational efficiency, its imperative the urge to the system perform in a short time given a clinical situation, therefore, we use the Frames Per Second (FPS) rate for this purpose.

One-stage detectors are becoming popular on real-time detection scene, as simulator's feedback requires low latency, these architectures in turn ensure a single end-to-end pipeline, by object localization and classification into a single pass through the network.

In this scenario, the YOLO (You Only Look Once) family has become the most prevalent technique [8,9,11,16,24,30,42–45,49,56,59,60]. Recent studies have seek for balancing elevated inference speed (FPS) and localization precision (Flow). Additionally, innovations such as Omni-Dimensional Dynamic Convolutions (ODConv) and Large Receptive Field blocks (RepLK), were design to capture both fine details and larger structures [49].

For tasks requiring precise pixel-level delimitation, architectures based on U-Net [3,23,25,52,58,60] and DeepLabV3+ [13,23] remain the primary benchmarks. The integration of robust backbones, such as ResNet-50/101 [2,5,13, 17,19,22,32,47,48,59] and ConvNeXt [20,59], aims to extract features that are resilient to visual interferences.

Specific mechanisms like SEBlock (Squeeze-and-Excitation) and CBAM (Convolutional Block Attention Module) were helpful to ignore background noise and to carefully observe reflective properties of metal surgical tools [2,13].

In order to handle natural switch between similar tools, it is important to keep identities consistent in video sequences. Recurrent neural networks as Long Short-Term Memory (LSTM) and its variants enables the understanding of temporal surgical workflow [10,15,29,50,54]. ByteTrack and BoT-SORT work as association algorithms for strong tracking [42,44].

A recent study calls attention to the need of datasets that shows clinically pertinent trajectories for training models, especially focusing to tracking continuity on instruments frequently switching [44].

The moment trend is the Foundation Models, like Segment Anything Model (SAM) [36,47,57], that act on zero-shot segmentation, in other words, presents an accurate segment of objects or classes unknown during training, relying on general visual features or external tips. A lightweight version of that architecture is the MobileSAM, that is making an effort to real-time apps on less processing settings [57].

On the other hand, there is also a combination of hybrid techniques (Transformers like Swin-UNet with Convolutional Neural Networks) to balance global attention with local efficiency [20,36,56,59]. Presenting a good performance on

hard tasks such 3D pose estimation, which requires spatial context and specific fine details.

2.3 Research Gap and Challenges

The SOTA analysis established that significant advances in raw accuracy occurred, however, there are still challenges to address, such as: High Computational Dependency, current SOTA models, especially Foundation Models, often need high power Graphic Process Units (GPUs) [35], making its widespread use in educational institutions unfeasible. Requiring architectures that provides energy efficiency and low latency on modest hardware.

As educational concerns stands out: Generalization power (Models simulated datasets struggle to generalize on real-world domain gap); and Pedagogical metrics (few studies integrate tracking with motion economy (Path Length) or smoothness (Jerk) [10,38]. Such gaps limits the capacity to provide actionable educational feedback, since there is a disconnection between CV metrics (mAP, Dice) and clinical proficiency indicators.

Due to these questions, the experimental method of this research shift the paradigm of "maximum precision at any cost" to "resilient efficiency".

3 Experimental Design, Materials and Methods

The methodological framework employed was made under the Cross-Industry Standard Process for Data Mining (CRISP-DM), an worldwide accepted guide for methodical and reproducible researches about machine learning models [21]. That chosen was driven to break the usual 'black-box' algorithms and so facilitate posterior audit and validation, considering the principles of Explainable Artificial Intelligence (XAI) [4], and always subordinate to clinical reason [37].

An adaptation of the CRISP-DM protocol follows these steps checklist: Problem in Business Terms; Data Understanding and Preparation; Model and Evalute; Deployment [18]. These steps were adapted to the surgical context discussed, so that the development of an object detection solution with few computational resources will facilitate future adoption in educational environments, consequently improving the surgical learning curve and, in turn, making robot-assisted procedures safer.

The experimental methodology includes two principle targets: Low-resource Optimization (by apllying Nano and Small versions of one-stage detector YOLO-v11 and examine its performance under hardware limitations, aiming to found settings of an optimal balance between FPS and efficient classification); and Unify Diverse Domains (considering the benchmarks identified in the Related Works section, a data-centric pipeline evaluate the model robustness across a range of scenarios).

The experiments were held on Google Colab, linked to Google Drive as a environment to manage data. The computer system available contains NVIDIA

T4 GPU with 16 GB of VRAM, capable of ran deep learning frameworks based on Python.

While the NVIDIA T4 is a high-performance accelerator utilized here for efficient training and validation, the selected YOLOv11-s architecture is inherently lightweight. This choice aims to balance high-speed inference with low computational overhead, making it significantly more efficient for real-time surgical applications compared to larger, more resource-intensive models.

All workflow and supplementary materials were organized to ensure methodological transparency and enable complete reproducibility [after the pair revision is finished, to avoid author identification, a link to the repository will be shared]. Primary datasets used in the development of this work are properly cited and referenced according to their original license agreement.

3.1 Business and Healthcare Understanding

Initial stage is to translate clinical needs into business terms. Track instruments is not an end in itself, it is a way to enable Computer-Assisted Interventions (CAI), including skill assessment, safety zone estimation, and create an efficient interface people-machine [44]. As to turn raw video from simulators into useful teaching feedback, to mitigate problems with cost and technical difficulty.

Accurate tracking grant an effective automated workflow identification [53]. Allows the system to discern between actions like suturing and dissection. In robotic surgery, instruments are categorized by their technical function, allying their visual design, examples include Graspers (change shape when pulled back [62]), Dissectors (like Hooks, that often produce surgical smoke [43]), and Needle Holders (which requires millimeter accuracy for Jerk and Path Length metrics [38]).

There is also difficult optical properties, stainless steel's specular reflections, tissues or fluids that frequently are blocking the view [55,58]. Finally, shift from laboratory models to real-time application call for minimal latency, as also to prevent the disruption in educational process [42].

On this occasion, "business terms" means a system that is technically accurate, useful for teaching, and financially possible.

3.2 Data Understanding

Google Colab was used to technical audit the data repositories that were found in the literature. This stage aimed to select benchmarks datasets that would help built a robust model capable of deal with a range of scenarios.

In Table 1 we present the study carried out in five public repositories, in order to posteriously choose for this work. These datasets demonstrated different surgical approaches, from controlled environments for multi-class complex clinical scenarios.

OSI26 is a dataset designed to be a balanced baseline, with 66 examples for each class. This consistency allows the model to learn rare instrument features

Table 1. Technical Review and Description of Surgical Datasets

Dataset	Vol.	Cls	Inst.	Type	Src	Res.	Challenges	Bal.
OSI26 [49]	0.23	26	1,716	Image	Sim.	1080p	Low: Occ., reflex	Uni.
CholecTrack20 [44]	32.9	7	65,247	Video	Real	480p	High: Blood, Smoke	Unb.
ROBUST-MIPS [22]	3.29	10	11,560	Seq.	Real	540p	High: Sev. Occ.	Unb.
m2cai16 [26]	0.13	7	3,929	Image	Real	334p	Mod: Blur, Overlap	Unb.
Badilla-S. [5]	0.98	18	1,019	Mask	R/S	480p	High: Intra-op noise	Unb.

Abbreviations: Vol: Volume (GB); **Cls**: Classes; **Inst**: Instances; **Src**: Source; **Res**: Resolution; **Occ**: Occlusion; **Bal**: Balance (Uni: Uniform/Unb: Unbalanced); **Sim.**: Simulated; **R/S**: Real/Synthetic.

giving the same importance as common ones. Originally, it had 1,716 instances, but after the authors add more data, the augmentation technique was to better separate overlapping tools in simulated environments [49].

The next evaluated was CholecTrack20, in contrast with the previous dataset, this presented a high class imbalance, natural in real world surgical routine. On this particular set, "Grasper" was the prevalent identity (52.62%), following by "Hook" (32.61%), "Bipolar" (3.61%) and "Specimen Bag" (5.37%), "Scissors", "Clipper" and "Irrigator" got less than 2%, along with visual challenges (such blur, blood, smoke, occlusion) [44].

ROBUST-MIPS focusing on pose estimation in complex structures, not just classify identities but also make effort to understand the global scene around among partial occlusions [22]. Unlike the other datasets, ROBUST-MIPS organizes annotations by surgical procedure type (Proctocolectomy, Rectal Resection, and Sigmoid), with instrument identities encoded at the keypoint level. This required an additional extraction step during Data Preparation to map skeletal nodes to the Universal Taxonomy.

Finally, the m2cai16-tool-locations provides a historical baseline for real-world bounding boxes, while the dataset by Badilla-Solórzano et al. utilizes Mask-Based Object Insertion (MBOI) to create ∼27,000 synthetic images, focusing on edge precision and semantic segmentation of specific robotic components [5,26].

3.3 Data Preparation

This stage focuses on the structural unification of the CholecTrack20 [44], ROBU- ST-MIPS [22], and Badilla-Solórzano et al. [5] datasets. To mitigate cloud-storage latency and Input/Output (I/O) overhead associated with large-scale surgical data, we implemented a partitioned storage architecture.

By segmenting the Unified Dataset into sub-directories (keeping each folder below the 10,000-item synchronization limit of Google Drive), we ensured high-throughput data streaming and avoided the common timeout errors of distributed file systems.

A core contribution of this preparation is the adoption of a Universal Taxonomy of 7 classes: Grasper, Hook, Scissors, Clipper, Irrigator, Bipolar, and Specimen Bag. This taxonomy considered the recent pioneer dataset for multi-perspective tracking by Nwoye et al. [44], following this protocol, additional categories from ROBUST-MIPS and Badilla-Solórzano datasets were semantically mapped and filtered to align with the selected Ground Truth.

That preparation was necessary to ensure that robotic, laparoscopic and simulated environments could all functioning together. First, each dataset had a type of annotation: sequential JSON (CholecTrack20), instance IDs (ROBUST-MIPS), and binary masks (Badilla-Solórzano). Making necessary a geometry extraction algorithm to change vector and matrix data into the YOLO PyTorch format (.txt), converging coordinates to normalized XYWH values based on the size of the frame [27].

Also, all the frames had to be resized to the standard 640×640 resolution, conforming YOLOv11 architecture demand [42]. As a low-resource environment, using less VRAM, that configuration preserve the tool-tip details.

Before the ultimate training with full data, we run a pilot phase with a stratified subset of 361 images (splitting in 70/20/10 for training, validation and testing, respectively) to validate hyperparameter configurations and taxonomic correct correspondence.

3.4 Modeling and Evaluation

The data-centric pipeline followed the modules of: Data Understanding (script made to study the architectural profile, verify spatial integrity and class imbalances of pre-selected datasets); Data Preparation (this module worked on math normalization, conversion of the different annotations into the standard YOLO format); Preliminary Analysis (the pilot phase of validation using a sample of the unified dataset); Model Training (full dataset split in 80% training and 20% validation to run the neural network training loop); Finally, the module Zero-Shot functioning as a test to validate the generalization of the model in an unannotated surgical footage.

To the pilot run, YOLOv11-n (Nano) architecture was execute, as the most recent version of a lightweight approach, with 2.59 million parameters and a lightweight computational footprint of 6.4 GFLOPs [27,61]. This first execution run for 10 epochs, consolidates the 7-class taxonomic and the model performance on a small scale, validating the pre-processing made.

As the pilot was completed, the ultimate detection task scaled to YOLOv11-s (Small) architecture, to better balance between mAp and Speed, essential in surgical procedures. The standard pre-trained weights of MS COCO dataset were used by Transfer Learning, that process allows fine-tuning focus on the specific surgical domain of the present research, considering that the weights ensure model already learned patterns (edges and textures) from millions of natural images before [33].

The ‘s’ (small) variant of the YOLOv11 family was specifically selected to ensure a lightweight framework. This version provides a strategic optimization,

delivering the precision required for delicate surgical tracking while maintaining a small parameter footprint and high frames-per-second (FPS) rates.

Following, YOLOv11 architecture have itself an automated augmentation pipeline, that improve the resistance of the system among expected visual challenges, by making a Mosaic Augmentation, where four aleatory training images are set together, as to avoid the model memorizing scenarios and focusing on the tools in different scenarios and spatial localizations, as so looking closer to their texture and shape by HSV color jittering [6].

At this point, the final training ran over 50 epochs with a batch size of 16, AdamW algorithm contributed to weight decay regularization, that mechanism optimize the learning rate by minimize losses on localization and classification [34], this stage started in a learning rate of 0.000909 and a momentum of 0.9.

According to the protocol for surgical detection settings evaluation [44], we employed the following multi-metric check to evaluate spatial accuracy and computational efficiency: mAP (mAP50 tested the capacity to identify tools that are 50% overlapping each other and mAP50-95 gave a complete evaluation across IoU thresholds from 50% to 95%, showing the level of precision for tool-tip track [33]; Prec and Rec (trade-off between false positives and false negatives, crucial for clinical safety [37]); and F1-score (confirm if the model is reliable across the 7-class taxonomy by calculate the harmonic mean between Prec and Rec).

A common issue in the process of recognize surgical instruments is the similarity inherent of them [62], for this matter, the Confusion Matrix checks the possible mistakes (like identify a "Irrigator" when it is a "Scissor").

Its also important to evaluate the practicability of low-resource solutions, outlined by [42]: Inference Speed (ms/frame) for measure the real-time feedback without any lag, and Computational Complexity (GFLOPs) for acknowledgment of computing environments required.

This work thoroughly addresses the modeling and evaluation stages of the computational pipeline; however, it is a fundamentally experimental analysis validated through retrospective datasets. The last step in the method, Deployment, demands further experimentation.

To turn this architecture from a computational proof-of-concept into a useful clinical tool, the trained YOLOv11-s model will need to be added directly to the video processing pipelines in operating rooms. Focusing on testing the model during live and planned intraoperative procedures. This will ensure that it works well with existing robotic and laparoscopic hardware.

4 Results and Discussion

Image pre-processing was done on a localized Solid State Drive (SSD) runtime to make sure that the computer worked quickly and that the data were accurate. This method got around the built-in latency and metadata synchronization problems of distributed cloud storage, which made it possible to resize and verify images quickly before putting them all together into a final dataset.

The unified YOLOv11-s model was able to detect things in the very diverse surgical dataset. The Precision-Recall (PR) Curve (Fig. 1) shows that the network got a mean Average Precision (mAP@0.5) of 0.902 for all instrument classes. The model also got the global F1-confidence score of 0.87 at a confidence threshold of 0.480. This means that it had a strong balance between precision (keeping false positives to a minimum) and recall (keeping false negatives to a minimum).

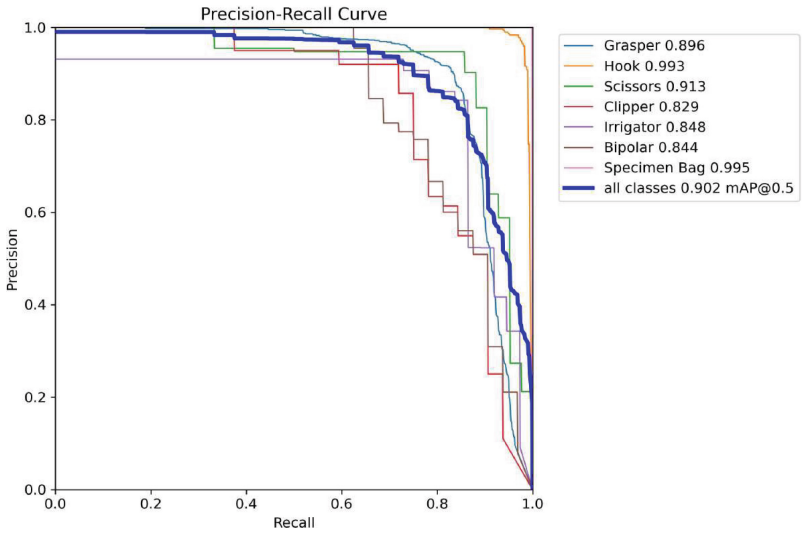


Fig. 1. Precision-Recall (PR) curve for the unified taxonomy (mAP@0.5: 90.1%). Detailed metrics in Table 2.

The model demonstrated significant resilience to the severe class imbalances inherent in the unified dataset. As detailed in Table 2, underrepresented classes such as 'Scissors', 'Hook', and 'Specimen Bag' achieved excellent mAP@0.5 scores of 91.2%, 99.3%, and 99.5%, respectively. This performance is visually corroborated by the PR curve, which maintains high stability and a large Area Under the Curve (AUC) across individual classes despite the extreme variation in sample representation.

We also qualitatively present the model's performance, as to enrich quantitative previous analysis, in Fig. 2, where its showed inference results of predicted bounding boxes on top of the real frames of the unified dataset used.

The network was able to frame several tools at once (like Grasper and Hook interactions) with tight bounding-box regression. The model consistently exhibited elevated classification confidence, irrespective of the background environment, effectively abstracting the mechanical essence of the instruments across diverse laparoscopic and robotic domains, undeterred by surgical smoke, tissue occlusion, or reflective glare.

Table 2. Per-class performance metrics for the unified surgical tracking model.

Class	Precision (P)	Recall (R)	mAP@0.5
Grasper	0.913	0.826	89.5%
Hook	0.976	0.974	99.3%
Scissors	0.946	0.838	91.2%
Clipper	0.846	0.750	82.5%
Irrigator	0.810	0.865	84.7%
Bipolar	0.843	0.688	84.3%
Specimen Bag	0.947	1.000	99.5%
All Classes	**0.897**	**0.849**	**90.1%**

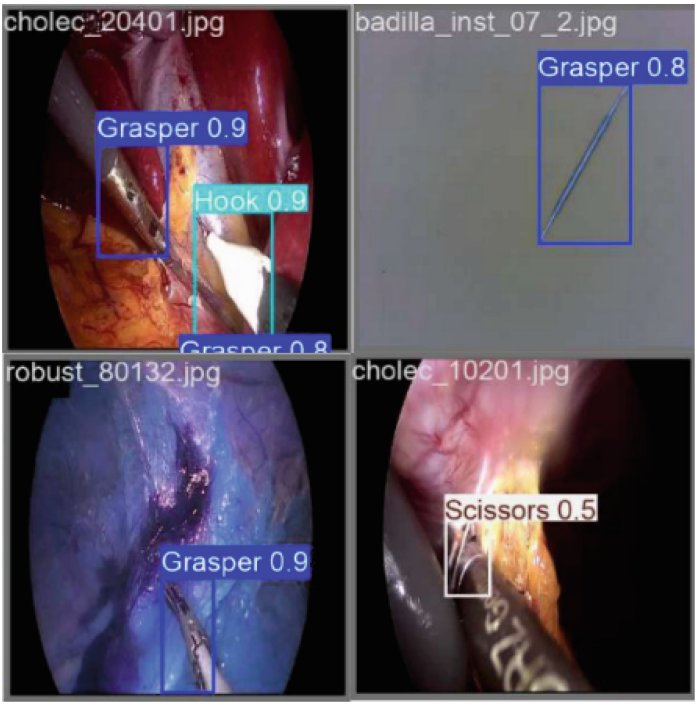


Fig. 2. Qualitative tracking results demonstrating cross-domain generalization. Top-left: CholecTrack20 (Grasper and Hook). Top-right: Badilla-Solórzano (synthetic background). Bottom-left: ROBUST-MIPS (colorectal view). Bottom-right: Accurate detection of a minority class (Scissors). The YOLOv11-s model maintains precise bounding box localization across distinct surgical environments.

An examination of the class distribution indicates a pronounced natural imbalance, with the 'Grasper' class prevailing in the dataset. This distribution

accurately reflects real-world surgical workflows, where graspers are used all the time to move tissue around, but scissors or clips are only used for a short time.

The model did not show any prediction bias toward the majority class, even though there was a big difference between the two groups. The Precision-Recall metrics show that the minority classes were very accurate. For example, "Scissors" had a mAP@0.5 of 91.3% and "Hook" had a mAP@0.5 of 99.3%.

The internal loss mechanisms of the YOLOv11 architecture (like Focal Loss) did a good job of punishing misclassifications of underrepresented classes. This made sure that the whole surgical toolkit could be used without overfitting to the most common instruments.

A zero-shot inference test was conducted using external, unannotated surgical footage to further validate the model's ability to generalize in real-world situations without any restrictions.

In accordance with the qualitative evaluation methodology established by Myo et al. [42], publicly accessible laparoscopic cholecystectomy videos from YouTube (Dr. R. K. Mishra's surgical channel [39]) were employed.

This evaluation substantiates the architecture's resilience, illustrating its capacity to precisely monitor instruments amidst novel illumination profiles and hardware discrepancies absent in the initial training distribution.

5 Limitations and Error Analysis OR Limitations and Gaps to Address

The OSI26 dataset was very useful for finding surgical instruments, but it had a lot of methodological problems during the integration phase. The main problem is that its metadata doesn't have standard class names; it only has numbers for its 26 categories.

Since there is no official documentation that links these IDs to specific surgical instruments, keeping this dataset would hurt the taxonomic consistency needed for a multi-dataset comparative analysis. Because of this, the OSI26 was left out to make sure that the evaluation of the proposed model was reliable and could be repeated.

Also, integrating large-scale surgical datasets was hard because of problems with the latency of cloud-based file systems. The distributed cloud environment often had I/O bottlenecks and synchronization latencies because there were so many small metadata files and the raw surgical frames were so high resolution.

To make sure the data was safe and to get around these problems, a localized caching strategy had to be put in place. This meant moving compressed archives to a SSD local runtime for atomic processing. These problems show how important it is to manage data pipelines well in computer vision tasks, where the physical limits of cloud storage can make things take a lot longer.

The Badilla-Solórzano dataset encodes class identity as raw pixel values in grayscale masks without a publicly available ID-to-instrument mapping. The correspondence adopted here was derived through structural analysis and cross-referencing with the original publication, mapping six out of eighteen detected pixel classes to the Universal Taxonomy.

The remaining twelve classes were excluded to preserve taxonomic consistency, resulting in 116 annotated frames from this source. The assumed mapping has not been independently verified with the original authors, representing a potential source of label noise in the Badilla-sourced training samples.

The high mAP@0.5 achieved across all classes suggests that the derived mapping is consistent with the instrument categories described in the original publication.

Qualitative evaluation also pointed out some specific cases where the visual model had trouble. For example, in the synthetic subset (which has isolated instruments on black backgrounds), a closed surgical scissor was sometimes wrongly labeled as a "Irrigator" on Badilla's dataset.

This mistake can be explained geometrically: when there is only a dark background and no anatomical tissue to give scale or functional cues, a closed scissor looks like the long, narrow metal shaft of an irrigator. Including these failure cases shows the real limits of the architecture and shows how important it is to have contextual anatomical cues when doing complex visual recognition tasks.

To mitigate this specific misclassification, future training protocols should incorporate targeted data augmentation—such as synthetic background randomization—and background-aware learning strategies. This would prevent the network from overfitting to homogeneous dark contexts and force it to rely strictly on the geometric features of the instrument tips.

6 Conclusions

This study introduced a data-driven approach for the identification of surgical instruments across diverse domains. A YOLOv11-s architecture was trained to identify tools in different surgical settings by geometrically combining different types of annotations—bounding boxes, keypoints, and semantic masks—from the Cholec, Robust, and Badilla datasets.

The unified model had a mAP@0.5 of 90.2% and an inference speed of 5.1 milliseconds per frame, which means it could be used for real-time intraoperative processing. Furthermore, the model maintained precision for minority classes such as 'Scissors' and 'Hook', demonstrating stability against the class imbalance inherent to surgical workflows.

Promising directions for future research include the implementation of targeted data augmentation techniques, such as generative synthetic oversampling, to balance minority classes during training; and the prospective deployment of the architecture in clinical and educational settings to evaluate interoperability under operational constraints.

Ethics Statement. The authors affirm that the present study complies with the ethical standards required for publication in the ICCSA. This research does not involve human participants, animal experimentation, or data derived from social media platforms. The datasets utilized in this study are publicly available and comply with the principles of Open Access scientific research.

References

1. Andrade, T.N., Morais, O.J., Ramirez, B.F.D.: Artificial intelligence in medicine: Surgical robots, digital neural networks, health big data, infodemiology, and bioprinting for tissue regeneration. In: Albino, J.P., Valente, V.C.P.N. (eds.) Artificial Intelligence and its Interdisciplinary Applications, chap. 11. e-Publicar, , Curitiba, Brazil (2023). https://doi.org/10.47402/ed.ep.c2023210411201
2. Ángeles Cerón, J.C., et al.: Real-time instance segmentation of surgical instruments using attention and multi-scale feature fusion. Med. Image Anal. **81**, 102569 (2022). https://doi.org/10.1016/j.media.2022.102569
3. Arevalo-Ancona, R.E., et al.: Advanced dual-branch U-Net decoder for precise and robust surgical instrument and organ segmentation. Biomed. Signal Process. Control (2025). https://doi.org/10.1016/j.bspc.2025.108348. in Press
4. Arrieta, A.B., et al.: Explainable Artificial Intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf. Fusion **58**, 82–115 (2020). https://doi.org/10.1016/j.inffus.2019.12.012
5. Badilla-Solórzano, J., et al.: Deep-learning-based instrument detection for intraoperative robotic assistance. Int. J. Comput. Assisted Radiol. Surg. **17**(9), 1685–1695 (2022). https://doi.org/10.1007/s11548-022-02715-y
6. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934 (2020), https://arxiv.org/abs/2004.10934
7. Bresler, L., et al.: Residency training in robotic surgery: The role of simulation. J. Visceral Surgery **157**(3S2), S123–S129 (2020). https://doi.org/10.1016/j.jviscsurg.2020.03.006, supplement 2
8. Chellamani, G.K.: A robust dataset for surgical instrument detection to aid autonomous robotic surgery. Data Brief **61**, 111798 (2025). https://doi.org/10.1016/j.dib.2025.111798
9. Chong, N., Yang, F.: CMMCAN: lightweight feature extraction and matching network for endoscopic images based on adaptive attention. Comput. Mater. Continua **79**(2), 3145–3164 (2024). https://doi.org/10.32604/cmc.2024.052217
10. Das, A.: PitVis-2023 challenge: workflow recognition in videos of endoscopic pituitary surgery. Med. Image Anal. **99**, 103716 (2025). https://doi.org/10.1016/j.media.2025.103716
11. Deol, E.S., et al.: Artificial intelligence model for automated surgical instrument detection and counting: An experimental proof-of-concept study. Patient Safety Surgery **18**(1), 24 (2024). https://doi.org/10.1186/s13037-024-00406-y
12. Domene, C.E.: Robotic surgery – a step towards the future. ABCD. Arquivos Brasileiros de Cirurgia Digestiva (São Paulo) **27**(4), 233–233 (2014). https://doi.org/10.1590/S0102-67202014000400001
13. Duan, X., Li, L., Li, X., Lyu, S., Ding, L., Li, C.: DINOv2-enhanced surgical instrument segmentation with lightweight CNN. Procedia Comput. Sci. **271**, 225–231 (2025). https://doi.org/10.1016/j.procs.2025.10.135
14. Duarte, L.D.C.: Artificial intelligence in surgery: the frontier between technological assistance and medical autonomy. Revista Caderno Pedagógico **22**(12), 1–17 (2025). https://doi.org/10.54033/cadpedv22n12-344
15. ElMoaqet, H., et al.: Deep learning architectures for single-label and multi-label surgical tool classification in minimally invasive surgeries. Appl. Sci. **15**(11), 6121 (2025). https://doi.org/10.3390/app15116121

16. Franco-González, I.T., et al.: Tracking 3D motion of instruments in microsurgery: a comparative study of stereoscopic marker-based vs. deep learning method for objective analysis of surgical skills. Inf. Med. Unlocked **51**, 101593 (2024). https://doi.org/10.1016/j.imu.2024.101593
17. Fujii, R.: Surgical tool detection in open surgery videos. Appl. Sci. **12**(20), 10473 (2022). https://doi.org/10.3390/app122010473
18. Géron, A.: Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd edn. O'Reilly Media, Sebastopol (2019)
19. Graëff, C., et al.: Introducing surgical workflow recognition in orthopaedic surgery with timestamp supervision. Comput. Biol. Med. **197**, 110995 (2025). https://doi.org/10.1016/j.compbiomed.2025.110995
20. Guan, B., et al.: Enhancing spatial perception in robot-assisted minimally invasive surgery with edge-preserving depth estimation and pose tracking. BMC Surgery **25**, 455 (2025). https://doi.org/10.1186/s12893-025-03198-9
21. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques, 4th edn. Elsevier, Amsterdam (2020)
22. Han, Z., et al.: ROBUST-MIPS: a combined skeletal pose and instance segmentation dataset for laparoscopic surgical instruments. Scientific Data (2026). https://doi.org/10.1038/s41597-026-06938-5
23. Hu, Z.: Automatic segmentation of surgical instruments in endoscopic spine surgery: a deep learning-based analysis. Asian J. Surg. **48**(3), 1977–1979 (2025). https://doi.org/10.1016/j.asjsur.2024.09.128
24. Hua, S., et al.: Deep learning-based recurrent laryngeal nerve detection during endoscopic thyroidectomy. Intelligent Medicine (August 2025). https://doi.org/10.1016/j.imed.2025.08.005, in Press
25. Jha, D.: Validating polyp and instrument segmentation methods in colonoscopy through Medico 2020 and MedAI 2021 challenges. Med. Image Anal. **99**, 103307 (2025). https://doi.org/10.1016/j.media.2024.103307
26. Jin, A., et al.: Tool detection and operative skill assessment in surgical videos using region-based convolutional neural networks. In: Proc. IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 691–699 (2018). https://doi.org/10.1109/WACV.2018.00081
27. Jocher, G., Qiu, J., Chaurasia, A.: YOLO11 by Ultralytics. Ultralytics (2023). https://github.com/ultralytics/ultralytics
28. Kezan, R.D., Chiavegatto, M.A., Marco, D.C.B.: Challenges and benefits of robotic surgery in the modern world. Revista Eletrônica Acervo Saúde **23**(11), e14031 (2023). https://doi.org/10.25248/reas.e14031.2023
29. Khalil, S., Shah, H.A., Bednarik, R.: Improving microsurgical suture training with automated phase recognition and skill assessment via deep learning. Comput. Biol. Med. **192**, 110238 (2025). https://doi.org/10.1016/j.compbiomed.2025.110238
30. Koskinen, J., et al.: Automated tool detection with deep learning for monitoring kinematics and eye-hand coordination in microsurgery. Comput. Biol. Med. **141**, 105121 (2022). https://doi.org/10.1016/j.compbiomed.2021.105121
31. Lavado, D.M.: Sorting Surgical Tools from a Cluttered Tray: Object Detection and Occlusion Reasoning, Coimbra, Portugal (2018). Master's dissertation, Faculdade de Ciências e Tecnologia
32. Lavanchy, J.L.: Automation of surgical skill assessment using a three-stage machine learning algorithm. Sci. Rep. **11**, 5197 (2021). https://doi.org/10.1038/s41598-021-84295-6

33. Lin, T.Y.: Microsoft COCO: Common Objects in Context. In: Proc. European Conference on Computer Vision (ECCV), pp. 740–755. Springer (2014). https://doi.org/10.1007/978-3-319-10602-1_48
34. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: Proc. 7th International Conference on Learning Representations (ICLR). New Orleans, LA, USA (2019). https://openreview.net/forum?id=Bkg6RiCqY7
35. Mahmood, T., Cho, S.W., Park, K.R.: DSRD-Net: dual-stream residual dense network for semantic segmentation of instruments in robot-assisted surgery. Expert Syst. Appl. **202**, 117420 (202https://doi.org/10.1016/j.eswa.2022.117420
36. Matasyoh, N.M., Mathis-Ullrich, F., Zeineldin, R.A.: SAMSurg: surgical instrument segmentation in robotic surgeries using vision foundation model. IEEE Access **12**, 193950–193959 (2024). https://doi.org/10.1109/ACCESS.2024.3520386
37. Medina, F.M.: Desafios da Formação Médica na Modernidade: A Experiência da Sociedade Brasileira de Oftalmologia. Revista Brasileira de Oftalmologia **84**, e0076 (2025). https://doi.org/10.37039/1982.8551.20250076
38. Mikhail, D., et al.: Quantitative analysis of instrument motion paths in cataract surgery across a resident's training. Ophthalmology Sci. **6**(2), 101014 (2026). https://doi.org/10.1016/j.xops.2025.101014
39. Mishra, D.R.K.: D. R. K. Mishra - YouTube Channel. YouTube (2026) https://www.youtube.com/channel/UC5GRDQvh5Ze-agGcaMeveKw
40. Moher, D., et al.: Preferred reporting items for systematic review and meta-analysis protocols (prisma-p) 2015 statement. Systematic Reviews **4**(1), 1 (jan 2015https://doi.org/10.1186/2046-4053-4-1
41. Morrell, A.L.G., et al.: The history of robotic surgery and its evolution: when illusion becomes reality. Revista do Colégio Brasileiro de Cirurgiões **48**, e20202798 (2021). https://doi.org/10.1590/0100-6991e-20202798
42. Myo, N.N., et al.: A two-point association tracking system incorporated with YOLOv11 for real-time visual tracking of laparoscopic surgical instruments. IEEE Access **13**, 12225–12238 (2025). https://doi.org/10.1109/ACCESS.2025.3529710
43. Nwoye, C.I., Padoy, N.: SurgiTrack: Fine-grained multi-class multi-tool tracking in surgical videos. Med. Image Anal. **101**, 103438 (2025). https://doi.org/10.1016/j.media.2024.103438
44. Nwoye, C.I.: Cholectrack20: A multi-perspective tracking dataset for surgical tools. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8942–8952 (2025)
45. Olei, S., et al.: The AENEAS project: Intraoperative anatomical guidance through real-time landmark detection using machine vision. Mayo Clinic Proceedings: Digital Health **4**(1), 100308 (2026). https://doi.org/10.1016/j.mcpdig.2025.100308
46. Pereira-Arias, J.G., et al.: How to build a robotic program. Archivos Españoles de Urología **72**(3), 227–238 (2019). pMID: 30945649
47. Rai, U., Xu, H., Giannarou, S.: SurgPose: Generalisable surgical instrument pose estimation using zero-shot learning and stereo vision. arXiv preprint arXiv:2505.11439, May 2025. https://doi.org/10.48550/arXiv.2505.11439, https://doi.org/10.48550/arXiv.2505.11439, preprint
48. Ramesh, S., et al.: Dissecting self-supervised learning methods for surgical computer vision. Med. Image Anal. **88**, 102844 (2023). https://doi.org/10.1016/j.media.2023.102844
49. Ran, B.: Surgical instrument detection algorithm based on improved yolov7x. Sensors **23**(11), 5037 (2023). https://doi.org/10.3390/s23115037

50. Roß, T., et al.: Beyond rankings: Learning (more) from algorithm validation. Medical Image Analysis **86**, 102765 (2023). https://doi.org/10.1016/j.media.2023.102765
51. Ruiz, N., et al.: Accuracy and reliability of artificial intelligence in surgical decision-making: a literature review. Cureus **17**(10), e95337 (2025). https://doi.org/10.7759/cureus.95337
52. Saadati, S., Hashemi, M., Nezhat, C.: Unsupervised deep learning for semantic segmentation using laparoscopic videos: A self-detection and self-learning approach. Mach. Learn. Appl. **22**, 100767 (2025). https://doi.org/10.1016/j.mlwa.2025.100767
53. Sasaki, K.: Automated Surgical Workflow Identification by Artificial Intelligence in Laparoscopic Hepatectomy: Experimental Research. Int. J. Surg. **105**, 106856 (2022). https://doi.org/10.1016/j.ijsu.2022.106856
54. Sugiyama, T., et al.: Artificial intelligence-integrated video analysis of vessel area changes and instrument motion for microsurgical skill assessment. Scientific Reports **15**, 27898 (2025). https://doi.org/10.1038/s41598-025-27898-z
55. Wang, Y.H., et al.: SMILEtrack: similarity learning for occlusion-aware multiple object tracking. In: Proc. AAAI Conference on Artificial Intelligence, vol. 38, pp. 5740–5748 (2024). https://doi.org/10.1609/aaai.v38i6.28386
56. Wiese, L., et al.: Detection of surgical instruments based on synthetic training data. Computers **14**(2), 69 (2025). https://doi.org/10.3390/computers14020069
57. Wu, Z., et al.: Augmenting efficient real-time surgical instrument segmentation in video with point tracking and Segment Anything. Healthcare Technol. Lett. (2024). https://doi.org/10.1049/htl2.12111
58. Xie, D., et al.: Surgical instruments hyalinization: Occlusion removal in minimally invasive endoscopic surgery. Biomimetic Intell. Robot. **3**(3), 100105 (2023). https://doi.org/10.1016/j.birob.2023.100105
59. Xu, H., et al.: Surgripe challenge: benchmark of surgical robot instrument pose estimation. Med. Image Anal. **105**, 103674 (2025). https://doi.org/10.1016/j.media.2025.103674
60. Xu, Z., et al.: Spine endoscopic atlas: an open-source dataset for surgical instrument segmentation. Scientific Data **12**, 1611 (2025). https://doi.org/10.1038/s41597-025-05897-7
61. Yang, X., et al.: CUE-YOLO: an efficient and robust detector for metal surface defect detection with adaptive attention and task alignment. IEEE Access **14**, 37020–37037 (2026). https://doi.org/10.1109/ACCESS.2026.3670661
62. Zhang, T., Yuan, X., Xu, H.: Surgical Instrument Segmentation via Segment-Then-Classify Framework with Instance-Level Spatiotemporal Consistency Modeling. J. Imaging **11**(10), 364 (2025). https://doi.org/10.3390/jimaging11100364

Deep Learning-Based Crack Detection on Complex Decorative Surfaces of Cultural Heritage Artifacts

HongYi Shih[1(✉)], Debora Verniz[1], Filiz Bunyak[2], and Jessica Boldt[3]

[1] Department of Architectural Studies, University of Missouri, Columbia, USA
{hsgg6,vernizd}@missouri.edu
[2] Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, USA
bunyak@missouri.edu
[3] Museum of Anthropology, University of Missouri, Columbia, USA
boldtj@missouri.edu

Abstract. Artifact condition monitoring plays an important role in heritage preservation, as surface deterioration may lead to irreversible damage. However, detecting fine cracks on decorative artifacts is particularly challenging due to the presence of complex surface patterns that can obscure crack features. In this study, we develop a deep learning-based approach for crack detection on texture-rich cultural heritage artifacts. A dataset of 481 images was collected from four hand-painted terracotta lids, including both cracked and intact samples. We trained a YOLOv8 segmentation model and evaluated its performance under different experimental settings. Specifically, we investigated the effects of several preprocessing strategies, including background removal, specular reflection mitigation, and contrast enhancement, as well as the influence of dataset size, input resolution, and transfer learning. Our results show that reducing visual interference leads to more reliable detection, while increasing the dataset size improves model stability and generalization. We also observed that contrast enhancement changes the visibility of cracks under complex surface conditions but does not consistently improve overall performance. In addition, transfer learning from an infrastructure crack dataset increases precision in some cases, but can reduce recall, likely due to domain differences. Despite these improvements, the model still struggles with highly patterned surfaces, indicating the inherent difficulty of crack detection in decorative artifacts. These findings highlight key factors that affect model performance and provide practical insights for applying deep learning methods in digital heritage preservation.

Keywords: Deep Learning · Cultural Heritage · Crack Detection · Image Processing · Digital Preservation

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 478–491, 2026.
https://doi.org/10.1007/978-3-032-30491-9_30

1 Introduction

Cultural heritage condition monitoring is essential for preservation, as ongoing deterioration can lead to irreversible damage. Early detection and intervention help mitigate further degradation and support long-term conservation. However, traditional condition assessment relies on visual inspection, which is time-consuming and often lacks consistency [1]. Among various forms of deterioration, cracks are one of the most common and critical types, as they directly reflect structural vulnerability and ongoing material degradation. If there is no timely intervention, cracks may propagate and result in more severe damages [2]. However, the fine and irregular features of cracks make detection particularly challenging [3].

These challenges highlight the need for automated and objective detection approaches in heritage preservation. For this purpose, many applications have emerged in recent years and among those, computer vision techniques provide automated, non-invasive approaches for deterioration detection [4], whereas convolutional neural network (CNN)–based deep learning models have become powerful tools for deterioration detection and demonstrated promising performance. For instance, on crack detection tasks, Yiğit and Uysal [5] achieved approximately 83% accuracy in heritage structural inspection. Similarly, Huang et al. [6] developed a two-stage deep learning framework, achieving approximately 3% crack length estimation error.

However, despite these promising results, condition assessment tasks become more challenging when complex decorative patterns are present [7]. This challenge is especially pronounced in cultural heritage artifacts, which often exhibit rich surface decorations that may visually resemble crack-like structures, introducing significant interference for automated detection [8]. Despite this challenge, limited research has systematically analyzed the impact of different factors on crack detection performance in such complex conditions.

This study investigates and evaluates key factors influencing crack detection performance on cultural heritage artifacts with complex decorative surfaces through ablation experiments. A dataset was constructed using artifacts from the Museum of Anthropology, University of Missouri as a case study. Key factors were identified through literature review and were systematically evaluated. Finally, this study provides insights into improving detection performance and outlines directions for future research.

2 Literature Review

This literature review synthesizes key factors in deep learning-based detection tasks and examines how they may affect crack detection under complex surface conditions.

Various sources of visual interference can degrade model performance in detection tasks. For instance, complex backgrounds can introduce irrelevant patterns that lead to false positives and feature confusion [9], making it more difficult to separate target objects from surrounding regions.

Beyond background complexity, variability in lighting conditions can significantly affect crack visibility, as changes in illumination may introduce misleading patterns that

resemble cracks, making detection more challenging [10]. In addition, specular reflections can produce strong highlights and distortions that complicate image interpretation and degrade the performance of detection tasks [11].

Another related factor is the contrast between cracks and their surrounding background material. For example, in pavement crack detection, background texture can interfere with the continuity of cracks and reduce their distinguishability [12]. Similar to concrete, material degradation reduces the contrast between cracks and the background, thereby impairing detection performance [13].

The above-mentioned factors make image preprocessing critical in detection pipelines that enhances relevant visual features and suppresses noise, thereby directly influencing the performance of deep learning models [14, 15].

In addition to image-level factors, data representation also plays a crucial role. In object detection tasks, higher-resolution inputs can improve the representation of small or subtle features, whereas lower resolutions may result in the loss of critical details [16].

Furthermore, transfer learning has been shown to improve model generalization. Zhang et al. [17] demonstrated that low-level features such as edges and textures can be effectively transferred to pavement crack detection tasks using transfer learning, improving the model's ability to distinguish cracks from complex background patterns. Suryawanshi et al. [18] further applied transfer learning to crack detection in ancient wall paintings, but highlighted challenges caused by visually similar features such as brush strokes and textures. These findings suggest that, despite its advantages, transfer learning may still be ineffective in scenarios with complex surface patterns.

Taken together, these studies highlight several factors that may influence crack detection performance under complex surface conditions, including background complexity, lighting conditions, specular reflections, contrast, input resolution, and transfer learning.

3 Methodology

This study has five main steps: data collection, image processing, annotation and dataset split, model training, and evaluation. The proposed pipeline is shown in Fig. 1.

3.1 Data Collection

490 images of four hand-painted terracotta lids (Fig. 2) were collected over three sessions. Images were captured using a turntable from multiple viewing angles, including both the front and reverse surfaces of each lid, under consistent lighting conditions. Three lids contained visible cracks (Fig. 2a-c), while one intact lid was included as a negative sample (Fig. 2d). After removing duplicate and blurred images, 481 images were retained.

3.2 Image Processing

To reduce visual interference and enhance crack visibility, several image preprocessing strategies were investigated in the ablation experiments. These strategies can be

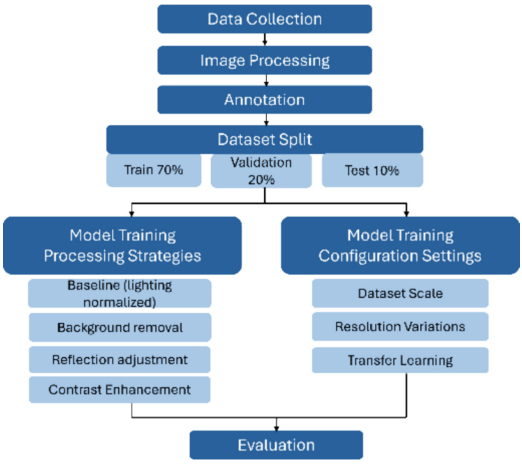


Fig. 1. Proposed crack detection workflow for artifacts with decorative patterns.

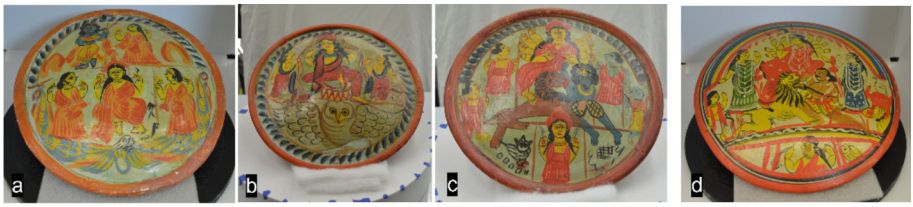


Fig. 2. (a–c) Three cracked samples. (d) One intact sample.

conceptually categorized into two groups: (1) interference reduction, including lighting normalization, background removal, and specular reflection mitigation to minimize noise; and (2) crack feature enhancement, including contrast enhancement and resolution adjustment. Lighting normalization was first applied to reduce illumination variability across acquisition rounds. Background removal was performed using a YOLOv8-based approach to localize the pottery lid region and isolate the artifacts from the background (Fig. 3b). Specular regions were identified based on intensity threshold and subsequently reconstructed using inpainting techniques (Fig. 3c). For crack feature enhancement, contrast enhancement was applied to improve crack visibility while minimizing texture amplification (Fig. 3d), whereas resolution adjustment was used to preserve fine structural details of cracks.

Fig. 3. (a) Raw image. (b) Background removed. (c) Specular reflection adjusted (before and after). (d) Contrast enhanced (before and after).

3.3 Crack Annotation

For the annotation process, Roboflow [19] was used to annotate cracks using the polygon tool. In this project, cracks were operationally defined as thin, elongated linear features exhibiting depth cues, such as subtle shadowing along the fractured path. Due to the difficulty of detecting fine linear cracks with deep learning models, annotated regions were slightly expanded to include surrounding context and improve model learning. The annotated dataset was split into 70% for training, 20% for validation, and 10% for testing.

3.4 Model Training

YOLOv8n model [20] was used for training for its efficiency and suitability for small datasets. A series of controlled experiments were designed to evaluate the impact of different processing strategies and training configurations on crack detection performance. A baseline model was first trained using images with lighting normalization applied to ensure consistent illumination. This was followed by a systematic evaluation of different preprocessing strategies.

The best-performing preprocessing configuration was then selected as the reference condition. Based on this configuration, further experiments were conducted to investigate the effects of dataset scale, input resolution, and transfer learning on model performance. Dataset scale refers to increasing the number of training images under the same preprocessing pipeline to evaluate the effect of training data size. For resolution analysis, the input size was increased from 640 × 640 to 1024 × 1024 to examine its impact on detecting fine crack structures. For transfer learning, a crack segmentation model pretrained on an infrastructure crack dataset [21] was fine-tuned on the terracotta crack dataset to evaluate cross-domain knowledge transfer. The configurations are summarized in Table 1.

Table 1. Summary of ablation experiments.

Configuration	Description
Baseline	200 images with lighting normalization applied.
Background and Reflection	200 images with background removal and reflection adjustment applied.
Dataset Scale	Expanded from 200 to 481 images, with background removal and reflection adjustment applied.
Contrast Enhancement	481 images with background removal, reflection adjustment, and contrast enhancement applied.
Resolution	Input resolution increased from 640 × 640 to 1024 × 1024, with background removal and reflection adjustment applied.
Transfer Learning	A pretrained crack segmentation model (infrastructure dataset) fine-tuned on terracotta images, with background removal and reflection adjustment applied.

(*continued*)

Table 1. (*continued*)

Configuration	Description
Note: Lighting normalization was applied in all configurations. All experiments used an input resolution of 640 × 640, except the resolution test (1024 × 1024).	

3.5 Evaluation

Model performance was evaluated using three approaches: (1) quantitative metrics, including precision, recall, and mean Average Precision at 0.5 Intersection over Union (mAP@0.5); (2) qualitative visualization of detection results, and (3) external images to assess generalization to unseen data. These evaluations were used to identify factors contributing to variations in model performance.

4 Results

4.1 Quantitative Evaluation on the Internal Dataset

The quantitative results of the six ablations are compared, including precision, recall, and mAP@0.5. Precision and recall follow Eqs. (1) and (2).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

True Positive (TP) represents correctly detected crack instances, False Positive (FP) corresponds to background regions incorrectly identified as cracks, and False Negative (FN) denotes crack instances that are not detected and are instead classified as background.

The mAP@0.5 metric evaluates detection performance by computing the area under the precision–recall curve at an Intersection over Union (IoU) threshold of 0.5. Predictions with IoU $\geq$ 0.5 are considered correct. The IoU is defined in Eq. (3).

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{3}$$

The quantitative outcomes of precision, recall, and mAP@0.5 are presented in Table 2. These metrics provide a comprehensive evaluation of detection performance, reflecting the trade-off between precision (false positives), recall (missed detections), and overall localization accuracy (mAP).

Table 2. Quantitative results of ablation experiments.

	Precision	Recall	mAP@0.5
Baseline	0.051	0.046	0.008
Background and reflection	0.138	0.074	0.050
Dataset expansion	0.615	0.408	0.4
Contrast enhancement	0.608	0.369	0.393
Resolution	0.573	0.464	0.461
Transfer learning	0.634	0.458	0.483

The precision-recall (PR) curves in Fig. 4 illustrate the trade-off between precision and recall across all ablation settings. These curves provide an assessment of detection performance across different configurations.

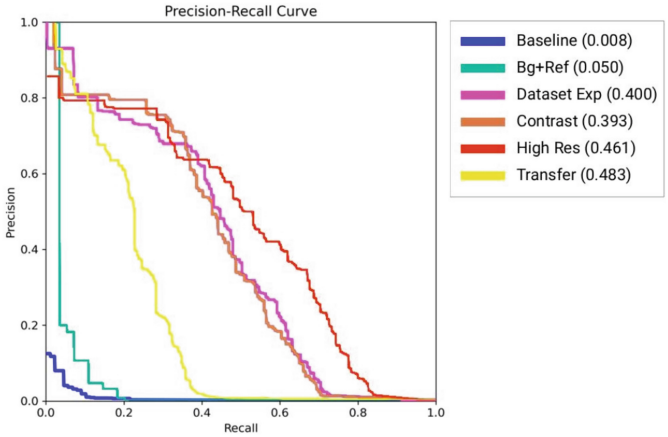
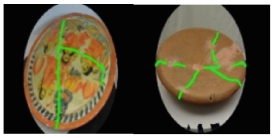
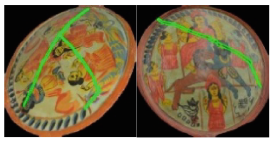
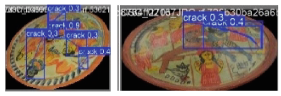
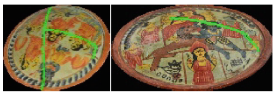


Fig. 4. Precision–recall curves for different configurations.

4.2 Qualitative Evaluation on the Internal Dataset

The qualitative results, including ground truth and detection outputs for the six ablation settings, are presented in Table 3 for comparison. These visual illustrations support quantitative findings and reveal detection behaviors.

Table 3. Qualitative comparison of crack detection results.

	Ground Truth	Crack detection results
Baseline		
Background and reflection		
Dataset expansion		
Contrast enhancement		
Resolution		
Transfer learning		

For the baseline model, few cracks are detected. With precision, recall, and mAP@0.5 all close to zero (0.051, 0.046, and 0.008 respectively), the model exhibits limited ability to learn discriminative features and is unable to distinguish among different visual interferences (specular reflections, cluttered backgrounds, and decorative patterns). The blue PR curve demonstrates that the performance remains close to the origin, reflecting consistent low precision and recall across different confidence thresholds. This deficiency is further explained by the visual results, where cracks are only detected on plain-pattern surfaces (e.g., the reverse side of the artifact), while detection fails on surfaces with complex decorative patterns. These results suggest that the model is highly sensitive to surface complexity and visual noise.

After applying background removal and reflection mitigation, the detection performance improves evidently. Precision, recall, and mAP@0.5 increase from 0.051 to 0.138 (+170.6%), 0.046 to 0.074 (+60.9%), and 0.008 to 0.050 (+525%), respectively. The PR curve (green line) moves slightly away from the origin, indicating a more stable detection behavior across confidence thresholds. The visualized results show that reducing background noise and specular reflections improves the model ‘s ability to localize crack

regions. However, the model still shows difficulty distinguishing cracks from decorative patterns, often producing fragmented or incomplete detections.

After validating the effectiveness of background and reflection removal, the dataset was expanded from 200 to 481 images to enhance data diversity and to improve detection performance. The model shows the detection performance improves substantially. Precision increases from 0.138 to 0.615 (+345.6%), recall from 0.074 to 0.408 (+451.3%), and mAP@0.5 from 0.050 to 0.400 (+700%). The PR curve (pink line) further indicates the performance improves across a wider range of confidence thresholds. The visualized results further depict this improvement. More crack regions are detected, and cracks appear more continuous and better aligned with ground truth annotations. However, missed detections still occur with dataset expansion.

Following dataset expansion and prior preprocessing, contrast enhancement was additionally applied. With contrast enhancement, precision slightly decreases from 0.615 to 0.608 (−1.1%), recall from 0.408 to 0.369 (−9.6%), and mAP@0.5 from 0.400 to 0.393 (−1.8%). The PR curve (orange line) remains similar to the dataset expansion model, indicating improvement in overall detection performance is limited. Qualitative results show some additional local responses in crack-like regions; however, the detections appear to be more fragmented and do not translate to better overall quantitative performance.

Since contrast enhancement did not improve performance, higher resolution (1024 × 1024) was applied based on the dataset expansion setting. With higher resolution, mAP@0.5 increases from 0.400 to 0.461 (+15.3%), and recall improves from 0.408 to 0.464 (+13.7%), while precision slightly decreases from 0.615 to 0.573 (−6.8%). The PR curve (red line) shifts further from the origin, particularly in the higher recall region, indicating detection coverage is improved. Qualitative results show that higher resolution enables better detection of fine crack details and more complete crack structures. However, higher resolution also introduces additional false positives, particularly in highly textured regions, that reveals a trade-off between recall and precision.

Based on the dataset expansion setting, transfer learning was applied using a pretrained infrastructure crack detection model. The transfer learning model achieves the highest precision (0.634) and mAP@0.5 (0.483), while recall (0.458) remains slightly lower than the higher resolution model. However, the PR curve (yellow line) sharply drops beyond the low-recall region, indicating that its performance is concentrated within a limited range of high-confidence detections rather than being sustained across the full recall range. Qualitative results further reveal that detection coverage is limited. Although some crack regions are detected with high confidence and appear to be less fragmented, other crack regions remain entirely undetected.

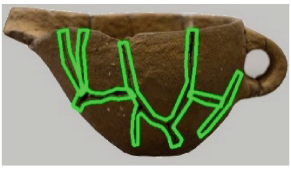
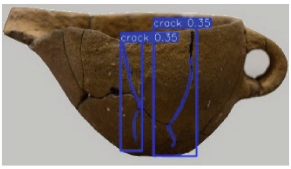
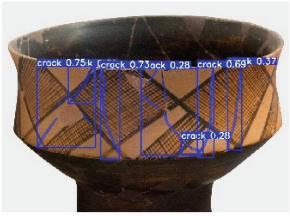
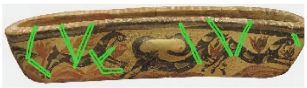
4.3 External Evaluation

For the external evaluation, artifacts with different shapes and levels of decorative complexity were selected to assess the model's generalization to unseen data. The first artifact is a cup with a plain surface and both thick and thin cracks (Fig. 5a). The second artifact is a fruit stand with regular geometric decorations on the surface (Fig. 5b). The third artifact is a kymbe (shallow vessel) with complex painted patterns and fine cracks (Fig. 5c).

Fig. 5. External pottery artifacts: (a) a cup with a plain surface, (b) a fruit stand with geometric decoration, and (c) a kymbe with complex painted patterns. Adapted from [22–24].

The transfer learning model which achieves the best performance (Table 2) was selected to detect the cracks in the external images. The results are qualitatively and quantitatively presented simultaneously for clear review. Visual comparisons between the ground truth and the detection results are provided, along with quantitative metrics of precision and recall, as shown in Table 4.

Table 4. Crack detection results on external artifacts.

Artifact	Ground Truth	Crack detection results	Precision	Recall
Cup			0.932	0.167
Fruit stand			0.927	0.412
Kymbe			0.983	0.080

Among the three sample artifacts, precision is always high (ranging from 0.927 to 0.983), where recall varies but remains low overall (between 0.080 and 0.412). For the cup artifact, the model detects several main cracks, but is unable to detect the thicker cracks, resulting in low recall (0.167) despite high precision (0.932). For the fruit stand artifact, the model is able to distinguish cracks from the geometric patterns with some cracks missing, leading to moderate recall (0.412) and high precision (0.927). For the kymbe artifact, the model fails to distinguish cracks from the complex patterns with only one crack detected. As a result, precision remains high (0.983), while recall is extremely low (0.080).

5 Discussion

5.1 Internal Evaluation

This study examines how different preprocessing strategies and model configurations may affect crack detection performance on artifacts with complex decorative patterns. The results reveal that detection performance is strongly influenced by surface texture and visual interference, and that different strategies introduce clear trade-offs between precision and recall. Based on the above stated observations, several key insights can be drawn as core emphasis.

First, reducing visual interference through background removal and specular reflection mitigation improves detection performance. Those factors often introduce patterns that resemble cracks, particularly on textured artifact surfaces. While this preprocessing leads to noticeable improvement compared to the baseline, it is not sufficient to fully address the challenges introduced by complex decorative patterns.

Second, dataset expansion plays an important role in improving detection performance. Increasing both the number and diversity of training images leads to consistent improvements across precision, recall, and mAP@0.5. This behavior suggests that representative in-domain data is more critical than preprocessing alone for supporting generalization on complex surfaces.

Third, visual enhancement strategies do not always lead to better results on texture-rich artifacts. Although both contrast enhancement and higher resolution increase the visibility of fine details, their effects differ and somewhat contradictive. Contrast enhancement tends to amplify surface texture and noise, which often results in more fragmented detections but not improving overall performance. In contrast, higher resolution improves recall and mAP@0.5 by preserving finer crack details but slightly reduces precision due to an increase in false positives.

Finally, transfer learning appears to be domain sensitive. Although the pretrained infrastructure crack model achieves the highest precision and mAP@0.5, its performance is concentrated in a limited high-confidence region and does not remain stable across the full recall range. Qualitative results also show incomplete crack coverage, suggesting that domain differences between infrastructure surfaces and decorated artifact surfaces restrain effective knowledge transfer.

5.2 External Evaluation

The external evaluation reveals the strengths and limitations of the model's ability to detect cracks across different levels of pattern complexity and crack types. The high precision and low recall scenario indicate accurate detection but a missing of large portion of cracks. This is likely due to visual interference and dataset bias.

First, detection performance is strongly affected by visual interference from surface patterns. This may be related to the structural characteristics of the patterns. The geometric patterns are relatively regular and well-structured, making them easier for the model to distinguish from cracks. In contrast, irregular and densely distributed patterns introduce greater visual interference and share similar features with cracks, making accurate detection more difficult to achieve.

Second, the training data is dominated by thin cracks, leading the model to be biased toward thin crack features and limiting its ability to detect thicker cracks.

It should be noted that although artifacts with different shapes were included in the external evaluation, no clear evidence was found to suggest that shape influences detection performance.

Overall, the results suggest that the model exhibits conservative detection behavior. While precision remains consistently high, recall varies across different conditions, indicating condition-dependent generalization.

6 Conclusion

This study shows that crack detection performance on artifacts with complex decorative patterns is strongly influenced by surface texture and visual interference. Different preprocessing strategies and configurations lead to clear trade-offs among precision, recall, and detection coverage, suggesting that no single approach consistently optimizes all aspects of performance. Considering the visual complexity of heritage artifacts, methods developed for conventional computer vision tasks may not transfer appropriately. Instead, the effectiveness of conventional computer vision methods appears to depend on the characteristics of the target surface and the intended detection goals.

Although the results highlight several important factors that affect crack detection, a number of challenges remain. First, while performance improves under different configurations, the overall detection metrics are still moderate, suggesting that further improvement is necessary. Second, the dataset size is relatively small and includes artifacts with similar decorative patterns. This is further reflected in the external evaluation that the generalization performance decreases under complex visual conditions. Third, the annotated cracks are primarily subtle and thin, which biases the model toward thin crack features and limits its ability to detect cracks with different scales and characteristics.

Future works will explore models that are better suited for capturing fine, crack-specific features, particularly for subtle and elongated structures. In addition, expanding the dataset to include a wider range of decorative patterns and crack variations is essential for improving generalization across diverse artifact surfaces.

Author Contributions. H. Shih designed and performed the experiments and wrote the manuscript. D. Verniz and F. Bunyak supervised the project and reviewed the final version of the manuscript. F. Bunyak provided guidance on image processing and deep learning methods. H. Shih, D. Verniz, and J. Boldt collected the data.

Acknowledgments. We are grateful to the Museum of Anthropology at the University of Missouri-Columbia for providing access to the collections.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Mishra, M., Lourenço, P.B.: Artificial intelligence-assisted visual inspection for cultural heritage: state-of-the-art review. J. Cult. Herit. **66**, 536–550 (2024). https://doi.org/10.1016/j.culher.2024.01.005

2. Wang, H., Shi, Y., Yuan, Q., Li, M.: Crack detection and feature extraction of heritage buildings via point clouds: a case study of Zhonghua Gate Castle in Nanjing. Buildings. **14**, 2278 (2024). https://doi.org/10.3390/buildings14082278
3. Zhang, Y.: Knowledge of earthen heritage deterioration in dry areas of China: salinity effect on the formation of cracked surface crust. Herit. Sci. **11**, 41 (2023). https://doi.org/10.1186/s40494-023-00890-y
4. Carvalho Ottoni, A.L., Cordeiro Ottoni, L.T.: A deep learning approach for cultural heritage building classification using transfer learning and data augmentation. J. Cult. Herit. **74**, 214–224 (2025). https://doi.org/10.1016/j.culher.2025.06.010
5. Yiğit, A.Y., Uysal, M.: Automatic crack detection and structural inspection of cultural heritage buildings using UAV photogrammetry and digital twin technology. J. Build. Eng. **94**, 109952 (2024). https://doi.org/10.1016/j.jobe.2024.109952
6. Huang, W.-C., Luo, Y.-S., Liu, W.-C., Liu, H.-M.: Deep learning-based crack detection on cultural heritage surfaces. Appl. Sci. **15**, 7898 (2025). https://doi.org/10.3390/app15147898
7. Karimi, N., Mishra, M., Lourenço, P.B.: Automated surface crack detection in historical constructions with various materials using deep learning-based YOLO network. Int. J. Archit. Herit. **19**, 581–597 (2025). https://doi.org/10.1080/15583058.2024.2376177
8. Zhou, J., Lu, Y., Zheng, K., Smith, K., Wilder, C., Wang, S.: Design Identification of Curve Patterns on Cultural Heritage Objects: Combining Template Matching and CNN-based Re-Ranking, http://arxiv.org/abs/1805.06862, (2018). https://doi.org/10.48550/arXiv.1805.06862.
9. Aamir, S.M., Ma, H., Khan, M.A.A., Aaqib, M.: Real-Time Object Detection in Occluded Environment with Background Cluttering Effects Using Deep Learning, http://arxiv.org/abs/2401.00986, (2024). https://doi.org/10.48550/arXiv.2401.00986.
10. Padalkar, M.G., Beltrán-González, C., Bustreo, M., Del Bue, A., Murino, V.: A versatile crack inspection portable system based on classifier ensemble and controlled illumination. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 4009–4016 (2021). doi:10.1109/ICPR48806.2021.9412039.
11. Li, G., Liu, Q., Ren, W., Qiao, W., Ma, B., Wan, J.: Automatic recognition and analysis system of asphalt pavement cracks using interleaved low-rank group convolution hybrid deep network and SegNet fusing dense condition random field. Measurement. **170**, 108693 (2021). https://doi.org/10.1016/j.measurement.2020.108693
12. Zou, Q., Zhang, Z., Li, Q., Qi, X., Wang, Q., Wang, S.: DeepCrack: learning hierarchical convolutional features for crack detection. IEEE Trans. Image Process. **28**, 1498–1512 (2019). https://doi.org/10.1109/TIP.2018.2878966
13. Nguyen, T.Q., Phan-Vu, P., Nguyen, P.T.: AI-based damage detection in prestressed concrete beams: a vision-integrated deep learning framework for crack localization and severity classification. Adv. Bridge Eng. **7**, 6 (2026). https://doi.org/10.1186/s43251-025-00194-0
14. Cha, Y., Choi, W., Büyüköztürk, O.: Deep learning-based crack damage detection using convolutional neural networks. Comput. Aided Civ. Inf. Eng. **32**, 361–378 (2017). https://doi.org/10.1111/mice.12263
15. Wei, J.: Based on an overview of crack detection. Sci. J. Technol. **6**, 133–137 (2024). https://doi.org/10.54691/dddp5674
16. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M.: YOLOv4: Optimal Speed and Accuracy of Object Detection, http://arxiv.org/abs/2004.10934, (2020). https://doi.org/10.48550/arXiv.2004.10934.
17. Zhang, K., Cheng, H.D., Zhang, B.: Unified approach to pavement crack and sealed crack detection using Preclassification based on transfer learning. J. Comput. Civ. Eng. **32**, 04018001 (2018). https://doi.org/10.1061/(ASCE)CP.1943-5487.0000736
18. Suryawanshi, V.D., Borse, D.I.S., Patil, D.R.V.: Deep learning-based crack detection in ancient paintings for digital art conservation. Open Access Int. J. Sci. Eng. **8** (2025)

19. Dwyer, B., Nelson, J., Hansen, T.: Roboflow [Software]. https://roboflow.com.
20. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLOv8. https://github.com/ultralytics/ultralytics (2023)
21. OpenSistemas/YOLOv8-crack-seg. https://huggingface.co/OpenSistemas/YOLOv8-crack-seg.
22. Zde: File: Early Cycladic pottery, Mykonos, 2700–2300 BC, AM Mykonos, 224928.jpg. Wikimedia Commons Licensed under CC BY-SA 4.0. (2022). https://commons.wikimedia.org/wiki/File:Early_Cycladic_pottery,_Mykonos,_2700-2300_BC,_AM_Mykonos,_224928.jpg
23. Zde: File: Fruitstand, Neolithic pottery, Franchthi Cave, AM of Nafplio, 202057.jpg. Wikimedia Commons (2020). Licensed under CC BY-SA 4.0. https://commons.wikimedia.org/wiki/File:Fruitstand,_Neolithic_pottery,_Franchthi_Cave,_AM_of_Nafplio,_202057.jpg
24. Zde: File: Pottery kymbe painted ibexes 1700–1600 BC, PMTh 356 0503214.jpg. Wikimedia Commons (2014). Licensed under CC BY-SA 3.0. https://commons.wikimedia.org/wiki/File:Pottery_kymbe_painted_ibexes_1700-1600_BC,_PMTh_356_0503214.jpg

StableZero: Improving General Image Generation Through Visualisation Feedback

Kazim Laos[1], Hector D. Menendez[2(✉)], and Gema Bello-Orgaz[3]

[1] London, UK
[2] King's College London, London, UK
hector.menendez@kcl.ac.uk
[3] Universidad Politecnica de Madrid, Madrid, Spain
gema.borgaz@upm.es

Abstract. Improving image generation systems is essential to reduce compute, energy, and token usage. A key bottleneck is reliance on textual prompts, which are often ambiguous and fail to steer generative models reliably. Prior work, such as StableYOLO, mitigates this via prompt and parameter optimisation with YOLO-derived feedback, but its effectiveness is bounded by YOLO's fixed object taxonomy. This paper introduces StableZero, an enhanced framework that closes this gap by incorporating image captioning into the optimisation loop. Caption-based feedback enables open-vocabulary, semantically grounded search beyond category limits. StableZero couples prompt engineering, semantic fitness scoring, and genetic-algorithm optimisation, building on the StableYOLO architecture. We evaluate on both object-centric and abstract prompts, targeting gains in realism and diversity. Experiments indicate complementary strengths: StableYOLO excels on object-focused prompts, whereas StableZero outperforms on abstract or holistic scenes. Analysis of prompt vocabularies and parameter distributions suggests hybrid strategies can further advance image quality. Overall, captioning-based evaluation offers a more adaptable alternative to category-restricted methods. By aligning feedback with semantics rather than fixed labels, it reduces wasted iterations and improves sample efficiency. StableZero establishes a robust foundation for automated image evaluation and optimisation in creative AI.

Keywords: Generative AI · Text-to-image synthesis · Stable Diffusion · Image captioning · Semantic similarity · Genetic-algorithm optimisation · YOLO · StableYOLO · StableZero

1 Introduction

Generative Artificial Intelligence (AI) refers to computational techniques capable of producing seemingly novel and meaningful content –such as images, text, or

K. Laos—Independent Researcher.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 492–509, 2026.
https://doi.org/10.1007/978-3-032-30491-9_31

audio– based on patterns learned from training data [1]. In the domain of image generation, models such as Stable Diffusion [17], DALL-E, and Midjourney have facilitated the creation of high-quality visuals from textual descriptions, also referred to as prompts [18]. These systems rely on various parameters, including guidance scale, inference steps, and random seed, all of which influence the quality and characteristics of the final output [3].

Improving the quality of generated images remains a central research challenge, particularly in producing photorealistic outputs [2,13]. A major difficulty lies in evaluating image quality, which often depends on subjective human judgment or automated metrics with limited scope, similar to LLMs [5]. Although object-detection models such as YOLO (You Only Look Once) can assist in assessing generated images, their utility is constrained by predefined categories, making it difficult to generalize evaluations for open-ended or highly creative prompts.

A recent approach addressing this issue is StableYOLO, which integrates Stable Diffusion with YOLO and leverages a genetic algorithm to optimize image quality [2]. In this work, we extend this framework by proposing a more flexible mechanism that is not bound by YOLO's restricted classification vocabulary and can instead provide holistic improvements across diverse photorealistic images.

The improved system, named StableZero, introduces an alternative feedback loop based on image captioning. By interpreting visual content in natural language, the system can guide the optimization process in a broader and semantically meaningful way. Its goal is to enhance image quality even when the underlying concepts fall outside YOLO's detection range. To achieve this, StableZero employs captioning –or image-description models– to compare the semantic alignment between the original prompt and the generated caption, using this similarity as the fitness function that drives the optimization.

We validate both StableZero and StableYOLO, finding that they significantly outperform their respective baselines (Wilcoxon $p < 0.01$), with StableYOLO excelling on object-centric prompts and StableZero performing better on more abstract or holistic scenes. Across methods, prompt-word and hyperparameter analyses reveal shared drivers of image quality –such as the importance of *photograph*, *depth*, *color*, and *field*– and similar optimal parameter ranges for inference steps (∼56–57), guidance scale (∼10–12), and guidance rescale (∼0.36–0.47). These observations suggest strong complementarity between the two systems and motivate the exploration of hybrid strategies. Our contributions are as follows:

- An evaluation across 20 prompts with multiple generations, showing that StableZero improves quality by +16.8 points over its baseline (0.459 → 0.627) and that StableYOLO achieves an even larger absolute gain (0.4497 → 0.781).
- A lexical analysis identifying shared positive (*photograph*, *depth*, *field*, *color*) and negative (*illustration*, *art*, *error*, *painting*) prompt terms associated with higher image quality.

- A comparative study demonstrating no significant differences in the distributions of key parameters (inference steps, guidance scale, guidance rescale), despite the models relying on different quality metrics.
- A qualitative characterization of the contexts in which each technique performs best, highlighting complementary strengths and motivating their potential combination for further improvements.

2 Background

StableZero builds on several areas of research, including generative AI for image synthesis, visual analysis and captioning systems, and semantic similarity metrics. This section reviews the current state and limitations of generative image models, with a particular focus on evaluating and improving image quality.

2.1 Generative AI for Image Generation

Generative AI has advanced rapidly, particularly in text-to-image generation. These systems use natural-language prompts to synthesise images, transforming how users interact with creative tools. Models such as DALL-E 2, Stable Diffusion, and DeepMind's Imagen [8] rely on transformer-based architectures trained on large-scale datasets containing millions of image-text pairs [3].

Latent Diffusion Models (LDMs), which operate in compressed latent spaces, have emerged as the dominant architecture for high-quality image generation [16]. Although Generative Adversarial Networks (GANs) once played a central role, diffusion models have largely replaced them due to improved stability and output quality [4]. Another relevant methodology is CLIP-based guidance, which aligns image and text semantics through contrastive learning [11].

StableZero builds on Stable Diffusion [19], an LDM-based model, and introduces a feedback loop that leverages caption analysis to optimise the generation process.

2.2 Visualisation Systems and Captioning Approaches

Traditional visualisation systems such as YOLO are designed for object detection and classification [9]. YOLO has been widely adopted for evaluating generated images because it provides structured numerical feedback. However, its reliance on predefined object categories and photorealistic training data limits its usefulness for diverse or creative generative tasks [2].

To overcome these limitations, this work examines captioning models. Models such as BLIP, CLIP [7], and LLaMA-VID [12] can generate descriptive natural-language captions for images, offering a richer and more contextual understanding of visual content. Unlike YOLO, these systems do not inherently produce numerical scores but a description. StableZero transforms description comparisons into quantitative measures using semantic similarity metrics, constructing a fitness function for optimisation (see Sect. 3).

2.3 Caption Comparison Metrics

Unlike YOLO, which provides direct numerical detection scores based on visible objects, caption-based systems generate free-form text and therefore do not yield an immediate optimisation signal. This requires deriving quantitative evaluations from comparisons between captions. To accomplish this, established metrics from automatic summarisation can be used. Metrics such as BLEU, ROUGE, METEOR, BERTScore, and SummaC have been extensively studied [14], with ROUGE being one of the most widely used. However, ROUGE lacks semantic understanding, making it insufficient for evaluating meaningfully aligned captions.

For this work, SimCSE (Simple Contrastive Learning of Sentence Embeddings) is employed as the primary similarity metric [6]. SimCSE enables robust semantic comparisons between captions and provides a reliable basis for the fitness function in the StableZero optimisation framework.

3 StableZero

This section presents StableZero, a photorealistic visual evaluation system extending StableYOLO [2]. The pipeline integrates prompt engineering, semantic evaluation, and evolutionary optimisation. Its goal is to address classical limitations of object-detection-based feedback –and its particularly poor generalisation to unseen categories and visually complex backgrounds– by coupling a generative image model with a caption-driven semantic fitness function.

The workflow consists of four components: (i) generation of synthetic yet realistic images using Stable Diffusion, (ii) the definition of positive/negative prompt modifier spaces and parameter configurations, (iii) a caption-based fitness function for semantic alignment, and (iv) evolutionary search to optimise prompt and parameter selections. This design enables controlled experimentation while increasing both the variability and quality of generated outputs. Figure 1 illustrates the full pipeline.

Unlike approaches relying on static datasets, StableZero employs a generative pipeline in which image creation is driven by user-defined prompts augmented with structured visual descriptors (positive modifiers), parameters, and constraints (negative modifiers). This enables fine-grained control over properties such as lighting, optics, and visual style.

3.1 Prompt Engineering and Search Space

The search space consists of two types of prompt modifiers –positive and negative– and a set of numerical parameters that influence the quality of images generated by Stable Diffusion. StableZero explores this joint space. It retains the original search space of StableYOLO, but extends the vocabulary of positive and negative modifiers. To define the search space, let Σ^* denote the set of

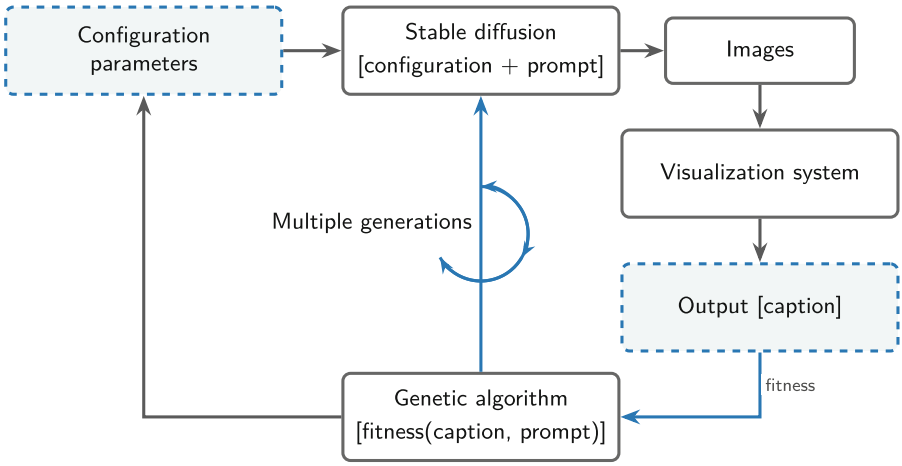


Fig. 1. StableZeropipeline running the four steps to image generation and parameter optimisation.

finite character strings. The user provides a base prompt $p \in \Sigma^*$. We define two ordered modifier sets:

$$\mathcal{P} = (P_1, \ldots, P_m), \qquad \mathcal{N} = (N_1, \ldots, N_n),$$

where $\mathcal{P}$ contains positive modifiers (e.g., lens, film stock, lighting) and $\mathcal{N}$ contains negative modifiers (e.g., "low quality", "blurry").

Binary Subset Selection and Integer Encoding. To keep the search process within a numerical space, a subset of modifiers is selected via the binary expansion of an integer. For a set $\mathcal{S} = (S_1, \ldots, S_\ell)$ and an integer $z \in \{0, \ldots, 2^\ell - 1\}$, let

$$\phi(z, \ell) \in \{0, 1\}^\ell$$

be the ℓ-bit binary vector of z (least/most significant bit convention fixed a priori). The selected subset is

$$\mathrm{Sel}(z; \mathcal{S}) \;=\; \{\, S_i \;\mid\; [\phi(z, \ell)]_i = 1 \,\}.$$

Out-of-range integers are reduced modulo 2^ℓ. This binary number represents a unique selection of words from $\mathcal{P}$ and $\mathcal{N}$.

Prompt Realisation. Given two integers $z^+ \in \{0, \ldots, 2^m - 1\}$ and $z^- \in \{0, \ldots, 2^n - 1\}$, the realised positive and negative prompts are

$$\mathrm{Pos}(p, z^+) \;=\; p \;\|\; \text{“, ”} \;\|\; \Big\|_{S \in \mathrm{Sel}(z^+; \mathcal{P})} S,$$

$$\mathrm{Neg}(z^-) \;=\; \Big\|_{S \in \mathrm{Sel}(z^-; \mathcal{N})} S,$$

where $\|$ denotes string concatenation with commas.

Configuration (phenotype) Space. An individual (phenotype) is the 6-tuple

$$x \;=\; (s,\, g,\, z^{-},\, z^{+},\, r,\, \xi), \quad \text{with}$$

$$s \in \mathbb{N} \quad \text{(inference steps)}, \qquad g \in \mathbb{R} \quad \text{(guidance scale)}, \qquad r \in \mathbb{R} \quad \text{(rescale)},$$

$$z^{-} \in \{0, \ldots, 2^{n} - 1\}, \quad z^{+} \in \{0, \ldots, 2^{m} - 1\}, \quad \xi \in \mathbb{N} \quad \text{(random seed)}.$$

Box constraints (lower/upper bounds) for (s, g, r) are fixed prior to optimisation, and decoding clips values to these bounds.

3.2 Fitness Function and Semantic Evaluation

Let SD denote Stable Diffusion. For a given phenotype $x = (s, g, z^{-}, z^{+}, r, \xi)$ and a base prompt $p \in \Sigma^{*}$, StableZero produces a batch of M images:

$$\mathcal{I}(x; p) \;=\; \mathrm{SD}\big(\,\mathrm{Pos}(p, z^{+}), \mathrm{Neg}(z^{-});\, s, g, r, \xi, M\,\big) \;=\; \{I_k\}_{k=1}^{M},$$

where the seed ξ determinises the generator.

Let $\mathrm{Cap} : \mathcal{I} \to \Sigma^{*}$ be the captioner [10], and let $f : \Sigma^{*} \to \mathbb{R}^{d}$ be the sentence encoder [6]. In the sentence space, we will employ the cosine similarity,

$$\cos(u, v) \;=\; \frac{\langle u, v \rangle}{\|u\|_2 \, \|v\|_2}.$$

Let $I_k \in \mathcal{I}(x; p)$, the *reference text* is the original base prompt p, and the *generated caption* is $c_k = \mathrm{Cap}(I_k)$. Thus, the per-image semantic score is

$$\varphi_k(x; p) \;=\; \cos\big(f(p),\, f(c_k)\big).$$

The overall fitness is the mean similarity over the batch:

$$F(x; p) \;=\; \frac{1}{M} \sum_{k=1}^{M} \varphi_k(x; p). \tag{1}$$

This replaces YOLO's confidence-based scores with a semantic objective reflecting how well the generated images match the intended prompt.

3.3 Genetic Encoding and Operators

We distinguish the *phenotype* x evaluated by the pipeline and a fixed-length *genotype* $b \in \{0, 1\}^{L}$ manipulated by genetic operators.

Genotype/Phenotype Mapping. The genotype is partitioned as

$$b \;=\; b^{(s)} \,\|\, b^{(g)} \,\|\, b^{(z^-)} \,\|\, b^{(z^+)} \,\|\, b^{(r)} \,\|\, b^{(\xi)},$$

where each block encodes one parameter. Integer fields use binary encoding; real fields use uniform quantisation within fixed bounds:

- $b^{(s)}$ encodes $s \in [s_{\min}, s_{\max}] \cap \mathbb{N}$;
- $b^{(g)}$ encodes $g \in [g_{\min}, g_{\max}]$ with quantisation step Δ_g;
- $b^{(z^-)} \in \{0,1\}^n$ encodes $z^- \in \{0, \ldots, 2^n - 1\}$ via binary;
- $b^{(z^+)} \in \{0,1\}^m$ encodes $z^+ \in \{0, \ldots, 2^m - 1\}$ via binary;
- $b^{(r)}$ encodes $r \in [r_{\min}, r_{\max}]$ with quantisation step Δ_r;
- $b^{(\xi)}$ encodes $\xi \in [0, \xi_{\max}] \cap \mathbb{N}$.

The decoder $D : \{0,1\}^L \to \mathcal{X}$ maps b to x by (i) interpreting integer blocks in binary and (ii) mapping real-valued blocks through uniform quantisation and clamping to the specified bounds. The encoder $E : \mathcal{X} \to \{0,1\}^L$ is defined accordingly.

Two-Point Crossover (Vector-Based). Given two parent genotypes $b^{(1)}, b^{(2)} \in \{0,1\}^L$, choose indices $1 \le c_1 < c_2 \le L$ uniformly at random. The offspring genotype $b^{(\text{off})}$ is formed by exchanging the bit segment between c_1 and c_2:

$$b^{(\text{off})}_{1:c_1-1} = b^{(1)}_{1:c_1-1}, \quad b^{(\text{off})}_{c_1:c_2} = b^{(2)}_{c_1:c_2}, \quad b^{(\text{off})}_{c_2+1:L} = b^{(1)}_{c_2+1:L},$$

with the complementary offspring obtained by swapping the parental roles. Because the genotype concatenates all fields, crossover is *vector-based* at the bit level and can cut within or between fields.

Bit-Flip Mutation. Given a genotype $b \in \{0,1\}^L$, the mutated genotype $\tilde{b}$ is obtained by independent Bernoulli flips at rate $\mu \in (0,1)$:

$$\mathbb{P}\big(\tilde{b}_\ell \neq b_\ell\big) = \mu \quad \text{for each } \ell \in \{1, \ldots, L\}.$$

On the phenotypic prompt selectors, bit flips toggle membership of individual modifiers. On real/integer parameters, bit flips change the quantised value after decoding; decoding D ensures all values respect bounds.

Evolutionary Loop. Let $\mathcal{P}_t$ be the population of genotypes at generation t. The algorithm iterates:

1. Decode $b \in \mathcal{P}_t$ to $x = D(b)$, evaluate $F(x; p)$ via (1).
2. Select parents (e.g., tournament or proportionate selection; choice fixed prior to runs).
3. Apply two-point crossover with probability p_c and bit-flip mutation with rate μ to produce offspring.
4. Form the next population $\mathcal{P}_{t+1}$ using a replacement strategy (e.g., $(\mu+\lambda)$ or generational).

Only the variation operators (two-point crossover and bit-flip mutation) are used; all other evolutionary components (selection, replacement) are standard and fixed by the experimental setup.

4 Experimental Setup

To evaluate the effectiveness of StableZero, the primary objective is to determine whether it can handle cases that are typically challenging for StableYOLO. These cases involve concepts outside YOLO's predefined taxonomy. The evaluation therefore employs a diverse set of prompts combining unfamiliar objects, complex backgrounds, and abstract concepts, enabling a comparison of the complementary strengths of both techniques. The study aims to answer the following research questions:

RQ1: Does StableZero improve image quality compared to the baseline? To address this question, the study applies StableZero to 20 prompts and evaluates whether its generated results are statistically better than those of the baseline. The baseline is formed by the images generated with the prompts with no extra words. Both the quality of StableZero and the baseline are measure with the caption quality metric.

RQ2: How does StableZero's improvement compare to the optimisation achieved by StableYOLO? Here, the same methodology as in RQ1 is applied, but with results compared against StableYOLO to examine whether its improvements follow a similar trend. This includes quantifying both image quality and the degree of optimisation achieved by each system. The baseline quality in this context is measured with the detection quality of YOLO.

RQ3: Which prompt words improve image quality for both StableZero and StableYOLO? This question analyses the top 10 images for each prompt and identifies the most frequent words appearing in successful generations. Aggregating these across all 20 prompts highlights the most influential positive and negative prompt terms.

RQ4: Which parameter selections best improve image quality for both StableZero and StableYOLO? Following the same approach as RQ3, this analysis focuses on Stable Diffusion parameters produced after optimisation, aiming to identify which parameter ranges most consistently yield high-quality results.

4.1 Experimental Protocol, Parameters, and Resources

We evaluated the system using a curated set of 20 complex prompts, more detailed than those used in StableYOLO, and executed all experiments on remote GPU servers to enable large-scale generation and real-time performance monitoring. Although no human data were collected, the use of AI-generated imagery and pre-trained models required institutional ethical assessment, and formal approval was obtained from Solent Southampton University, where the

first author developed the work. All models are publicly available under permissive open licences, and all prompts and captions are synthetic and ensuring GDPR compliance.

The genetic algorithm parameters follow those of the original StableYOLO work to ensure a fair comparison. The number of generations is set to 50. Mutation and crossover probabilities are both 0.2. Elitism is 10, with $\mu = 5$ and $\lambda = 5$. The inner mutation probability is also 0.2. The population size is 25, and tournament selection uses a tournament size of 5.

All experiments were executed on a high-performance computing environment consisting of an AMD EPYC Milan 32-core CPU, 723 GB of RAM, 400 GB SSD storage, and two NVIDIA L40S GPUs with 48 GB VRAM each, running Ubuntu 22.04. This configuration provided sufficient compute capacity for the StableZero pipeline, with GPU resources being particularly crucial for the generative processes.

4.2 List of Prompts Used

Table 1 lists the prompts used for evaluation. The prompts were selected to provide a diverse assessment set covering a wide range of contexts. They include realistic scenarios (e.g., "Older woman baking in a cozy kitchen"), historical figures (e.g., "Abraham Lincoln writing a letter"), and fictional or imaginative scenes (e.g., "Darth Vader in a field of flowers", "Frankenstein's Halloween coffee"). The set also spans varied object categories –humans, animals, and inanimate objects –to assess how YOLO handles typical and unconventional imagery. This diversity ensures that the evaluation does not focus solely on standard cases, but also challenges both systems with uncommon and creative situations.

We extend the prompts with two bag-of-words lists. The first is a positive bag containing 60 candidate augmenters for the positive prompt (e.g., "photograph", "digital", "color", "ultra-real", "film grain"). The second is a negative bag containing 26 candidate terms (e.g., "illustration", "painting", "drawing", "art", "sketch", "low-res"). This design doubles the number of options in the original StableYOLO setup and subsumes all of its original bags-of-words [2].

4.3 List of Models Used

The models used in this study include the Stable Diffusion v2 hosted on Hugging Face, developed by Rombach and Esser [16]. This diffusion-based text-to-image generator is designed primarily for English prompts and released under the CreativeML Open RAIL++-M License. As a Latent Diffusion Model, it operates in a compressed latent space with a fixed OpenCLIP-ViT/H text encoder.

For image captioning, we employ BLIP (Bootstrapping Language–Image Pre-training) [10], which generates natural-language descriptions of images and is well suited for multimodal understanding.

Semantic similarity is computed using SimCSE (Simple Contrastive Learning of Sentence Embeddings) [6], based on BERT-base. SimCSE produces robust

Table 1. List of Prompts for the Evaluation.

ID	Prompt	YOLO classes
0	Darth Vader in a Flower Field	None
1	Elderly Woman Baking in Cozy Kitchen	Person/Dining table
2	Handsome Man in Beachwear with Photo on iPhone 14	Person
3	Electrical Engineer Inspecting Control Panel	Person
4	Abraham Lincoln Writing Letter	Person
5	Woman Standing at a Wedding Cake Table	Cake/Person
6	Cheerful Man in Goofy Hat	Person
7	Happy Man in Flower Field	Person
8	Circus Cat on Telephone Pole	Cat
9	Asian Man Sleeping on Couch	Person
10	Sunset in a Study Prompt	Chair/Table
11	Chinese Woman in Relaxing Yard Scene	Person
12	Aging Jack Russell Terrier Napping	Dog
13	Donkey Kong as a Chimpanzee	None
14	Frankenstein's Halloween Coffee	Cup
15	Putin in Dung	Person
16	Dog Using Laptop	Dog/Laptop
17	Mouse Holding Computer Mouse	Mouse
18	Security Guard and Woman	Person
19	Mafia Couple Ruling Hell	Person

sentence embeddings that enable reliable semantic comparisons between textual descriptions.

Finally, we use Ultralytics' YOLOv8 model for StableYOLO, released on 10 January 2023 [15], which provides state-of-the-art performance across a wide range of object detection tasks.

5 Results

This evaluation compares the performance of StableZero and StableYOLO. The experimental evaluations show that both systems significantly outperform their respective baselines, with each method excelling under different conditions: StableYOLO performs best on object-focused prompts (Sect. 5.2), whereas StableZero achieves higher quality on abstract or holistic prompts (Sect. 5.1). Analyses of prompt vocabulary (Sect. 5.3) and parameter distributions (Sect. 5.4) further reveal shared factors influencing image quality, suggesting that both approaches operate under similar optimal settings. Overall, these findings highlight the complementary nature of the two systems and indicate that combining

their strengths could lead to further improvements in image generation quality (Sect. 5.5).

5.1 RQ1: StableZero Quality

The first experiment evaluates whether StableZero improves image quality compared to its baseline. Using the full set of 20 prompts, each repeated four times over multiple generations, the results (Fig. 2) show that StableZero outperforms the baseline in every case. On average, the improvement is 16.8 points, demonstrating the effectiveness of the approach.

The mean quality score achieved by StableZero is 0.627 ± 0.193, compared to 0.459 ± 0.187 for the baseline. A Wilcoxon test confirms that the improvement is statistically significant, with a p-value of $1.91 \cdot 10^{-6}$, well below the 0.01 threshold used to reject the null hypothesis.

We also examined which prompts were easiest or most challenging for the system. StableZero reached its highest scores on prompts 0 (0.95) and 5 (0.84), both of which involve relatively simple semantic descriptions ("Darth Vader in a Flower Field" and "Woman Standing at a Wedding Cake Table"). These results suggest that BLIP, the captioning model, incorporates cultural knowledge enabling it to describe iconic characters such as Darth Vader. In contrast, prompts 19 (0.22) and 4 (0.24) yielded the lowest scores. These prompts ("Mafia Couple Ruling Hell" and "Abraham Lincoln Writing Letter") focus heavily on specific individuals and rely on contextual or symbolic cues that are harder for the model to interpret.

RQ1 Answer. StableZero improves image quality across all 20 prompts, achieving an average gain of 16.8 points over the baseline.

5.2 RQ2: Comparison Between StableZero and StableYOLO

For StableYOLO, the improvement over its baseline is even greater, partly because its metric focuses directly on image quality and allows more fine-grained optimisation (see Fig. 2). The baseline score is 0.450 ± 0.160, whereas StableYOLO achieves 0.781 ± 0.123. As with StableZero, these improvements are statistically significant according to the Wilcoxon test ($p = 1.91 \cdot 10^{-6}$).

StableYOLO reaches its best performance on prompts 3 (0.90) and 2 (0.89), both dominated by person-related elements (see Table 1). In contrast, its lowest scores appear for prompts 0 (0.42) and 1 (0.61), in which the first provides no viable optimisation path and the second contains multiple competing objects.

A comparison of the images with the highest improvements reveals that StableYOLO and StableZero frequently complement each other. For prompts such as 0, 1, and 5, StableYOLO delivers substantially better results, whereas in other cases StableZero performs more effectively. This pattern arises because StableYOLO excels when there are recognisable objects in the image

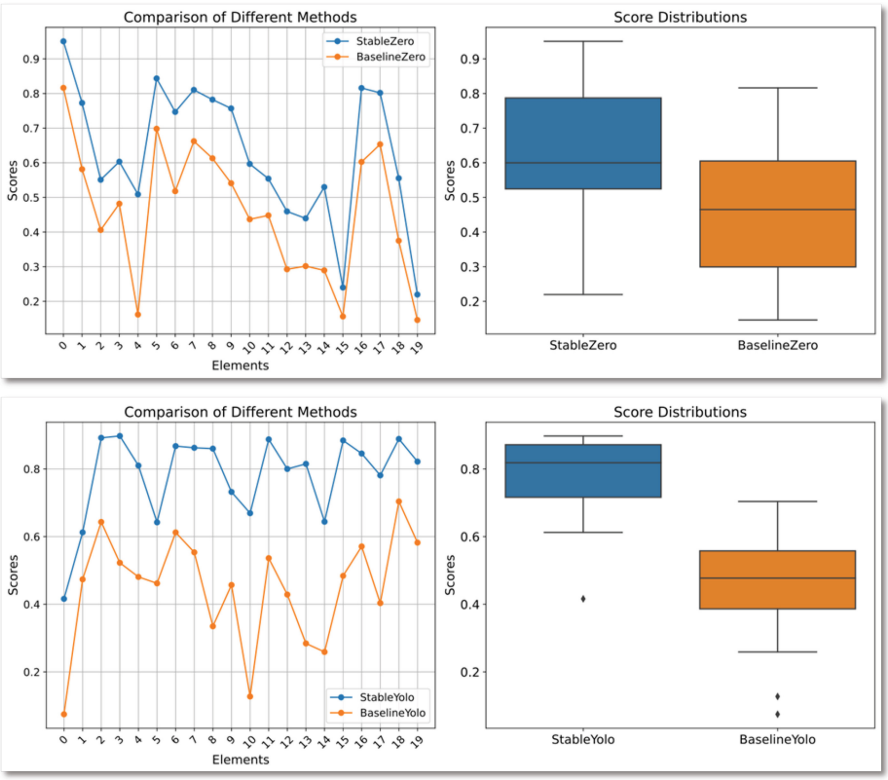


Fig. 2. Comparison between StableZero and its baseline (top), and StableYOLO and its baseline (bottom).

that guide the optimisation process, while StableZero performs better when the scene lacks strong object-level cues. In both situations, the techniques outperform their baselines consistently.

Several prompts –particularly 4 and 19, and to a lesser extent 12 and 13– prove difficult for StableZero, which struggles to achieve meaningful improvement. In these cases, StableYOLO generates noticeably higher-quality outputs. In contrast, those prompts where StableYOLO was able to identify no classes, like Dark Vader's prompt, make it hard to optimize the image quality following StableYOLO's fitness.

Both systems achieve statistically significant improvements over their baselines ($p < 0.01$). This is also visually evident in the result distributions, where a consistent performance gap appears in every run.

RQ2 Answer. StableZero and StableYOLO show complementary strengths. While StableYOLO achieves larger overall improvements, it performs poorly on prompts where StableZero can still enhance quality; conversely, StableYOLO benefits from a clearer optimisation gradient when identifiable objects are present.

5.3 RQ3: Best Words Associated with Image Quality

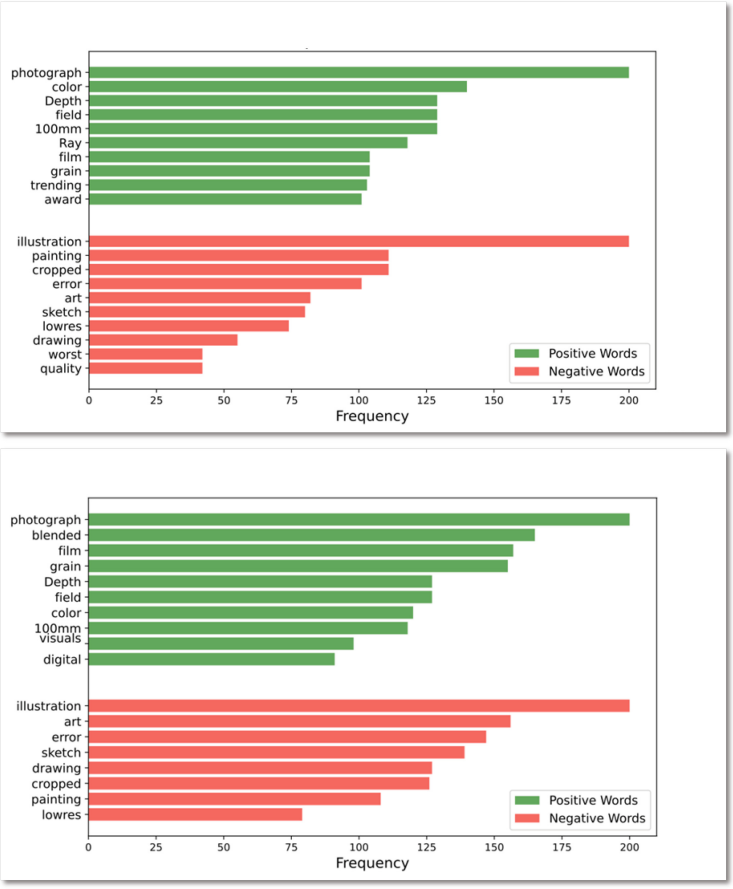


Fig. 3. Most frequent words among the top 10 images for all 20 prompts using StableZero (top) and StableYOLO (bottom).

Many of the most frequent positive-prompt words are shared between the two techniques (see Fig. 3). The word *photograph* appears most often for both. For

StableZero, the next most frequent words are *color*, *depth*, *field*, and *100 mm*, each appearing more than 125 times. For StableYOLO, the leading terms include *blended*, *field*, and *grain* (over 150 occurrences), as well as *depth* and *field* (over 125 occurrences). This pattern indicates that *depth* and *field* are important across both models, while *blended* is specific to StableYOLO.

Other differences include *film*, which is less important for StableZero but remains relevant for StableYOLO. Unique words also appear: StableYOLO favours *visual* and *digital*, whereas StableZero employs *trending* and *award*.

For negative prompts, both models use *illustration* most frequently. StableYOLO exhibits a more even distribution across *art*, *error*, *sketch*, *drawing*, and *crop*. In contrast, StableZero's negative-prompt distribution is more skewed, with a sharp dominance of *illustration*. Some words such as *painting* and *low resolution* appear more often in StableZero, while *quality* and *worst* appear exclusively there, likely reflecting differences in search behaviour rather than aesthetic preferences.

Prompt-length analysis (Table 2) shows that StableZero tends to use longer positive prompts (20.29 words on average) than StableYOLO (17.17 words). For negative prompts the pattern reverses: StableZero uses fewer words (4.49 vs. 5.41). In both cases, Wilcoxon tests confirm statistically significant differences ($p < 0.01$).

Table 2. Average prompt length for StableZero and StableYOLO, with significant differences in both and negative prompts ($p < 0.01$). Parameter comparison between StableZero and StableYOLO (no significant differences according to the Wilcoxon test).

Measure	StableZero	StableYolo	p-value
Average words (positive prompt)	20.29	17.17	7.0×10^{-6}
Average words (negative prompt)	4.49	5.41	2.1×10^{-6}
Inference steps (*mean* $\pm$ SD)	56.77 ± 23.83	56.44 ± 27.24	0.87
Guidance scale (*mean* $\pm$ SD)	10.07 ± 4.80	11.68 ± 5.81	0.43
Guidance rescale (*mean* $\pm$ SD)	0.47 ± 0.27	0.36 ± 0.24	0.11

RQ3 Answer. Overall, both systems prioritise similar positive prompt words—*photograph*, *depth*, *color*, and *field* –and similar negative words– *illustration*, *art*, *error*, and *painting*.

5.4 RQ4: Parameter Selection and Image Quality

RQ4 examines how different Stable Diffusion parameters affect image quality. A notable finding is that both StableZero and StableYOLO converge to very

similar parameter distributions, despite using different evaluation criteria. The three parameters studied are the number of inference steps, the guidance scale, and the guidance rescale (see Table 2).

For inference steps, StableZero uses 57.77 ± 23.83 on average, while StableYOLO uses 56.44 ± 27.24—nearly identical values, though StableYOLO shows more variability. A Wilcoxon test indicates no significant difference ($p = 0.87$).

The guidance scale also proves similar across models: 10.07 ± 4.80 for StableZero and 11.68 ± 5.81 for StableYOLO ($p = 0.43$). Likewise, guidance rescale exhibits no significant difference: 0.47 ± 0.27 vs. 0.36 ± 0.24 ($p = 0.11$).

These results show that both techniques operate under nearly identical optimal parameter settings. Despite different optimisation heuristics and fitness functions, both approaches gravitate toward similar stable regions of the parameter space, suggesting potential for cross-method improvement.

RQ4 Answer. Both systems converge to near-identical parameter ranges: ∼[56,57] inference steps, ∼[10,11] guidance scale, and ∼[0.36,0.47] guidance rescale.

5.5 Discussions

The highest-performing StableYOLO prompts (Fig. 4), such as "Electrical Engineer Inspecting Control Panel" and "Chinese Woman in Relaxing Yard Scene," succeed because they provide strong object-level cues that guide the optimisation. In contrast, StableZero produces more holistic images by focusing on the entire scene.

Weaker StableYOLO prompts –such as "Darth Vader in a Flower Field" and "Sunset in a Study Prompt"– perform poorly when multiple objects interact or when object detection fails. In these scenarios, StableZero produces more coherent and detailed images.

Similarly, StableZero works best on prompts like "Elderly Woman Baking in Cozy Kitchen" and "Woman Standing at a Wedding Cake Table," where global scene coherence is crucial. Meanwhile, StableYOLO tends to emphasise recognisable objects.

Weaker StableZero prompts –such as "Abraham Lincoln Writing Letter" and "Mafia Couple Ruling Hell" – are difficult due to their abstract or symbolic nature, resulting in unrealistic or blurry outputs. Here, StableYOLO performs somewhat better by grounding its optimisation in detectable objects (Fig. 5).

Fig. 4. Comparison of StableYOLO's best and worst prompts (above) with corresponding StableZero outputs (below).

Fig. 5. Best and worst prompts from StableZero (below) with corresponding StableYOLO outputs (above).

6 Conclusions and Future Work

This work presents StableZero, a substantial extension of StableYOLO that replaces YOLO's restricted category-based feedback with a caption-driven evaluation mechanism. StableZero proved effective in producing more flexible assessments and improving image quality in tasks where YOLO encountered limitations. The evaluation shows that StableZero and StableYOLO offer complementary strengths: StableYOLO performs best on object-focused prompts, while StableZero excels on abstract or open-ended scenes. This suggests that combining both approaches could yield a more robust solution than either system alone.

Several avenues for future work emerge from this evaluation. More advanced captioning models could be explored to reduce noise in the feedback loop, and more efficient optimisation strategies may help mitigate the computational cost associated with genetic algorithms. Hybrid systems that integrate both captioning and object detection could also provide a stronger balance between precision and flexibility.

Additional opportunities for extending StableZero include testing it on larger and more diverse datasets, incorporating caption-based evaluation directly into model training rather than solely during inference, and investigating human-in-the-loop methods that combine automated scoring with user preference signals. These extensions would not only strengthen the technical performance of the system but also better align it with human perceptions of creativity and quality.

In conclusion, StableZero demonstrates that caption-driven evaluation is a promising direction for generative image AI. While further refinement is still needed, this work lays a solid foundation for developing more adaptable, scalable, and human-aligned evaluation methods in the future.

Acknowledgments. The support of the UKRI Trustworthy Autonomous Systems Hub (reference EP/V00784X/1), Trustworthy Autonomous Systems Node in Verifiability (reference EP/V026801/2) and the Comunidad Autónoma de Madrid under ALENTAR-J-CM project (reference TEC-2024/COM-224) is gratefully acknowledged.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Archana Balkrishna, Y.: An analysis on the use of image design with generative AI technologies. Int. Trend Sci. Res. Devopment **8**(1), 596–599 (2024). https://www.ijtsrd.com/papers/ijtsrd63468.pdf
2. Berger, H., et al.: Stableyolo: optimizing image generation for large language models. In: Arcaini, P., Yue, T., Fredericks, E.M. (eds.) SSBSE 2023. LNCS, vol. 14415, pp. 133–139. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-48796-5_10
3. Bolya, D., Hoffman, J.: Token merging for fast stable diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4599–4603 (2023). https://doi.org/10.1109/CVPRW59228.2023.00484
4. Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., Bharath, A.A.: Generative adversarial networks: an overview. IEEE Signal Process. Mag. **35**(1), 53–65 (2018). https://doi.org/10.1016/j.jjimei.2020.100004
5. Del Barrio, M., Laos, K., Lara, M.J.V., García, C.G., Menendez, H.D.: Size doesn't matter: Assessing the trustworthiness of large language models in medical contexts: a focus on epidural information retrieval. Artif. Intell. Med. 103379 (2026). https://doi.org/10.1016/j.artmed.2026.103379

6. Gao, T., Yao, X., Chen, D.: SimCSE: simple contrastive learning of sentence embeddings. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 6894–6910 (2021). https://doi.org/10.18653/v1/2021.emnlp-main.552
7. Hafner, M., et al.: Clip and complementary methods. Nature Rev. Meth. Primers **1**(1), 20 (2021). https://doi.org/10.1101/2020.10.06.327445
8. Jamal, S., Wimmer, H., Rebman Jr, C.M.: Perception and evaluation of text-to-image generative AI models: a comparative study of dall-e, google imagen, grok, and stable diffusion. Issues Inf. Syst. **25**(2), 277–292 (2024). https://doi.org/10.48009/2_iis_2024_123
9. Jiang, P., Ergu, D., Liu, F., Cai, Y., Ma, B.: A review of yolo algorithm developments. Procedia Comput. sci. **199**, 1066–1073 (2022). https://doi.org/10.1016/j.procs.2022.01.135
10. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: bootstrapping language-image pre-training for unified vision-language understanding and generation. In: International Conference on Machine Learning, pp. 12888–12900. PMLR (2022). https://doi.org/10.5555/3618408.3619222
11. Li, Y., et al.: Supervision exists everywhere: a data efficient contrastive language-image pre-training paradigm. arXiv preprint arXiv:2110.05208 (2021). https://doi.org/10.48550/arXiv.2110.05208
12. Li, Y., Wang, C., Jia, J.: Llama-vid: an image is worth 2 tokens in large language models. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds.) ECCV 2024. LNCS, vol. 15104, pp. 323–340. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-72952-2_19
13. Menendez, H.D., Bello-Orgaz, G., Atencia, C.R.: Deepstableyolo: Deepseek-driven prompt engineering and search-based optimization for ai image generation. In: XVI Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (2025). https://openreview.net/forum?id=hZucDPawRu
14. Menendez, H.D., Dakhama, A.: Automatic summarization evaluation: methods and practices. In: Menéndez, H.D., et al. (eds.) ICTSS 2024. LNCS, vol. 15383, pp. 169–181. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-80889-0_12
15. Rahim, A., Yuan, F., Barabady, J.: An ultralytics yolov8-based approach for road detection in snowy environments in the arctic region of norway. Comput. Mater. Continua **83**(3) (2025). https://doi.org/10.32604/cmc.2025.061575
16. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10684–10695 (2022). https://doi.org/10.1109/CVPR52688.2022.01042
17. Tang, R., et al.: What the DAAM: interpreting stable diffusion using cross attention. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 5644–5659 (2023). https://doi.org/10.18653/v1/2023.acl-long.310
18. Tansri, F.F., Monem, N., Weinberg, L.: Authenticity in biased diversity: Investigating the language of prompt performances in ai image generators. J. Aesthet. Creat. Art Manag. **4**(1), 75–101 (2025). https://doi.org/10.59997/jacam.v4i1.5414
19. Zhang, C., Zhang, C., Zhang, M., Kweon, I.S., Kim, J.: Text-to-image diffusion models in generative AI: a survey. arXiv preprint arXiv:2303.07909 (2023). https://doi.org/10.48550/arXiv.2303.07909

From Random to Pre-trained ViT Backbones for Improved Feature Extraction: Application to Diagnosis with Sensor Images

Luan B. Guerra, Ana Beatriz S. Zerati, Wallace Casaca, and Lucas C. Ribas(✉)

São Paulo State University, Institute of Biosciences, Humanities and Exact Sciences, São José do Rio Preto, Brazil
{luan.bonizi,ab.zerati,wallace.casaca,lucas.ribas}@unesp.br

Abstract. The development of robust and low-cost diagnostic tools is essential for the early and fast detection of infectious and chronic diseases, such as COVID-19 and prostate cancer. Biosensors based on molecular recognition have emerged as promising alternatives to conventional laboratory techniques. Recent studies indicate that microscopy images acquired from these materials contain discriminative morphological patterns associated with analyte presence, enabling the development of machine learning models capable of automatically performing medical diagnoses based on these images. In this work, we propose a feature fusion strategy that combines representations extracted from random and pretrained Vision Transformers (ViTs) to improve image embeddings from genosensors and immunosensors and enrich domain-specific descriptors. The proposed approach is built upon three token-based feature extraction strategies, which are explored to capture complementary information from the models. To systematically assess the effectiveness of the method, we evaluate the approach using ten different ViT backbones. Experiments conducted on SARS-CoV-2 immunosensor and PCA3 genosensor microscopy image datasets demonstrate that feature fusion consistently improves performance and that randomly initialized ViTs frequently achieve results comparable to, or even surpassing, pretrained models. Furthermore, the proposed method significantly outperformed various baseline Convolutional Neural Networks (CNNs) across all datasets. These findings highlight the strong intrinsic representational capacity of Transformer architectures and their suitability for small-scale biosensor image analysis.

Keywords: Biosensors · Vision Transformers · Feature fusion · Random models · Transfer Learning

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 510–528, 2026.
https://doi.org/10.1007/978-3-032-30491-9_32

1 Introduction

The rapid and accurate detection of infectious and chronic diseases is a critical challenge for healthcare systems worldwide [2]. The COVID-19 pandemic reinforced the urgent demand for sensitive, low-cost, and easily deployable diagnostic technologies [8,22,34], while prostate cancer continues to rank among the leading causes of cancer-related mortality in men [16,28]. In this context, biosensors based on molecular recognition have emerged as promising alternatives to conventional laboratory techniques, enabling early detection, portability, and reduced operational costs [18,25,34].

Two important classes of biosensors are genosensors and immunosensors. Genosensors rely on the specific recognition between nucleic acid sequences, enabling the detection of viral genetic material or molecular biomarkers associated with diseases [4,20]. Immunosensors, in turn, are based on antigen–antibody interactions and are widely used for detecting tumor-related biomarkers [13,22]. Recent studies [12,22,34,42] have demonstrated that, beyond traditional electrochemical measurements, microscopy images of functionalized substrates contain morphological and textural patterns associated with analyte presence. Such patterns can generate informative visual features that can be directly analyzed and classified for diagnostic purposes. This observation opens new opportunities for computer vision and machine learning approaches aimed at the automatic interpretation of biosensing signals.

Deep learning models have been employed as automatic feature extractors in image analysis [1,5,29,32]. More specifically, in the field of texture analysis, which includes biosensor images, CNNs predominate as backbones for feature extraction [22,28,36]. CNNs achieve strong performance in tasks involving texture and biosensor images because their convolutional operations introduce an inductive bias that effectively captures the local patterns typical of these images [27]. However, these models have a limited receptive range, causing them to capture fewer long-range relationships [17]. As an alternative to CNNs, Vision Transformers (ViTs) [11] are capable of modeling global dependencies through attention mechanisms, which is particularly beneficial when the diagnostic information relies on long-range spatial relationships between different regions [40]. Nevertheless, there are no studies in the literature that explore ViTs as feature extractors for images obtained from genosensors and immunosensors, even though these models present state-of-the-art results in several applications [3,38], often surpassing CNN-based approaches [35].

Despite their strong global modeling capability, ViTs present limitations in capturing local patterns and typically require large-scale datasets and high computational resources to achieve competitive performance [44]. These challenges are especially relevant in biosensing applications, where datasets are often limited and subtle local morphological variations may be crucial for analyte detection [28]. To address the first ViT limitation, several studies have proposed complementary strategies such as feature fusion, which combines representations extracted at different layers or levels to enrich visual descriptions and recover fine-grained information [39,43]. Regarding data dependency, recent works have

explored training-free Transformer architecture search [44,45], which uses models with randomly initialized weights to investigate the intrinsic representational capacity of the architecture itself and to automatically design ViT architectures. These studies show that, under certain conditions, untrained ViTs can achieve competitive results, enabling architectural exploration with reduced computational cost.

This work investigates the use of Vision Transformers as feature extractors for images obtained from genosensors and immunosensors. In addition, both pretrained and randomly initialized models are explored to assess the intrinsic representational capacity of transformer architectures. We propose feature fusion strategies based on representations extracted from both pretrained and randomly initialized models. In particular, we explore token-based representations obtained from the CLS token and from global average pooling (GAP) over all tokens, which allows capturing richer textural and fine-grained patterns in order to improve feature extraction and produce more robust representations of biosensor images. The combination of these representations aims to produce more robust image descriptors and mitigate known limitations of Vision Transformers in modeling local patterns, making the approach particularly suitable for small-scale biosensing datasets. The main contributions of this work are summarized as follows:

- A feature fusion strategy that integrates representations extracted from random and pretrained Vision Transformers to obtain richer and more robust texture image descriptors for biosensor images.
- An analysis of token-based feature extraction strategies, evaluating three different approaches to capture complementary information from ViT representations.
- A systematic evaluation across ten ViT backbones using microscopy images obtained from genosensors and immunosensors, assessing the effectiveness of the proposed approach for biosensor image representation.

2 Proposed Methodology

This section describes the proposed methodology for extracting and analyzing representations from Vision Transformers applied to biosensor images. An overview of the proposed pipeline is illustrated in Fig. 1.

2.1 Feature Extraction from Vision Transformers

When used as a feature extractor, the backbone is employed without its classification head, and the internal representations generated by the network are used as image descriptors. In this work, Vision Transformers are employed to generate such representations from biosensor images.

In a ViT, an input image is first divided into fixed-size patches, which are linearly projected into embedding vectors, combined with positional encodings

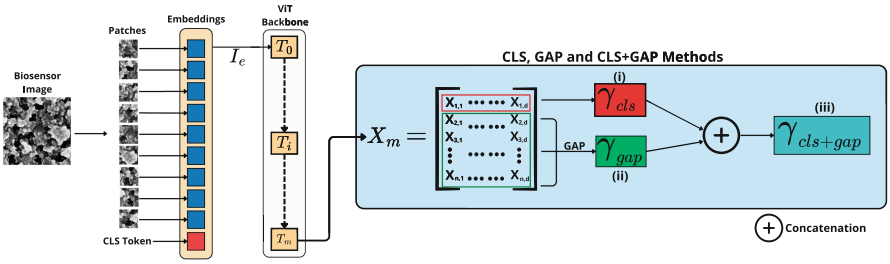


Fig. 1. Illustration of the methodology of this work. The Υ_{CLS} and Υ_{GAP} features are represented in (i) and (ii), respectively. (iii) represents the $\Upsilon_{CLS+GAP}$ feature, which is a combination of (i) and (ii).

and processed through a sequence of transformer blocks T. Each transformer block produces a matrix of token embeddings X that encode visual information extracted from different regions of the image. The matrix resulting of block T_i can be denoted as $X_i \in \mathbb{R}^{n \times d}$, where n is the number of tokens and d is the hidden dimension. Among these tokens, the first one corresponds to the classification token (CLS), while the remaining tokens correspond to spatial tokens associated with image patches.

The CLS token provides a compact global representation of the image, and is commonly used as a compact representation for image classification tasks. In contrast, spatial tokens encode spatial information from different image regions and, although they require additional processing to form a single representation, prior works [30,31] highlight their importance, showing that spatial tokens contribute significantly to robustness in complex visual scenarios. In order to investigate how these different sources of information contribute to the representation of biosensor images, we explore multiple strategies for extracting features from the final transformer layer. It is important to note that, since our work also explores the use of random ViTs, the representations were extracted using both pre-trained and random models.

CLS-Based Representation. The first strategy uses only the CLS token extracted from the final transformer block, shown in Fig. 1 (i). Formally, the CLS feature vector is defined as

$$\Upsilon^{\theta}_{CLS} = X_m[0], \tag{1}$$

where X_m denotes the output matrix of the last transformer block T_m and $\theta \in \{r, p\}$ denotes the model parameterization used to extract the representation, with r for randomly initialized and p for pretrained models. This representation captures global contextual information about the image.

Spatial-Token Representation. To obtain a compact representation from the spatial tokens, we apply global average pooling (GAP) over all tokens except the CLS token:

$$\Upsilon^{\theta}_{GAP} = GAP(X_m[1], X_m[2], ..., X_m[n-1]). \tag{2}$$

This operation aggregates the spatial information encoded by the transformer into a single feature vector while preserving contributions from all image regions. We refer to this method as the GAP method, which can be seen in Fig. 1 (ii).

Combined Representation. Since CLS and spatial tokens capture complementary information, we also investigate a combined representation obtained by concatenating both feature vectors:

$$\Upsilon^{\theta}_{CLS+GAP} = [\Upsilon^{\theta}_{CLS}; \Upsilon^{\theta}_{GAP}], \tag{3}$$

where $[;]$ denotes concatenation along the feature dimension, originating the CLS+GAP method (Fig. 1 (iii)). This strategy allows the representation to simultaneously encode global contextual information and aggregated spatial patterns.

All representations are extracted directly from the models without performing any fine-tuning or architectural modification. In addition, feature extraction is performed using both pretrained and randomly initialized Vision Transformers, enabling the analysis of the intrinsic representational capacity of the transformer architecture when applied to biosensor image recognition tasks.

2.2 Combination of Random and Pretrained Features

In addition to analyzing pretrained and randomly initialized models independently, we investigate whether combining their representations can yield more informative feature vectors. Pretrained ViTs are typically trained on large-scale natural image datasets such as ImageNet-21k, which may introduce biases related to natural image statistics. In contrast, randomly initialized models produce representations that are not influenced by such pretraining. As a result, the representations generated by these two types of models may capture different aspects of the visual patterns present in biosensor images.

To explore this hypothesis, we combine features extracted from pretrained and random models through feature concatenation. The fused representations are defined as

$$\Upsilon^{r+p}_{CLS} = [\Upsilon^{r}_{CLS}; \Upsilon^{p}_{CLS}], \tag{4}$$

$$\Upsilon^{r+p}_{GAP} = [\Upsilon^{r}_{GAP}; \Upsilon^{p}_{GAP}], \tag{5}$$

$$\Upsilon^{r+p}_{CLS+GAP} = [\Upsilon^{r}_{CLS+GAP}; \Upsilon^{p}_{CLS+GAP}], \tag{6}$$

where the features with p represent the vectors extracted from pretrained models, while the representations with r represent the vectors extracted from random models. Here the symbol [;] also represents a horizontal concatenation of the representations.

3 Experimental Setup

3.1 Selected Vision Transformer Architectures

To provide a comprehensive evaluation of transformer-based feature extractors, we considered ten ViT backbones that differ in architectural design, model size, internal modules and pre-training. As evaluating all existing models is impractical, we selected a representative set of widely used backbones that capture different design choices explored in the literature. The main characteristics of the selected architectures are summarized in Table 1. All models were obtained from the *timm* PyTorch library, which provides standardized and widely adopted implementations of modern vision architectures.

Table 1. Details of the ViT backbones evaluated in this work. The number of parameters refers only to the feature extractor, excluding the final classification layer. In the features column, [1] denotes the feature size for CLS and GAP, while [2] denotes the size for CLS+GAP.

Backbone	Pre-training	Feature extraction cost		Features	
		GFLOPs	param. (M)	[1]	[2]
ViT-B/16 [11]	IN-1k and IN-21k	16.9	85.8	768	1536
ViT-S/16 [11]	IN-1k and IN-21k	4.25	22	384	768
ViT-Betwix [10]	IN-1k	11.7	60.3	640	1280
ViT-RelPos [41]	IN-1k	16.9	86.4	768	1536
ViT-B/14 dinov2 [23]	IN-1k	16.9	85.8	768	1536
DeiT-B/16 [37]	IN-1k	16.9	85.8	768	1536
BeiTv2-B [24]	IN-1k	12.7	85.8	768	1536
BeiT-B [6]	IN-22k	12.7	85.8	768	1536
DeiT3-B [38]	IN-1k or IN-22k	16.9	85.8	768	1536
FlexiViT [7]	IN-1k or IN-21k	16.9	86.6	768	1536

3.2 Datasets

– **PCA3** [28]: The PCA3 dataset consists of 32 grayscale (8-bit) scanning electron microscopy (SEM) images of genosensors developed to detect the prostate cancer biomarker PCA3. The sensors were fabricated using layer-by-layer films with gold nanoparticles and an immobilized complementary DNA

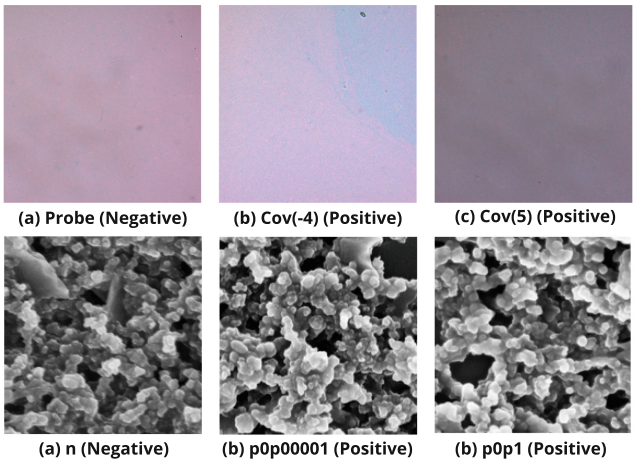


Fig. 2. Some samples of the datasets used in this work. The first row shows images from different classes of dataset SARS-CoV-2 Immunosensor [22], while the second row shows images from dataset PCA3 [28].

probe on carbon electrodes, and imaged after exposure to different PCA3 concentrations to capture morphology changes caused by DNA hybridization, covering eight experimental conditions: one negative control, one blank control, and six positive classes with concentrations ranging from 10^{-5} to $1\ \mu \cdot \text{mol L}^{-1}$. The pre-processed images have 300×300 pixels, following a similar protocol to that observed in [28]. We extracted three 300×300 pixels windows from each image in the dataset in order to increase the total number of samples, resulting in 96 images. Thus, the dataset supports both binary (presence vs. absence of PCA3) and multiclass classification settings. Some samples of this dataset can be seen in the second row of Fig. 2 and the number of samples per class for each classification configuration is reported in [28].

– **SARS-CoV-2 Immunosensor** [22]: The immunosensor dataset consists of 894 RGB optical microscopy images acquired from plasmonic substrates functionalized for the detection of SARS-CoV-2 antibodies. The sensing platform was based on gold-coated substrates modified with specific biological recognition elements, and images were captured after exposure to samples containing different antibody concentrations to record surface alterations induced by antigen–antibody interactions. The dataset comprises optical images obtained under controlled illumination and magnification conditions, representing distinct experimental groups including negative controls and positive samples with varying antibody concentrations ranging from $10\ \text{ng} \cdot \text{mL}^{-1}$ to $100\ \mu\text{g} \cdot \text{mL}^{-1}$. Similar to the PCA3 dataset, this dataset can be formulated either as a binary classification task (presence vs. absence of antibodies) or as a multiclass problem depending on the concentration levels considered. The per-class sample counts for the binary and multiclass problem are reported in

[22]. In the experiments, we rescaled all images to 1344×1024 pixels. Some samples of this dataset can be seen in the first row of Fig. 2.

3.3 Classification Setup

After extracting all feature vectors, a linear Support Vector Machine (SVM) classifier [9] was trained and used to classify the test features. The SVM was configured with a linear kernel and a penalty parameter $C = 1$. For the PCA3 dataset, a resampling procedure was adopted during the classification step using 100 random configurations (i.e., 100 trials). In each configuration, the same number of samples was selected from each class, resulting in a uniform class distribution. For each trial, a 10-fold cross-validation scheme was employed to split the data into training and test sets. The average accuracy and the standard deviation across all trials were then computed to evaluate the performance of the proposed method. For the SARS-CoV-2 dataset, a 5-fold cross-validation scheme was adopted and repeated three times. The final performance was reported as the mean accuracy and the corresponding standard deviation.

We also fine-tuned the ViT models to compare their performance with that of the proposed representations. In this case, the models were fine-tuned using the same cross-validation configuration as the respective datasets, where, for each configuration of the training and test folds, fine-tuning was performed over 40 epochs with a learning rate of 3×10^{-5}. From this, we calculated the mean and standard deviation of the accuracies.

4 Results and Discussion

As described in Sect. 3.2, the datasets used in this study support both binary and multiclass classification settings. To provide a more comprehensive evaluation of the proposed methodology, experiments were conducted under both scenarios.

4.1 Binary Classification Analysis

Table 2 presents the results of the binary classification on the SARS-CoV-2 dataset. It can be observed that there is no clear pattern among the results of the different models, with performance varying between CLS, GAP, and CLS+GAP methods. This variation is due to architectural differences among the models. Nevertheless, the GAP and CLS+GAP methods have higher average accuracy compared to the CLS method, indicating that spatial tokens provide valuable information about immunosensor images, especially when these tokens are combined with the CLS token.

Furthermore, an important observation is that combining random and pre-trained features leads to an increase in the average accuracy across all methods for the SARS-CoV-2 dataset. This result suggests that features extracted from randomly initialized models can provide complementary information to

Table 2. Accuracy of each proposed method considering the binary classification of the dataset SARS-CoV-2 Immunosensor. The results in bold represent the best result for a specific backbone.

Backbone	Pretrained			Random+Pretrained			Fine-tunning
	CLS	GAP	CLS+GAP	CLS	GAP	CLS+GAP	
ViT-B [11]	89.97 ± 2.15	89.11 ± 2.17	90.86 ± 1.72	96.05 ± 1.32	**96.42 ± 1.08**	96.23 ± 1.28	90.16 ± 4.90
ViT-S [11]	90.19 ± 2.31	88.29 ± 3.40	89.78 ± 2.59	95.12 ± 1.52	**95.53 ± 1.53**	95.45 ± 1.22	93.4 ± 0.65
ViT-Betwix [10]	92.73 ± 1.95	91.76 ± 2.34	92.47 ± 2.24	93.25 ± 2.02	92.50 ± 2.50	**94.44 ± 2.34**	93.28 ± 0.65
ViT-RelPos [41]	74.68 ± 1.99	89.22 ± 2.19	84.38 ± 1.22	77.52 ± 2.69	91.50 ± 2.07	87.58 ± 2.21	**95.19 ± 2.15**
ViT-B dinov2 [23]	83.60 ± 2.87	85.20 ± 2.78	85.31 ± 2.60	85.91 ± 2.43	87.36 ± 2.19	**87.77 ± 2.53**	84.45 ± 2.78
DeiT-B [37]	88.44 ± 2.33	89.23 ± 2.00	89.60 ± 2.22	97.20 ± 1.17	96.65 ± 1.31	**97.32 ± 1.28**	86.25 ± 9.29
BeiTv2-B [24]	89.15 ± 2.53	89.67 ± 2.59	89.78 ± 2.79	92.17 ± 1.58	89.41 ± 2.70	**93.44 ± 1.70**	74.85 ± 11.97
BeiT-B [6]	95.00 ± 1.94	94.11 ± 1.89	95.67 ± 1.88	97.69 ± 1.13	**97.72 ± 1.11**	97.13 ± 1.47	83.68 ± 6.53
DeiT3-B [38]	85.64 ± 3.05	84.94 ± 3.04	84.93 ± 3.00	88.03 ± 1.81	87.92 ± 3.00	88.74 ± 2.30	**89.03 ± 2.68**
FlexViT [7]	87.70 ± 2.74	87.92 ± 2.12	88.33 ± 2.08	96.12 ± 0.73	**96.94 ± 1.06**	96.53 ± 0.99	89.60 ± 2.36
Mean accuracy	87.71	88.95	89.11	91.91	93.20	**93.46**	78.98

the image representation. Unlike pretrained models, whose representations may reflect biases inherited from large-scale datasets such as ImageNet-21k, random models produce unbiased representations that can act as a form of regularization when combined with pretrained features. As a result, the fusion of both types of representations leads to more domain-specific and robust image descriptors for biosensor image recognition. It is also worth noting that the proposed method outperforms even results of fine-tunned models, where the best-performing method outperformed the average accuracy of the tuned models by 14.36%, demonstrating the effectiveness of our approach when applied to biosensor datasets, which often suffer from a lack of data for an efficient training of deep learning models. Among the evaluated methods, the CLS+GAP strategy combined with random and pretrained features achieved the highest mean accuracy.

Table 3. Accuracy of each proposed method considering the binary classification of the dataset PCA3. The results in bold represent the best result for a specific backbone.

Backbone	Pretrained			Random+Pretrained			Fine-tunning
	CLS	GAP	CLS+GAP	CLS	GAP	CLS+GAP	
ViT-B [11]	97.50 ± 7.50	97.22 ± 5.69	97.50 ± 7.50	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	99.6 ± 3.07
ViT-S [11]	93.89 ± 9.44	97.22 ± 5.69	95.00 ± 8.03	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.00**
ViT-Betwix [10]	95.83 ± 7.16	95.00 ± 8.03	95.00 ± 8.03	**100.0 ± 0.0**	96.51 ± 3.13	99.65 ± 1.31	93.23 ± 12.89
ViT-RelPos [41]	83.06 ± 14.40	95.83 ± 7.16	95.83 ± 7.16	81.96 ± 5.26	**99.30 ± 1.79**	97.23 ± 3.23	94.86 ± 11.62
ViT-B dinov2 [23]	94.17 ± 11.81	95.28 ± 9.46	95.28 ± 9.46	98.60 ± 2.33	**99.30 ± 1.79**	98.95 ± 2.11	92.13 ± 13.82
DeiT-B [37]	99.17 ± 2.50	97.78 ± 6.67	98.06 ± 3.94	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	99.65 ± 3.08
BeiTv2-B [24]	98.33 ± 5.00	98.33 ± 5.00	98.33 ± 5.00	98.95 ± 2.11	**99.30 ± 1.79**	**99.30 ± 1.79**	95.45 ± 9.04
BeiT-B [6]	94.17 ± 9.50	95.28 ± 8.06	95.28 ± 8.06	99.30 ± 1.79	97.58 ± 3.22	98.26 ± 2.46	**100.0 ± 0.0**
DeiT3-B [38]	96.94 ± 4.72	98.33 ± 5.00	99.17 ± 2.50	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	96.34 ± 8.51
FlexViT [7]	99.17 ± 2.50	96.94 ± 4.72	98.33 ± 5.00	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	90.19 ± 15.38
Mean accuracy	95.28	96.19	96.78	97.88	99.20	**99.34**	96.14

Table 3, with the results from the binary classification of PCA3 database, shows something similar to the Table 2, where there is no discernible pattern among the results of the different ViTs and we have also a increase in the average accuracy of the GAP and CLS+GAP methods compared to the CLS method, which indicates that spatial tokens provide valuable information about genosensor images. Once again, it is very clear to verify an increase in the mean accuracy of all methods when combining random and pretrained features, again indicating that features extracted by random models contain important information and serve as a regularization, which reduces the biases of the representations and makes them much more robust, since, in this case, many methods have achieved 100% accuracy. In this case, our approach also outperformed the fine-tuned models, which achieved similar results to those of methods that rely solely on features from pre-trained ViTs, but were outperformed by methods that combine random and pre-trained features, resulting in a 3.2% increase in average accuracy. The CLS+GAP method with combined features also presented the best results for the PCA3 binary classification.

It is worth noting that, as shown in Table 3, several models achieve 100% accuracy on PCA3. However, due to the small size of this dataset, such performance may reflect the inherent simplicity of the binary task or potential overfitting, rather than clear superiority of a given method. In the original PCA3 study [28], a t-SNE analysis showed that the binary features form two well-defined and easily separable clusters, even in two dimensions. This supports the interpretation that the PCA3 binary classification task is intrinsically simple, corroborating the near-perfect supervised results reported in Table 3.

4.2 Multiclass Classification Analysis

From Table 4, which presents the results of the multiclass classification on the SARS-CoV-2 dataset, a noticeable decrease in accuracy can be observed across all methods when compared to the binary classification results. This drop in performance is expected, since multiclass classification requires models to recognize subtle differences in patterns among multiple classes, making the problem considerably more challenging. However, similarly to the binary classification scenario, the GAP and CLS+GAP strategies achieve higher average accuracies than the CLS method, demonstrating, once again, the importance of spatial tokens in detecting patterns of biosensor images. Moreover, when random and pretrained features are combined, an improvement in performance is again observed across all models. This indicates that random features, despite the increased difficulty of distinguishing interclass patterns, still contain valuable information and contribute to more robust image descriptors when fused with pretrained features. Once again, the proposed method outperformed the fine-tuned models in all cases without the need to train the ViTs, achieving an 8.11% increase in average accuracy, indicating the effectiveness of different feature extraction methods for biosensor images, particularly when random and pre-trained representations are combined. Unlike the previous cases, the method that achieved the highest aver-

age accuracy in the SARS-CoV-2 multiclass classification was the GAP strategy with combined random and pretrained features.

Table 4. Accuracy of each proposed method considering the multiclass classification of the dataset SARS-CoV-2 Immunosensor.

Backbone	Pretrained			Random+Pretrained			Fine-tunning
	CLS	GAP	CLS+GAP	CLS	GAP	CLS+GAP	
ViT-B [11]	73.19 ± 2.83	71.59 ± 3.52	72.78 ± 2.82	79.38 ± 2.19	**79.57 ± 2.56**	79.57 ± 2.50	69.57 ± 1.75
ViT-S [11]	69.20 ± 3.03	68.42 ± 3.16	69.20 ± 2.48	76.62 ± 3.22	77.63 ± 3.57	**77.67 ± 3.32**	69.23 ± 3.9
ViT-Betwix [10]	73.98 ± 3.13	73.12 ± 3.35	74.09 ± 3.03	78.97 ± 3.41	77.41 ± 3.04	**79.46 ± 2.75**	68.68 ± 5.06
ViT-RelPos [41]	56.71 ± 2.59	74.91 ± 2.75	68.09 ± 2.82	61.04 ± 1.77	**79.12 ± 2.44**	70.36 ± 3.01	76.85 ± 1.41
ViT-B dinov2 [23]	63.46 ± 3.95	65.62 ± 3.71	65.29 ± 4.01	71.59 ± 3.72	**73.41 ± 3.02**	72.37 ± 3.20	65.55 ± 1.81
DeiT-B [37]	72.34 ± 2.57	71.96 ± 2.89	72.75 ± 3.49	80.91 ± 2.55	80.95 ± 2.79	**80.98 ± 2.81**	70.25 ± 1.62
BeiTv2-B [24]	76.33 ± 2.57	76.81 ± 2.44	77.78 ± 2.71	81.66 ± 2.36	76.92 ± 2.37	**82.10 ± 2.94**	73.38 ± 1.87
BeiT-B [6]	80.09 ± 2.91	74.65 ± 2.95	78.41 ± 3.63	**83.56 ± 2.83**	81.24 ± 2.50	81.99 ± 2.65	72.60 ± 2.5
DeiT3-B [38]	66.93 ± 3.51	67.94 ± 2.53	67.75 ± 3.06	75.02 ± 3.51	**75.69 ± 2.76**	75.36 ± 3.56	64.54 ± 1.03
FlexViT [7]	70.88 ± 3.24	69.16 ± 2.11	70.66 ± 2.82	78.86 ± 2.51	**79.23 ± 2.86**	79.01 ± 3.31	69.35 ± 2.55
Mean accuracy	70.31	71.42	71.68	76.76	**78.12**	77.89	70.01

Table 5. Accuracy of each proposed method considering the multiclass classification of the dataset PCA3.

Backbone	Pretrained			Random+Pretrained			Fine-tunning
	CLS	GAP	CLS+GAP	CLS	GAP	CLS+GAP	
ViT-B [11]	41.11 ± 16.56	39.72 ± 15.81	43.06 ± 13.90	67.42 ± 9.50	65.95 ± 10.02	**68.46 ± 9.86**	49.3 ± 16.08
ViT-S [11]	37.78 ± 15.53	30.83 ± 20.20	34.17 ± 15.22	53.88 ± 8.44	**61.51 ± 8.14**	59.81 ± 8.00	48.56 ± 17.15
ViT-Betwix [10]	33.89 ± 16.52	30.83 ± 17.32	32.22 ± 13.56	**55.49 ± 9.54**	47.21 ± 10.52	51.68 ± 8.56	44.34 ± 16.08
ViT-RelPos [41]	13.33 ± 7.22	32.78 ± 8.77	28.61 ± 9.79	23.23 ± 10.32	**56.16 ± 9.36**	52.05 ± 8.29	40.20 ± 17.90
ViT-B dinov2 [23]	34.17 ± 20.03	34.17 ± 22.36	29.17 ± 18.64	48.65 ± 7.51	**52.11 ± 6.63**	51.05 ± 8.27	35.09 ± 23.07
DeiT-B [37]	40.83 ± 11.25	43.61 ± 15.52	45.00 ± 14.74	59.09 ± 8.80	**62.16 ± 7.87**	60.79 ± 8.48	37.45 ± 16.53
BeiTv2-B [24]	61.39 ± 16.59	61.39 ± 15.03	59.44 ± 14.50	**66.33 ± 8.61**	65.93 ± 8.88	64.58 ± 9.99	38.05 ± 22.87
BeiT-B [6]	39.44 ± 11.17	40.83 ± 13.15	36.39 ± 13.97	**64.61 ± 7.72**	56.26 ± 6.06	61.51 ± 5.81	44.51 ± 13.39
DeiT3-B [38]	29.44 ± 15.48	32.22 ± 11.12	30.56 ± 13.61	58.33 ± 7.84	**61.79 ± 6.26**	60.42 ± 7.25	45.43 ± 16.38
FlexViT [7]	53.89 ± 14.07	41.11 ± 15.41	50.83 ± 15.52	**70.53 ± 10.02**	68.75 ± 7.36	70.14 ± 8.30	49.57 ± 20.11
Mean accuracy	38.53	38.75	38.94	56.76	59.78	**60.05**	43.25

Table 5, which presents the results of the multiclass classification on the PCA3 dataset, shows an even greater drop in performance when compared to the binary classification results for this dataset. As expected, this is mainly due to the increased difficulty of multiclass classification. In addition, the PCA3 dataset poses a more challenging classification problem than the SARS-CoV-2 dataset, exhibiting even smaller variability in patterns among classes [28]. Nevertheless, results similar to those observed in Table 4 can still be identified. In particular, combining random and pretrained features consistently improves the performance of the evaluated methods, further highlighting the importance of

random features and feature fusion for constructing more robust image representations. In this scenario, fine-tunned models outperformed methods that rely solely on pre-trained features in terms of average accuracy, however, there is a significant improvement in performance compared to methods that combine random and pre-trained features, with an increase of 16.8% in the mean accuracy, which reaffirms the importance of features extracted by models without training. Finally, in this case, the CLS+GAP strategy with combined random and pretrained features achieved the highest average accuracy.

4.3 Impact of Transformers Architecture on Feature Extraction

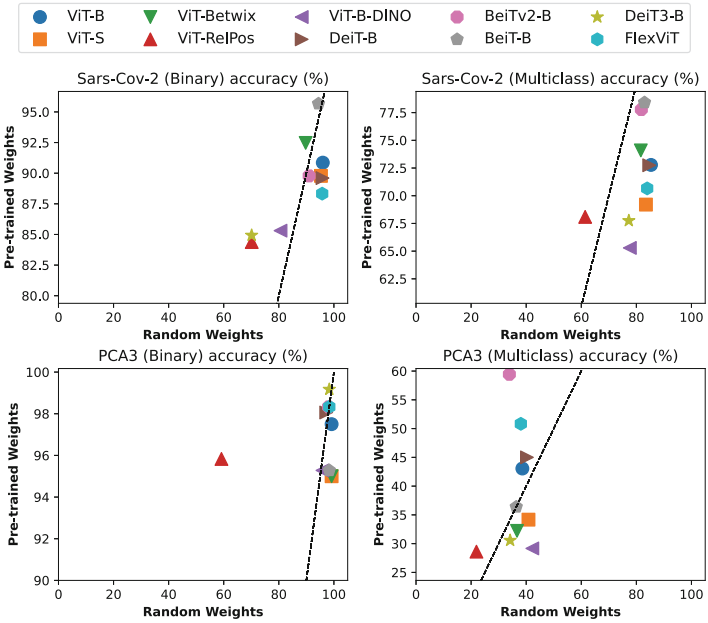


Fig. 3. Correlation between the accuracies of random and pretrained ViT models. The dotted line represents the diagonal of the plane, i.e., x = y.

In addition to exploring other feature extraction methods using ViTs, we also decided to analyze the influence of the Transformer architecture itself by comparing the robustness of features extracted by random and pretrained ViTs. Figure 3 shows the results of the SVM model used to classify the features extracted with the CLS+GAP method, since it was the method that yielded the best results in most cases. The graphs show the direct relationship between random ViT results (X-axis) and pretrained results (Y-axis), where it is also possible to verify the contribution of learned weights over random ones by observing the distance of

the points from the dashed line, which represents where the random results would be equal to the pretrained ones, i.e., $X = Y$.

For the SARS-CoV-2 dataset, in both binary and multiclass classification scenarios, the vast majority of models lie close to the dotted line or to its right. This indicates that, in general, randomly initialized models perform as well as, or even better than, their pretrained counterparts. These results suggest that, for the type of application explored in this work, Vision Transformers do not necessarily depend on pretrained weights to achieve strong performance, and that pretraining may even degrade performance in some cases.

In the binary classification of the PCA3 dataset, we observed the same case as before, where the models are close to or to the right of the dashed line, indicating that the random models achieved results as good as, or even better than, the pretrained ones, with the exception of the ViT-RelPos model, which showed a clear dependence on pretrained weights. In the multi-class classification of the PCA3 dataset, half of the models are to the left of the dashed line, indicating a certain dependence on pre-training for some models. However, these models still remain relatively close to this line, while many others lie above or to the right of it, once again indicating that ViT models are not strongly dependent on pretrained weights to perform feature extraction in biosensor images.

To provide a more rigorous analysis of why random ViTs can rival, and in some cases outperform, pretrained models, we conducted a comparative analysis of the attention maps produced by both settings. This analysis allows us to examine potential differences in attention patterns and in the way each model captures spatial and structural relationships within the images, offering more concrete insights into the factors that may influence their performance. Figure 4 shows representative samples from both datasets, together with the attention maps generated by the randomly initialized and pretrained models. A clear distinction can be observed between the two settings: randomly initialized models exhibit more spatially distributed attention, whereas pretrained models tend to concentrate attention on more localized regions. This behavior may be attributed to the biases acquired during ImageNet pretraining, whose images are predominantly object-centric. As a result, pretrained ViTs tend to focus on salient object regions learned from object-centric pretraining datasets [21], often assigning less attention to other image areas. In contrast, randomly initialized models are not constrained by these learned biases and therefore distribute attention more uniformly across the image. This property is particularly advantageous for texture-related tasks, where discriminative information is often homogeneous and spatially distributed, as observed in biosensor images.

This analysis further clarifies why concatenating random and pretrained features can improve performance, as reported in Sects. 4.1 and 4.2. The attention maps suggest that randomly initialized ViTs, even without training, can capture distributed textural information in biosensor images, likely because they are not constrained by the biases acquired during pretraining on generic object-centric datasets such as ImageNet [21]. In contrast, pretrained models benefit from large-scale training and can capture relevant structural patterns, but their learned

attention mechanisms may also lead them to focus on localized regions that are not always the most informative for texture-oriented biosensor data. Therefore, combining random and pretrained representations yields a more robust image descriptor by integrating complementary information captured by both model types. These findings also highlight the importance of the Transformer architecture itself for biosensor image feature extraction, since randomly initialized models achieved performance comparable to, and in several cases even better than, that of pretrained models.

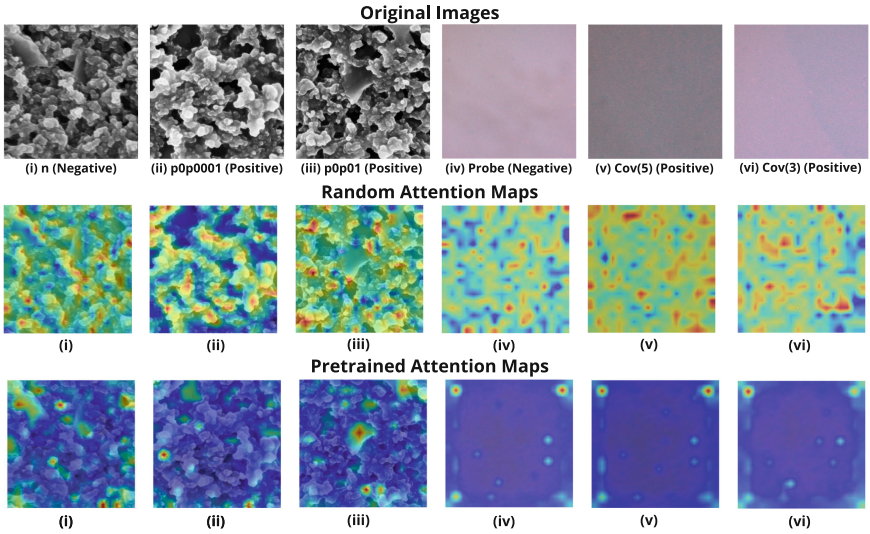


Fig. 4. Representative samples from the evaluated datasets with their corresponding attention maps obtained from random and pretrained ViT backbones.

4.4 CNN Baseline Comparison

Finally, in order to conduct a more in-depth analysis of our proposal, we decided to perform a brief comparison between the performance of different baseline CNNs and the proposed method. Table 6 shows the results of different CNNs used as feature extractors in the PCA3 and SARS-CoV-2 Immunosensor datasets, along with our own results. It is worth noting that, regarding CNNs, the features of the last layer were extracted and these features were used to train and test an SVM classifier, respecting the respective configurations of each dataset. In the case of ViTs, the results of the CLS+GAP method combining random and pretrained features were considered.

As we can see on Table 6, the features proposed in this study achieved overall better results compared to baseline CNNs. For the PCA3 dataset, the model

Table 6. Comparison of classification accuracies between baseline CNN-based features and the proposed representation (CLS+GAP strategy combining random and pretrained features) on the PCA3 and SARS-CoV-2 Immunosensor datasets. Bold values indicate the best performance.

Backbone	PCA3		SARS-CoV-2 Immunosensor	
	Binary	Multiclass	Binary	Multiclass
ConvNext-B [19]	96.11 ± 7.88	48.89 ± 9.31	90.45 ± 2.02	75.73 ± 3.08
ResNet-101 [14]	95.28 ± 8.06	37.78 ± 19.73	82.66 ± 2.64	63.05 ± 3.7
MobileNetV4 [26]	90.83 ± 10.1	20.28 ± 11.92	83.41 ± 2.47	62.6 ± 2.46
VGG 11 [33]	93.06 ± 9.4	30.56 ± 14.33	83.45 ± 3.36	66.45 ± 3.4
DenseNet-121 [15]	93.06 ± 11.74	42.22 ± 19.28	81.02 ± 1.88	63.8 ± 2.99
Proposed representation				
ViT-B	**100.0 ± 0.0**	68.46 ± 9.86	96.23 ± 1.28	79.57 ± 2.50
ViT-S	**100.0 ± 0.0**	59.81 ± 8.00	95.45 ± 1.22	77.67 ± 3.32
ViT-Betwix	99.65 ± 1.31	51.68 ± 8.56	94.44 ± 2.34	79.46 ± 2.75
ViT-RelPos	97.23 ± 3.23	52.05 ± 8.29	87.58 ± 2.21	70.36 ± 3.01
ViT-B dinov2	98.95 ± 2.11	51.05 ± 8.27	87.77 ± 2.53	72.37 ± 3.20
DeiT-B	**100.0 ± 0.0**	60.79 ± 8.48	**97.32 ± 1.28**	80.98 ± 2.81
BeiTv2-B	99.30 ± 1.79	64.58 ± 9.99	93.44 ± 1.70	**82.10 ± 2.94**
BeiT-B	98.26 ± 2.46	61.51 ± 5.81	97.13 ± 1.47	81.99 ± 2.65
DeiT3-B	**100.0 ± 0.0**	60.42 ± 7.25	88.74 ± 2.30	75.36 ± 3.56
FlexViT	**100.0 ± 0.0**	**70.14 ± 8.30**	96.53 ± 0.99	79.01 ± 3.31

FlexViT obtained the best overall results, Surpassing ConvNext results by 3.89% and 21.25% in binary and multi-class classifications, respectively. It is interesting to note that even though ConvNext-B obtained the best result among the CNNs, its performance was still worse than all ViTs, both in binary and multiclass classification in the PCA3 dataset.

In the case of the SARS-CoV-2 Immunosensor dataset, the model with the best performance in binary classification was DeiT-B, surpassing the accuracy of ConvNext-B by 6.87%, while in multiclass classification, BeiTV2-B showed the highest accuracy, surpassing the performance of ConvNext-B by 6.46%. Again, in the SARS-CoV-2 Immunosensor dataset, with the exception of the ViT-RelPos, ViT-B dinov2 and DeiT3-B models, the ViTs models significantly outperformed ConvNext, which was the CNN with the best result.

5 Conclusion

In this work, we conducted an investigation of Vision Transformers as feature extractors for biosensor image analysis, considering both genosensor and immunosensor datasets for the detection of COVID-19 and prostate cancer. Different token-based extraction and feature fusion strategies were explored, and

the use of both pretrained and random models allowed us to analyze the representational capacity of the Transformer architecture itself. This way, our work demonstrated that ViTs can serve as powerful feature extractors in biosensing applications without any type of end-to-end training or fine-tuning, making the approach viable for small-scale datasets. The results highlight the central role of random representations and feature fusion in improving classification performance. The combination of features extracted from random and pretrained models consistently increased the average accuracy across datasets and experimental settings, with random features acting as a form of regularization that mitigates biases inherited from large-scale pretraining datasets and yields more robust image descriptors. Moreover, in many scenarios, randomly initialized ViTs achieved performance comparable to, and sometimes surpassing, their pretrained versions, reinforcing the importance of the Transformer architecture itself in capturing relevant morphological and textural patterns from biosensor images, even without learned semantic priors from large-scale datasets.

For future work, it would be valuable to explore multi-layer feature aggregation in ViTs to investigate whether fusing tokens from different layers can further enrich image descriptors. Additionally, future studies could extend the approach to other biosensing applications, including genosensor-based imaging for COVID-19 detection and other infectious or oncological biomarkers, enabling a broader evaluation of Transformer-based representations in diverse biochemical and morphological contexts.

Acknowledgments. The authors acknowledge support from the São Paulo Research Foundation (FAPESP), Brazil (Process N. #2025/14029-8, #2025/27044-5, #2025/01057-3 #2024/04492-0 and #2024/04492-0) and Brazilian National Council for Scientific and Technological Development (CNPq) (grants #305206/2024-9 and #130337/2026-9).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Agrawal, S., Sahu, S.P.: Image-based Parkinson disease detection using deep transfer learning and optimization algorithm. Int. J. Inf. Technol. (Singapore) **16**(2), 871–879 (2024)
2. Akbari Nakhjavani, S., Mirzajani, H., Carrara, S., Onbaşlı, M.C.: Advances in biosensor technologies for infectious diseases detection. TrAC - Trends Anal. Chem. **180** (2024)
3. Azad, R., et al.: Advances in medical image analysis with vision transformers: a comprehensive review. Medical Image Anal. **91** (2024)
4. Babaei, A., Pouremamali, A., Rafiee, N., Sohrabi, H., Mokhtarzadeh, A., de la Guardia, M.: Genosensors as an alternative diagnostic sensing approaches for specific detection of virus species: a review of common techniques and outcomes. TrAC, Trends Anal. Chem. **155**, 116686 (2022)

5. Bansal, M., Kumar, M., Sachdeva, M., Mittal, A.: Transfer learning for image classification using vgg19: Caltech-101 image data set. J. Ambient. Intell. Humaniz. Comput. **14**(4), 3609–3620 (2023)
6. Bao, H., Dong, L., Piao, S., Wei, F.: Beit: Bert pre-training of image transformers (2022). https://arxiv.org/abs/2106.08254
7. Beyer, L., et al.: Flexivit: one model for all patch sizes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14496–14506 (2023)
8. Bhatt, G., Bhattacharya, S.: Biosensors on chip: a critical review from an aspect of micro/nanoscales. J. Micromanuf. **2**(2), 198–219 (2019)
9. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
10. Darcet, T., Oquab, M., Mairal, J., Bojanowski, P.: Vision transformers need registers. In: International Conference on Learning Representations (ICLR) (2024)
11. Dosovitskiy, A., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
12. Fam, M.K., Ismail, N.I., Zahari, A.I., Ghazali, N.A., Hassan, N.: Biosensor technologies in cancer: tools for early detection and prognostic monitoring. Biomarker Res. **14**(1) (2026)
13. Gomes, H.C., et al.: Laser-induced graphene-based fabry-pérot cavity label-free immunosensors for the quantification of cortisol. Sensors Actuators Rep. **7** (2024)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
15. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
16. Ibau, C., Md Arshad, M., Gopinath, S.C.: Current advances and future visions on bioelectronic immunosensing for prostate-specific antigen. Biosens. Bioelectron. **98**, 267–284 (2017)
17. Kang, M., Song, H., Park, S., Yoo, D., Pereira, S.: Benchmarking self-supervised learning on diverse pathology datasets. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3344–3354 (2023)
18. Kassal, P., Steinberg, M.D., Steinberg, I.M.: Wireless chemical sensors and biosensors: a review. Sens. Actuators, B Chem. **266**, 228–245 (2018)
19. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
20. Nakowong, P., et al.: Detection of high-risk HPV 16 genotypes in cervical cancers using isothermal DNA amplification with electrochemical genosensor. Talanta **269** (2024)
21. Naseer, M., Ranasinghe, K., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Intriguing properties of vision transformers. In: Proceedings of the 35th International Conference on Neural Information Processing Systems. NIPS '21, Red Hook, NY, USA. Curran Associates Inc. (2021)
22. Oiticica, P.R.A., et al.: Using machine learning and optical microscopy image analysis of immunosensors made on plasmonic substrates: Application to detect the sars-cov-2 virus. ACS Sensors **10**(2), 1407–1418 (2025). pMID: 39960416
23. Oquab, M., et al.: Dinov2: learning robust visual features without supervision (2024)

24. Peng, Z., Dong, L., Bao, H., Ye, Q., Wei, F.: Beit v2: masked image modeling with vector-quantized visual tokenizers (2022). https://arxiv.org/abs/2208.06366
25. Qian, L., et al.: Biosensors for early diagnosis of pancreatic cancer: a review. Transl. Res. **213**, 67–89 (2019). https://doi.org/10.1016/j.trsl.2019.08.002
26. Qin, D., et al.: Mobilenetv4: Universal models for the mobile ecosystem. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds.) ECCV 2024. LNCS, vol. 15098, pp. 78–96. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-73661-2_5
27. Ribas, L.C., Casaca, W., Fares, R.T.: Conditional generative adversarial networks and deep learning data augmentation: a multi-perspective data-driven survey across multiple application fields and classification architectures. AI **6**(2), 32 (2025)
28. Rodrigues, V.C., et al.: Electrochemical and optical detection and machine learning applied to images of genosensors for diagnosis of prostate cancer with the biomarker pca3. Talanta **222**, 121444 (2021)
29. Scabini, L., Zielinski, K.M., Ribas, L.C., Gonçalves, W.N., Baets, B.D., Bruno, O.M.: Radam: texture recognition through randomized aggregated encoding of deep activation maps. Pattern Recogn. **143**, 109802 (2023)
30. Scabini, L., Sacilotti, A., Zielinski, K.M., Ribas, L.C., De Baets, B., Bruno, O.M.: A comparative survey of vision transformers for feature extraction in texture analysis. J. Imaging **11**(9), 304 (2025)
31. Scabini, L., et al.: Vortex: challenging CNNs at texture recognition by using vision transformers with orderless and randomized token encodings. Neurocomputing 133852 (2026). https://doi.org/10.1016/j.neucom.2026.133852
32. Senan, E.M., Jadhav, M.E., Rassem, T.H., Aljaloud, A.S., Mohammed, B.A., Al-Mekhlafi, Z.G.: Early diagnosis of brain tumour MRI images using hybrid techniques between deep and machine learning. Comput. Math. Methods Med. **2022** (2022)
33. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICLR) (2015)
34. Soares, J.C., et al.: Detection of a sars-cov-2 sequence with genosensors using data analysis based on information visualization and machine learning techniques. Mater. Chem. Front. **5**, 5658–5670 (2021)
35. Takahashi, S., et al.: Comparison of vision transformers and convolutional neural networks in medical image analysis: a systematic review. J. Med. Syst. **48**(1) (2024)
36. Togaçar, M.: Biotransnet: Detection of plant stress through the conversion of biosensor data into RGB channels and combination with transformer networks. J. Crop Health **77**(5) (2025)
37. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention (2021). https://arxiv.org/abs/2012.12877
38. Touvron, H., Cord, M., Jégou, H.: Deit III: Revenge of the VIT (2022). https://arxiv.org/abs/2204.07118
39. Wang, J., Yu, X., Gao, Y.: Feature fusion vision transformer for fine-grained visual categorization (2022)
40. Wang, X.O.: Transformer-based unsupervised contrastive learning for histopathological image classification. Med. Image Anal. **79**, 102430 (2022)
41. Wu, K., Peng, H., Chen, M., Fu, J., Chao, H.: Rethinking and improving relative position encoding for vision transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)

42. Zerati, A.B.S., Guerra, L.B., Oliveira, O.N., Jr., Ribas, L.C.: Sars-cov-2 detection in genosensor images via deep learning and data augmentation. Revista de Informática Teórica e Aplicada **33**(2), 269–275 (2026)
43. Zhang, K., Cao, J., Li, H.s., Cai, Q.: Cross-layer feature fusion vision transformer for fine-grained visual classification. In: 2023 5th International Conference on Data-driven Optimization of Complex Systems (DOCS), pp. 1–6 (2023)
44. Zhou, H., Ma, K., Chen, H., Yao, R.: Training-free transformer architecture search. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9682–9691 (2022)
45. Zhou, Q., Zhu, Y.: When training-free NAS meets vision transformers: a neural tangent kernel perspective. In: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7405–7409. IEEE (2024)

Classification of Dynamic Libras Signs Related to the Hospital Context Using Recurrent Neural Networks

Pedro de Araújo Bhering Bittencourt, Luiz Maurílio da Silva Maciel(✉), Marcelo Bernardes Vieira, and Saulo Moraes Villela

Department of Computer Science, Federal University of Juiz de Fora, Juiz de Fora, Brazil
pedro.bittencourt@estudante.ufjf.br, {luiz.maciel,marcelo.bernardes,saulo.moraes}@ufjf.br

Abstract. Communication between Brazilian healthcare professionals and Deaf individuals remains a significant challenge in hospital environments, especially due to the scarcity of professionals fluent in Brazilian Sign Language (Libras). In this context, sign recognition systems emerge as a promising alternative to reduce communication barriers. This work proposes the construction of a dataset composed of ten dynamic Libras signs related to hospital triage, as well as the development of recurrent neural network–based models for their classification. The signs are represented by three-dimensional landmarks extracted using MediaPipe Holistic, capturing spatial and temporal information of the movements, which are classified by LSTM networks. Initially, we investigated the 10-class classification problem, achieving an average accuracy of 83.83% and revealing frequent confusions between pairs of semantically similar signs. To address this limitation, we proposed a hierarchical strategy based on expert models, consisting of a general 7-class classifier combined with dedicated models for distinguishing frequently confused signs. This approach resulted in a significant improvement in overall performance, increasing the average accuracy to 92.44% and substantially reducing confusion errors between classes. Additional experiments conducted on an external subject, under less controlled acquisition conditions, achieved an average accuracy of 98.17%, further supporting the generalization capability of the proposed approach. These results indicate the potential of the method for communication support applications in healthcare environments.

Keywords: Machine learning · Social inclusion · Computer vision · Sign language

1 Introduction

Brazilian Sign Language (Libras) plays a fundamental role in promoting the social inclusion of the Deaf community by reducing communication barriers

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 529–543, 2026.
https://doi.org/10.1007/978-3-032-30491-9_33

and ensuring the full exercise of citizenship. In Brazil, this right is guaranteed by the Brazilian Law for the Inclusion of Persons with Disabilities (Law No. 13,146/2015), which establishes equal access to essential services, including healthcare, through appropriate communication resources [1].

However, in the hospital context, communication between healthcare professionals and Deaf patients remains a significant challenge. The shortage of professionals trained in Libras compromises the quality of care and perpetuates inequalities in access to health services. Studies indicate that most medical students and healthcare professionals are not fluent in Libras, despite recognizing its relevance in clinical practice [2]. This scenario reveals a structural gap that directly affects the safety, autonomy, and dignity of Deaf patients.

Given these limitations, assistive technologies based on automatic sign recognition have shown promise in supporting communication in clinical environments. Several studies have explored the use of machine learning and neural networks for the classification of Libras signs, often accompanied by the construction of custom datasets [3–5]. However, most of these datasets are collected in controlled environments, with limited variability in participants and scenarios, which restricts the generalization capability of the models in real-world contexts, such as hospitals.

Recent approaches have sought to mitigate these limitations by representing gestures through body landmarks, which capture relevant spatial and temporal information in a compact and robust manner. This representation directly integrates with sequential models, such as Long Short-Term Memory (LSTM) neural networks, which have been widely employed in the interpretation of dynamic sign language gestures [6–8].

In this context, this work proposes the creation of a dataset composed of dynamic Libras signs related to the healthcare domain, with a focus on hospital triage, as well as the development and evaluation of recurrent neural network–based models for the classification of these signs. The proposed approach aims to contribute to the reduction of communication barriers in clinical environments and to the promotion of more inclusive healthcare for the Deaf community.

2 Related Work

In recent years, advances in deep learning have significantly increased research in sign language recognition, particularly in video-based approaches. Among the most promising strategies are the extraction of body landmarks – such as hands, face, and pose – and the use of temporal modeling architectures, including LSTM networks.

Huang and Chouvatut [8] proposed a video-based pipeline that combines a ResNet network for visual feature extraction with an LSTM layer to capture temporal dependencies, achieving competitive results when evaluated on the LSA64 dataset [9].

Gandhi, Shah, and Chandane [7] integrated the MediaPipe library for the automatic extraction of hand, face, and pose landmarks, using these vectors as input to an LSTM network for American Sign Language (ASL) recognition.

Samaan et al. [6] compared different recurrent architectures, including GRU, LSTM, and Bi-LSTM, demonstrating that LSTM-based architectures achieve superior performance in scenarios with higher temporal complexity.

In the Brazilian context, Sarmento [10] highlighted the effectiveness of using only landmarks in combination with LSTM networks for Libras sign classification, outperforming image-based approaches.

Other studies have also reported promising results by integrating spatial feature extraction techniques with temporal modeling [11,12], reinforcing the robustness of landmark-based representations and recurrent neural networks for dynamic sign language recognition.

In summary, the literature indicates that the combination of landmark-based representations and LSTM models constitutes an effective approach for dynamic sign classification. In this sense, the present work adopts a similar strategy, focusing on the recognition of Libras signs in the healthcare context.

3 Proposed Dataset

We did not identify public datasets that include signs from Brazilian Sign Language (Libras) specifically targeted at the hospital context. Thus, this work proposes the construction of a dedicated dataset to represent a more realistic application scenario in clinical environments.

The dataset is composed of 10 dynamic Libras signs related to hospital triage, performed by 9 participants fluent in Libras, with diversity in physical and visual characteristics. In addition, one participant was recorded on two occasions under different conditions – varying clothing, accessories (such as glasses), and hairstyle (tied or loose hair) – which contributed to increasing the diversity of the collected material. Each participant performed each sign up to five times, totaling 492 videos. We carried out the recordings in an indoor environment with controlled lighting, a fixed camera, and a uniform background in order to reduce external interference and ensure full visibility of hand and arm movements.

The selected signs – Allergy, Autism, Bronchitis, Cramp, Seizure, Diarrhea, Fracture, Athlete's Foot, Heart Attack, and Measles – are dynamic in nature and involve coordinated movements of the hands, arms, and body posture, thus requiring appropriate temporal modeling. These signs were chosen to represent common symptoms and medical conditions encountered in hospital settings. Videos were recorded at a spatial resolution of 1280×720 pixels (HD) or 640×480 pixels, with a temporal resolution of 30 frames per second (FPS).

This dataset provides an appropriate basis for evaluating Libras sign recognition models in the healthcare context, serving as experimental support for the methods proposed in this work.

4 Proposed Method

Initially, we applied preprocessing to the videos to standardize the model input. As the duration of sign execution varies between participants, we resampled each

video to contain 30 frames. Frame selection was performed using a sampling strategy based on a normal distribution centered at the midpoint of the video duration. This strategy prioritizes frames with the most relevant movements without completely disregarding the remaining parts of the sequence.

To reduce overfitting and increase the variability of the dataset, we applied data augmentation. The geometric transformations used included mild rotations, horizontal and vertical translations, and spatial cropping, which were consistently applied to all frames of each video to preserve the temporal coherence of the gesture. Operations such as horizontal flipping and color transformations were not used to avoid semantic distortions of the signs and potential interference with the landmark extraction process.

Furthermore, these augmentation operations were integrated into the frame sampling process. This procedure was repeated 20 times for each video. As a result, each video produced multiple variations of the same gesture, contributing to more diverse and representative cases.

After preprocessing, body and hand landmarks were extracted using MediaPipe Holistic, which enables the simultaneous detection of these regions with spatial and temporal consistency. Although facial landmarks were available, they were discarded because facial expressions were not sufficiently perceptible in some of the videos, which could introduce inconsistencies into the data.

For each frame, 75 three-dimensional landmarks were considered: 33 correspond to body pose and 42 to the hands (21 for each hand). We concatenated these points into a single one-dimensional vector containing 225 values per frame. Thus, each video was represented by a tensor of size 30×225, which we used as input to the recurrent neural network. To optimize processing and avoid repeating the extraction step, all feature vectors were stored in binary files using the NumPy library.

Figure 1 illustrates an example of pose and hand landmarks estimated on a video frame from the dataset used in this work. Although the visualization is presented in two dimensions, the extracted points correspond to three-dimensional coordinates and preserve the temporal structure of the gesture throughout the sequence.

The model proposed in this work is composed of four main layers: Input, Masking, LSTM, and Classification. This architecture was designed to handle sequential data obtained from landmark extraction while preserving the temporal dependencies of the performed gestures.

Initially, the flattened landmark vectors are fed into the Input layer. Next, a Masking layer is applied to ignore null values resulting from occasional failures in the extraction process, preventing such information from influencing the model's learning.

The core of the architecture consists of an LSTM recurrent network, which is responsible for capturing the temporal dependencies present in the frame sequences. Other recurrent architectures, including Gated Recurrent Units (GRU) and bidirectional LSTM variants, were also evaluated during model development. However, these alternatives did not show consistent improvements

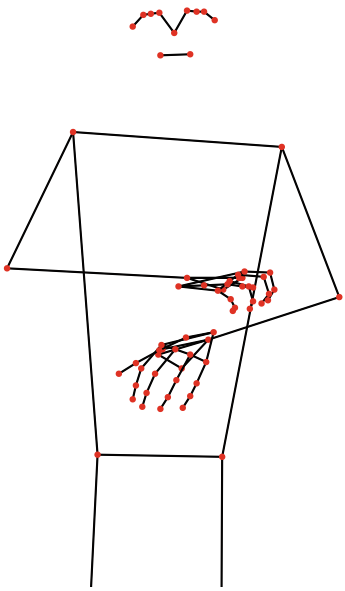


Fig. 1. Pose and hand landmarks estimated by MediaPipe Holistic.

over the standard LSTM in terms of average accuracy or validation stability. Therefore, LSTM was chosen as the primary architecture, offering the best balance between performance, complexity, and generalization. This layer employs the ReLU activation function and uses recurrent dropout as a regularization technique. The application of dropout directly to recurrent connections follows well-established approaches in the literature for recurrent neural networks, contributing to the reduction of overfitting without compromising the modeling of long-term dependencies [13].

Finally, a fully connected layer performs the classification, producing the probability that the video belongs to each of the ten sign classes considered in this work.

5 Experiments and Results

We conducted the experiments on a machine equipped with an Intel(R) Core(TM) i5-10300H CPU at 2.50 GHz, 16 GB of RAM, and running Windows 11, with the implementations executed within the Windows Subsystem for Linux (WSL2) using Ubuntu 22.04.5 LTS. All models were implemented in Python (version 3.10.2) using the TensorFlow library.

The dataset was evaluated using the k-fold cross-validation technique with $k = 10$, following the leave-one-subject-out scheme. In this procedure, at each iteration, the data of one individual are used exclusively for testing, while the data of the remaining individuals compose the training set. Taking into account

the different recording conditions adopted, we treated each occurrence of a subject recorded twice as a distinct subject, resulting in a total of ten subjects in the dataset.

In all experiments, the AdamW optimizer was employed, configured with a weight decay of 0.005. The hyperparameters related to the learning rate, the number of neurons in the LSTM layer, and the recurrent dropout rate vary among the evaluated models.

In addition, each cross-validation fold was evaluated 20 times. This strategy arises from the frame selection process based on sampling from a normal distribution, which can generate a distinct version of each video for each sample. Thus, for each subject used as the test set, we obtained 20 independent evaluations, leading to more consistent results.

Note that these 20 samples correspond to the versions generated during the frame sampling process. The sampled sequences corresponding to the test subject were used without any modification. However, for the training subjects, these sequences were used to generate augmented variations. The performance of the models was assessed using the metrics: accuracy, precision, recall, F1 score, and F2 score. We evaluated each sample and reported the mean values.

5.1 10-Class Classification Problem

For the classification problem encompassing the 10 classes of the dataset, the model that achieved the best performance was configured with a batch size of 128, a learning rate of 0.001, 64 neurons in the LSTM layer, a recurrent dropout rate of 0.2, and 300 training epochs. This model achieved an average general accuracy of 83.83%, with a standard deviation of 0.72%, indicating consistent performance across the different samplings.

Table 1 presents the mean values and standard deviations of the performance metrics per class achieved from the evaluation of the 20 samples used in the cross-validation procedure.

We observe that classes such as Autism, Bronchitis, Seizure, Diarrhea, and Measles achieved high performance with a relatively balanced relationship between precision and recall across the 20 test sets, indicating the good generalization capability of the model for these signs. The Diarrhea class stands out, maintaining mean recall values close to 98% and F1 scores above 98%, which evidences high gestural separability.

On the other hand, the Allergy, Cramp, and Athlete's Foot classes presented the lowest average performance, particularly in terms of recall, indicating a higher incidence of false negatives across the evaluated sets. In particular, the Cramp class exhibited a mean recall below 60%, while Athlete's Foot showed difficulties in both precision and recall, resulting in reduced F1 score values. These results highlight the structural limitations of the model in distinguishing these classes.

The average of all classes, around 84%, reinforces the presence of relevant performance asymmetries between classes. This indicates that despite satisfac-

Table 1. Performance evaluation per class for the 10-class classification problem.

Class	Precision	Recall	F1 score	F2 score
Allergy	73.86 ± 2.05	73.33 ± 3.30	73.53 ± 2.13	75.70 ± 2.77
Autism	94.89 ± 2.11	93.50 ± 2.44	94.15 ± 1.43	93.75 ± 1.92
Bronchitis	85.84 ± 1.32	98.96 ± 1.04	91.92 ± 0.94	96.02 ± 0.90
Cramp	85.77 ± 4.20	56.87 ± 3.30	68.31 ± 2.78	60.94 ± 3.07
Seizure	83.08 ± 3.27	90.10 ± 0.77	86.42 ± 1.89	88.58 ± 1.07
Diarrhea	99.80 ± 0.61	97.86 ± 1.76	98.81 ± 0.89	98.23 ± 1.40
Fracture	71.61 ± 2.15	95.10 ± 2.05	81.68 ± 1.77	89.23 ± 1.75
Athlete's Foot	64.36 ± 2.62	55.90 ± 3.32	59.79 ± 2.61	57.38 ± 3.01
Heart Attack	98.95 ± 1.16	83.50 ± 2.09	90.55 ± 1.32	86.18 ± 1.79
Measles	85.74 ± 2.50	93.62 ± 1.65	89.49 ± 1.82	91.92 ± 1.61
Average	84.39 ± 0.71	83.87 ± 0.72	83.47 ± 0.76	83.56 ± 0.75

tory overall performance, certain categories negatively impact the quality of the model.

Figure 2 shows the confusion matrix obtained for the test set that achieved the highest overall accuracy, namely 84.96%. We note recurrent error patterns between specific pairs of signs, such as Allergy and Athlete's Foot, Bronchitis and Heart Attack, and Cramp and Fracture. These confusions are attributed to intrinsic gestural similarities between the signs, which motivated the reformulation of the problem through class merging and the development of expert models to perform refined classification within each group. This approach leads to the development of a 7-class classification model and three expert models, each designed to handle a specific pair of confused classes.

5.2 7-Class Classification Problem

For the 7-class classification problem, the best-performing experiment used a batch size of 128, a learning rate of 0.001, 32 neurons in the LSTM layer, a recurrent dropout rate of 0.2, and 400 training epochs. To ensure class balance after unification, 10 samples from each original class were selected prior to merging, while the test sets were kept unchanged.

Table 2 presents the mean values and standard deviations of the performance metrics obtained from the evaluation of the 20 test sets used in the cross-validation procedure for the 7-class problem.

Note that all classes achieved mean values greater than 90% for precision, recall, F1, and F2 scores, indicating high and stable performance. The average of all classes was close to 96%, showing balanced sensitivity and precision between classes.

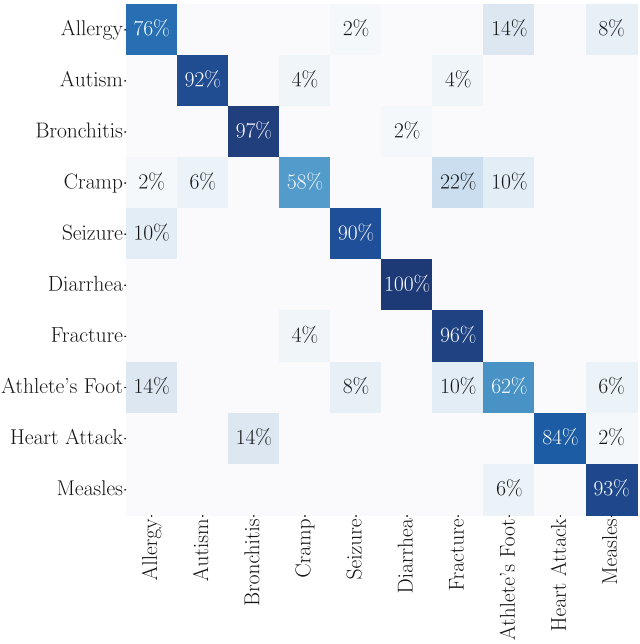


Fig. 2. Confusion matrix for the original 10-class classification problem.

5.3 Classification Problems Using Expert Models

Autism, Bronchitis+Heart Attack, and Cramp+Fracture classes performed particularly well, with mean metrics near 100%, reflecting the improved separability after merging previously confusing classes.

After the 7-class model, we developed the model responsible for classifying the signal between the Bronchitis and Heart Attack classes, as well as the model responsible for classifying the signal between the Cramp and Fracture classes. This choice was motivated by the fact that both followed the same development methodology described in the previous processes.

For the model that classifies signals between Bronchitis and Heart Attack, the experiment that achieved the best performance used a batch size of 32, a learning rate of 0.001, 32 neurons in the LSTM layer, a recurrent dropout rate of 0.2, and 150 training epochs. This configuration achieved a mean accuracy of 93.42% across the evaluated test sets, with a standard deviation of 0.60.

Table 3 presents the mean values and standard deviations of the performance metrics achieved from the evaluation of the 20 test sets for the Bronchitis and Heart Attack signal classification problem.

By comparing the results in Table 3 with the 10-class scenario (Table 1), we can observe a clear improvement in the balance between precision and recall. In the original model, the Heart Attack class had reduced recall due to frequent confusion with Bronchitis, an issue that was mitigated in the expert model.

Table 2. Performance evaluation per class for the 7-class classification problem.

Class	Precision	Recall	F1 score	F2 score
Allergy+Athlete's Foot	91.93 ± 1.02	91.00 ± 1.34	91.46 ± 0.86	91.18 ± 1.10
Autism	99.71 ± 0.70	100.00 ± 0.00	99.85 ± 0.35	99.94 ± 0.14
Bronchitis+Heart Attack	99.29 ± 0.72	99.08 ± 0.64	99.18 ± 0.57	99.12 ± 0.58
Cramp+Fracture	97.82 ± 0.95	98.11 ± 1.03	97.96 ± 0.70	98.05 ± 0.85
Seizure	97.28 ± 1.78	88.40 ± 1.36	92.62 ± 1.16	90.04 ± 1.20
Diarrhea	98.15 ± 1.26	96.73 ± 1.76	97.43 ± 1.12	97.01 ± 1.45
Measles	88.23 ± 1.42	99.47 ± 1.14	93.50 ± 1.06	96.99 ± 1.02
Average	96.06 ± 0.50	96.11 ± 0.45	96.00 ± 0.46	96.05 ± 0.45

Table 3. Performance evaluation per class for the Bronchitis and Heart Attack signal classification problem.

Class	Precision	Recall	F1 score	F2 score
Bronchitis	89.54 ± 0.85	98.02 ± 0.80	93.59 ± 0.58	96.20 ± 0.64
Heart Attack	97.92 ± 0.84	89.00 ± 1.00	93.24 ± 0.63	90.65 ± 0.83
Average	93.73 ± 0.59	93.51 ± 0.60	93.41 ± 0.60	93.42 ± 0.60

As a result, Bronchitis maintained a high recall while showing increased precision, and Heart Attack achieved more balanced performance metrics, reflected in the F1 and F2 scores, indicating a reduction in confusion errors between classes.

For the model that classifies signals between Cramp and Fracture, the best-performing experiment used a batch size of 64, a learning rate of 1×10^{-4}, 32 neurons in the LSTM layer, a recurrent dropout rate of 0.2, and 150 training epochs. This configuration achieved an average accuracy of 90.87% across the evaluation sets, with a standard deviation of 2.18.

Table 4 presents the means and standard deviations of the performance metrics achieved from the evaluation of the 20 test sets for the Cramp and Fracture classification problem.

Table 4. Performance evaluation per class for the Cramp and Fracture signal classification problem.

Class	Precision	Recall	F1 score	F2 score
Cramp	94.65 ± 1.59	86.25 ± 4.29	90.19 ± 2.56	87.77 ± 3.61
Fracture	87.96 ± 3.28	95.30 ± 1.45	91.44 ± 1.88	93.71 ± 1.33
Average	91.30 ± 1.92	90.78 ± 2.21	90.82 ± 2.22	90.74 ± 2.25

Table 4 indicates that the expert model for the Cramp and Fracture signals achieved high and well-balanced performance, with consistent values of precision, recall, F1 score, and F2 score for both classes.

Compared to the 10-class scenario (Table 1), we note a significant improvement, particularly for the Cramp class, which previously exhibited low recall. The expert model substantially mitigated this limitation, while the Fracture class maintained high recall and achieved gains in precision, resulting in greater classification stability.

After resolving the first two pairs of unified classes, we developed an expert model to classify between the Allergy and Athlete's Foot signals. Unlike the previous cases, we observed that the main distinction between these signals occurs at the beginning of the movement. Therefore, we proposed a strategy for the automatic detection of gesture onset based on the temporal analysis of hand landmarks.

This approach consists of estimating motion energy from the temporal variation in the vertical positions of the hand centers. For each valid frame, the center of each hand is computed as the mean of its landmark coordinates. The resulting vertical trajectories are then smoothed using a Savitzky–Golay filter [14] to reduce frame-to-frame noise. Hand velocities are obtained via temporal gradients, and motion energy is defined as the sum of the magnitudes of the vertical velocities of both hands.

To detect the start of the gesture, we employ an adaptive threshold based on the initial behavior of the signal. The first five frames are assumed to represent a resting phase, and the threshold is defined as the mean energy within this window plus one standard deviation. The onset is identified as the first frame at which the energy remains above this threshold for five consecutive frames, thereby preventing isolated fluctuations from being misclassified as motion.

If the detected onset occurs within this initial window, a second threshold is estimated using frames 10 to 20. In this case, a more restrictive criterion is adopted, defined as the mean plus two standard deviations of the energy. The detection process is then repeated, ensuring greater robustness in sequences with noisy or unstable initial behavior.

Once the gesture onset is determined, the subsequent 30 frames are selected to compose the input for classification. These frames are uniformly sampled according to the video frame rate, ensuring temporal consistency of the gesture representation.

The best-performing model used a batch size of 32, a learning rate of 0.001, 128 neurons in the LSTM layer, a recurrent dropout rate of 0.1, and 250 training epochs. Cross-validation resulted in an overall accuracy of 97.00%, with 98.00% for the Allergy class and 96.00% for Athlete's Foot, indicating high and consistent performance.

Table 5 presents the precision, recall, F1 score, and F2 score metrics achieved for this classification problem. We observe a balanced behavior between the classes, with values above 96% across all metrics, indicating that the proposed strategy was effective in reducing the confusions observed in the 10-class model.

Table 5. Performance evaluation per class for the Allergy and Athlete's Foot signal classification problem.

Class	Precision	Recall	F1 score	F2 score
Allergy	96.08	98.00	97.03	97.61
Athlete's Foot	97.96	96.00	96.97	96.39
Average	97.02	97.00	97.00	97.00

5.4 Integration of 7-Class Model with Expert Models

Finally, the integration of the 7-class model with the expert models was performed. This combination resulted in an average accuracy of 92.44% in the 20 test sets, with a standard deviation of 0.78.

Table 6 presents the means and standard deviations of the performance metrics obtained after applying the expert-based strategy. This table enables a direct comparison with the results of the original 10-class model (Table 1).

Table 6. Performance evaluation per class using the expert model strategy.

Class	Precision	Recall	F1 score	F2 score
Allergy	82.45 ± 2.32	88.50 ± 1.77	85.35 ± 1.77	87.21 ± 1.67
Autism	99.71 ± 0.70	100.00 ± 0.00	99.85 ± 0.35	99.94 ± 0.14
Bronchitis	88.20 ± 1.40	96.35 ± 1.45	92.09 ± 1.14	94.60 ± 1.25
Cramp	89.85 ± 2.94	82.71 ± 5.15	86.05 ± 3.50	84.00 ± 4.46
Seizure	97.28 ± 1.78	88.40 ± 1.36	92.62 ± 1.16	90.04 ± 1.20
Diarrhea	98.15 ± 1.26	96.73 ± 1.76	97.43 ± 1.12	97.01 ± 1.45
Fracture	88.01 ± 3.23	95.10 ± 1.61	91.38 ± 1.96	93.57 ± 1.50
Athlete's Foot	97.79 ± 0.07	88.60 ± 2.69	92.95 ± 1.52	90.29 ± 2.25
Heart Attack	97.91 ± 0.84	88.80 ± 1.17	93.13 ± 0.72	90.48 ± 0.97
Measles	88.23 ± 1.42	99.47 ± 1.14	93.50 ± 1.06	96.99 ± 1.02
Average	92.76 ± 0.75	92.47 ± 0.79	92.43 ± 0.80	92.41 ± 0.80

Compared to the initial 10-class problem, in which the F1 score was around 83% (average of all classes), the strategy based on expert models increased this value to approximately 92%, accompanied by consistent improvements in average precision and recall. These results indicate a general enhancement in discriminative capacity.

The greatest gains are concentrated in the classes that exhibited greater gestural overlap in the original scenario, particularly the Allergy+Athlete's Foot and Cramp+Fracture pairs, whose performance metrics became more balanced after the application of expert models. In contrast, previously well-defined classes

such as Autism, Diarrhea, and Measles maintained good performance levels, demonstrating that the proposed approach did not compromise their recognition.

Figure 3 presents the confusion matrix for the same sample that generated the confusion matrix shown in Fig. 2. We can observe a significant reduction in class confusions, reflected in the increase in overall accuracy from 84.96% to 93.50%. Thus, decomposing the problem into expert classifiers has resulted in a more consistent system, outperforming the original model both in terms of average performance and in reducing ambiguities among semantically similar classes.

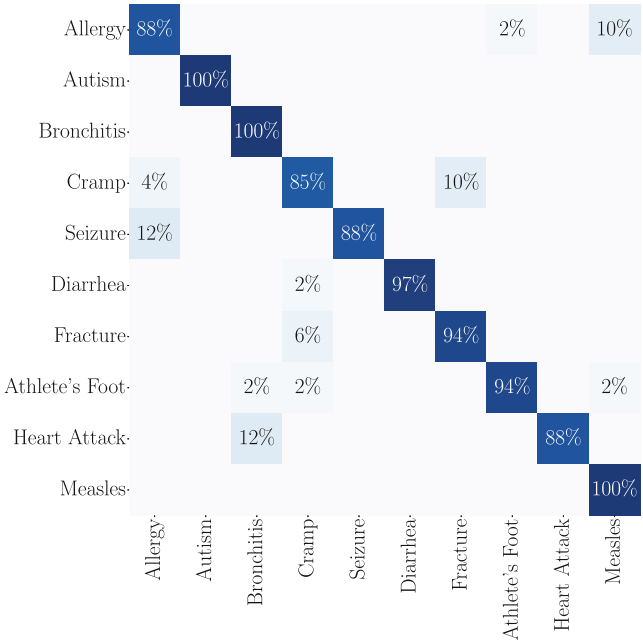


Fig. 3. Confusion matrix for the expert-model strategy.

5.5 Additional Test Set

In addition to the dataset collected with Libras professionals, an additional test set was recorded with a participant who is also fluent in Brazilian Sign Language.

We recorded a total of 60 videos, corresponding to 6 samples for each sign. Unlike the controlled acquisition process adopted for the main dataset, these recordings were performed under uncontrolled conditions, including variations in background and lighting. Furthermore, a different recording device was used, namely a Samsung S20 FE smartphone, with a resolution of 1080×1920 pixels and a frame rate of 60 FPS.

Despite these differences, the recording dynamics and the segmentation of individual sign executions followed the same procedures described for the main dataset.

This additional dataset is used to evaluate the generalization capability of the models trained on the original dataset, simulating more realistic usage conditions.

To ensure consistency with the evaluation protocol adopted in the previous experiments, we also performed 20 independent evaluations on this additional test set.

Table 7 presents the performance metrics obtained for this additional test set using the expert model strategy. The results show a general average accuracy of 98.11%, with a standard deviation of 1.66%, and all average metrics above 98%. This improvement can be attributed to the reduced variability of the dataset, which consists of a single participant, as well as the consistency in gesture execution.

Table 7. Performance evaluation per class using the expert model strategy in the additional test set.

Class	Precision	Recall	F1 score	F2 score
Allergy	98.57 ± 4.29	100.00 ± 0.00	99.23 ± 2.31	99.68 ± 0.97
Autism	99.29 ± 3.11	100.00 ± 0.00	99.62 ± 1.68	99.84 ± 0.70
Bronchitis	90.42 ± 9.34	99.17 ± 3.63	94.33 ± 5.75	97.10 ± 3.87
Cramp	96.43 ± 6.19	100.00 ± 0.00	98.08 ± 3.33	99.19 ± 1.40
Seizure	100.00 ± 0.00	96.67 ± 6.67	98.18 ± 3.64	97.24 ± 5.52
Diarrhea	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
Fracture	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
Athlete's Foot	100.00 ± 0.00	95.83 ± 7.22	97.73 ± 3.94	96.55 ± 5.97
Heart Attack	100.00 ± 0.00	90.00 ± 11.06	94.36 ± 6.43	91.63 ± 9.36
Measles	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
Average	98.47 ± 1.35	98.17 ± 1.66	98.15 ± 1.68	98.12 ± 1.71

Despite being recorded under uncontrolled conditions, the results indicate that the proposed approach is capable of generalizing well to new users and acquisition setups. However, it is important to note that the limited number of participants may simplify the classification task, and further evaluation with a larger and more diverse set of participants is necessary to comprehensively assess the generalization capability of the proposed method in real-world environments.

6 Conclusion

In this work, we proposed a method for the classification of Brazilian Sign Language (Libras) signs related to the healthcare domain. To this end, we constructed a dedicated dataset composed of 10 dynamic Libras signs associated

with hospital triage scenarios. Based on this dataset, we developed machine learning models using LSTM to handle temporal motion sequences and adequately represent the dynamic nature of the signs.

We observed that the model considering the 10 classes of the dataset presented recurrent confusions between specific pairs of signs, achieving an average accuracy of 83.83%. To address these confusions, we proposed a strategy based on expert models, restructuring the problem into a general 7-class classifier combined with dedicated classifiers for the most conflicting sign pairs. With this approach, the average accuracy increased to 92.44%, along with a substantial reduction in misclassifications between semantically similar signs. In addition, the low standard deviation observed across the test sets indicates that the system maintains consistent performance when exposed to new instances, reinforcing its practical potential for communication support applications in healthcare environments.

Additionally, an evaluation conducted on an additional set comprising an external subject not included in the original dataset, demonstrated that the proposed approach is capable of generalizing to new users and acquisition conditions. Despite variations in recording setup and environment, the model maintained high performance, reinforcing its robustness and potential applicability in real-world healthcare scenarios. Nevertheless, further validation with a larger and more diverse population remains necessary to fully assess its generalization capability.

In future works, it is possible to expand the sign set, include a greater diversity of users, and investigate more recent architectures, such as models based on attention mechanisms or vision transformers, with the aim of further improving performance and generalization capacity.

Acknowledgments. The authors would like to thank the Minas Gerais State Research Foundation - FAPEMIG, Brazil. This study was financed in part by the Federal University of Juiz de Fora - UFJF.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. BRASIL: Lei nº 13.146, de 6 de julho de 2015. Diário Oficial da União: seção 1, Brasília, DF, 7 jul. 2015 (2015). https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13146.htm, institui a Lei Brasileira de Inclusão da Pessoa com Deficiência (Estatuto da Pessoa com Deficiência). Accessed on: September 5, 2025. (in Portuguese)
2. Oliveira, A.S.R., Oliveira, C.V., Jesus, R.F., Quintanilha, L.F., Avena, K.M.: Ensino da língua brasileira de sinais durante a graduação em medicina: a percepção dos futuros médicos. Audiol. - Commun. Res. **27** (2022). https://doi.org/10.1590/2317-6431-2022-2634, fapUNIFESP (SciELO). (in Portuguese)

3. Rocha, J., Lensk, J., Ferreira, T., Ferreira, M.: Towards a tool to translate Brazilian sign language (libras) to Brazilian Portuguese and improve communication with deaf. In: 2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp. 1–4. IEEE (2020). https://doi.org/10.1109/VL/HCC50065.2020.9127257
4. Costa, C.F.F., Souza, R.S., Santos, J.R., Santos, B.L., Costa, M.G.F.: A fully automatic method for recognizing hand configurations of Brazilian sign language. Res. Biomed. Eng. **33**, 78–89 (2017). https://doi.org/10.1590/2446-4740.03816
5. Rezende, T.M., Almeida, S.G.M., Guimarães, F.G.: Development and validation of a Brazilian sign language database for human gesture recognition. Neural Comput. Appl. **33**(16), 10449–10467 (2021). https://doi.org/10.1007/s00521-021-05802-4
6. Samaan, G.H., et al.: Mediapipe's landmarks with RNN for dynamic sign language recognition. Electronics **11**(19), 3228 (2022). https://doi.org/10.3390/electronics11193228, https://www.mdpi.com/2079-9292/11/19/3228
7. Gandhi, D., Shah, K., Chandane, M.: Dynamic sign language recognition and emotion detection using mediapipe and deep learning. In: 13th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–6. IEEE, Virtual Conference (2022). https://doi.org/10.1109/ICCCNT54827.2022.9984592
8. Huang, J., Chouvatut, V.: Video-based sign language recognition via resnet and LSTM network. J. Imaging **10**(6), 149 (2024). https://doi.org/10.3390/jimaging10060149, published: 20 June 2024
9. Ronchetti, F., Quiroga, F., Estrebou, C., Lanzarini, L., Rosete, A.: Lsa64: an Argentinian sign language dataset (2023). https://doi.org/10.48550/arXiv.2310.17429, instituto de Investigación en Informática LIDI, Facultad de Informática, Universidad Nacional de La Plata; Instituto Superior Politécnico José Antonio Echeverría
10. Sarmento, A.H.A.: Integração de Datasets de Vídeo para Tradução Automática da LIBRAS com Aprendizado Profundo. Dissertação de mestrado, Universidade de São Paulo, São Paulo, SP (2023). https://doi.org/10.11606/D.55.2023.tde-10012024-093541. (in Portuguese)
11. Ma, D., Hirota, K., Dai, Y., Jia, Z.: Dynamic sign language recognition based on improved residual-LSTM network. In: The 7th International Workshop on Advanced Computational Intelligence and Intelligent Informatics (IWACIII2021). Beijing, China (2021). https://iwaciii2021.bit.edu.cn/docs/2021-12/9caf45c507224bdca4933ad139fff639.pdf
12. Ridwang, Ilham, A.A., Nurtanio, I., Syafaruddin: Dynamic sign language recognition using mediapipe library and modified LSTM method. Int. J. Adv. Sci. Eng. Inf. Technol. (IJASEIT) **13**(6), 2171–2180 (2023). https://doi.org/10.18517/ijaseit.13.6.19401
13. Semeniuta, S., Severyn, A., Barth, E.: Recurrent dropout without memory loss (2016). https://doi.org/10.48550/arXiv.1603.05118
14. Konstantinovsky, T.: Introduction to the Savitzky–Golay filter: a comprehensive guide (using python) (2024). https://medium.com/pythoneers/introduction-to-the-savitzky-golay-filter-a-comprehensive-guide-using-python-b2dd07a8e2ce. Accessed 22 Oct 2025

A Comparative Study of Prompt Engineering Strategies for Data Visualization Scenarios

Camile Maria Cunha Aguiar[1(✉)], Walbert Cunha Monteiro[2], Thiago Augusto Soares de Sousa[1], Roberto Yuri da Silva Franco[3], and Bianchi Serique Meiguins[1]

[1] Federal University of Pará, Pará, Brazil
{camile.aguiar,thiago.sousa}@icen.ufpa.br, bianchi@ufpa.br
[2] Federal Institute of Pará, Pará, Brazil
walbert.monteiro@ifpa.edu.br
[3] Federal Rural University of the Amazon, Pará, Brazil
roberto.franco@ufra.edu.br

Abstract. The creation of data visualizations using Large Language Models (LLMs) depends on the quality of user prompts, which may present ambiguities when not explicitly structured. In this context, this work investigates, through quantitative analysis, how four different prompt engineering strategies: no prompt engineering, few-shot prompting, Visualization Query Language (VQL), and a hybrid approach (few-shot and VQL), influence the process of prompt construction and quality in visualization generation. A repeated-measures experiment was conducted with 16 participants to compare the prompt engineering strategies. The analysis considered efficiency, structure, completion time, prompt length, and prompt complexity metrics. The results show that although no significant differences were observed in average performance, structured approaches reduce variability and increase consistency across participants. In particular, the hybrid approach achieved the best overall performance (score = 0.858), combining higher structural completeness with lower ambiguity. These findings indicate that prompt-based effectiveness is strongly associated with the explicit representation of task structure rather than any single strategy alone, highlighting the structured support mechanisms in LLM-based systems.

Keywords: Prompt Engineering · Data Visualization · Large Language Models · User evaluation · Variability · Ambiguity

1 Introduction

The use of generative Artificial Intelligence (AI) has evolved beyond text production, also supporting the creation of visualizations [20,23]. In this context, specifying user intentions through prompts becomes an important factor for the

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 544–560, 2026.
https://doi.org/10.1007/978-3-032-30491-9_34

quality of the created visualizations [3,16,22]. In systems based on Large Language Models (LLMs), prompts are the main mechanism of interaction between the user and the model, directly influencing task and design decisions [11,21,22].

However, the interpretation of user-provided instructions may be ambiguous, especially when the task objective is not explicitly defined or when visualization requirements are not clearly specified [18]. As a result, the quality of the created visualizations may vary significantly. Although prompt engineering strategies have been proposed to mitigate this issue, there remains a gap in understanding how different approaches systematically affect prompt construction and the outcomes in data visualization tasks [8].

This work investigates the design of prompt engineering for generating data visualizations with LLMs. We evaluate how four prompt engineering approaches, prompt without engineering, prompts with few examples, Visualization Query Language (VQL), and a hybrid method (Few-shot and VQL), influence user efficiency, instruction structure, and quality of user-generated instructions, as well as the performance in the visualization generation process.

To achieve this objective, we organized our study into three core axes. The first axis, instruction structure, examines prompt elements and completeness using specific metrics. In the second construction process, we analyze user effort by measuring completion time and instruction length. The third, performance and variability, we investigate how prompt strategies differ across various visualization types and analytical tasks.

We explore four research questions: (RQ1) how do prompt engineering strategies influence efficiency; (RQ2) how do these strategies affect the structure, completeness, and complexity; (RQ3) how do approaches impact variability and consistency in participant performance; and (RQ4) how do visualization type and the nature of the analytical task influence the prompt construction process and performance.

This work offers two levels of contribution. Level one proposes a set of quantitative metrics to evaluate the prompt construction process in data visualization tasks, including completion time, prompt length, and prompt complexity, and defines a structured codebook for content analysis. Level two presents a controlled study comparing different prompt engineering strategies and provides evidence of their effects on user efficiency, consistency, and performance variability.

The remainder of this paper is organized as follows: Sect. 2 presents the concepts and prompt engineering approaches relevant to this study; Sect. 3 discusses related work; Sect. 4 describes the experiment and the adopted methodology; Sect. 5 presents the results and discussion; Sect. 6 provides final remarks; and Sect. 7 outlines the limitations and provides directions for future work.

2 Prompt Engineering Approaches

Prompt engineering has been investigated as a mechanism for guiding Large Language Models (LLMs) to perform specific tasks. Different strategies can be

used to structure the instructions provided to the model, ranging from direct approaches based solely on textual instructions to more structured methods that include examples or formal task representations. Among the approaches explored in the literature [3,17,21], the most prominent are one-shot prompting, few-shot prompting, chain-of-thought prompting, and the use of structured query languages for data visualization (VQL).

One-Shot Prompting. Involves giving the model a single instruction along with an example showing how to perform the task. This can enhance the quality of the model's responses relative to instructional prompts (zero-shot), particularly in tasks that necessitate specific formatting or output structures [3]

Few-Shot Prompting. This strategy consists of providing multiple examples as part of the model input to condition the response, thereby aligning it more closely with the task context [7]. Recent studies [7,19,20] indicate that more consistent results can be achieved when five examples are used.

Chain-of-Thought (CoT). This strategy encourages the model to explicitly generate intermediate reasoning steps before producing the final answer [16]. Studies show that CoT can improve performance in tasks involving logical reasoning and problem decomposition [9]. Although it is not the main focus of this work, this approach is relevant within the broader context of prompt engineering.

Visualization Query Language (VQL). This approach uses formal structures inspired by query languages, such as SQL, to explicitly describe visualization components [21]. In some systems, these specifications can be represented in structured formats such as JSON. In this context, the prompt is organized into instructions that define different aspects of the visualization, such as: **VISUALIZE** (type of technique), **SELECT** (variables of interest), **FROM** and **WHERE** (data source and filtering), and **GROUP BY** and **ORDER BY** (aggregation and sorting operations).

3 Related Work

This section presents studies that explore the use of prompt engineering strategies to generate and evaluate data visualizations using LLMs.

In the context of visualization generation, the work of [19] proposes LLM4Vis, an automated recommendation approach that uses few-shot prompting to leverage models' reasoning capabilities and overcome the limitations of traditional methods. Similarly, [21] investigates the use of LLMs in the NL2Vis scenario, evaluating how natural language queries can be transformed into visualizations through structured languages such as VQL.

Likewise, [24] presents LightVA, an agent-based framework that integrates multiple prompt engineering strategies, including structured templates, few-shot prompting, and self-evaluation mechanisms, focusing on visual analytics tasks. Additionally, [16] combines fine-tuning with structured logic techniques, such as Chain-of-Thought (CoT), to refine command interpretation and the creation of visualizations.

Other works investigate visualization evaluation. [5] proposes a Visual Question Response (VQA)-based framework for the automatic evaluation of visualizations created by LLMs, considering aspects of comprehension and communicative effectiveness. In contrast, [7] analyzes the use of ChatGPT as an assistant to evaluate visualization design, comparing its responses with human feedback.

Despite these advancements, existing studies primarily focus on improving model performance or automating the generation and evaluation of visualizations. However, limited attention has been given to understanding how different prompt engineering strategies influence prompt construction processes in user studies, particularly in visualization tasks.

To address this gap, this work investigates how user prompts are shaped by different prompt engineering strategies in the context of data visualization. This perspective complements existing research by focusing on the comparative evaluation of prompt engineering strategies.

4 Experiments

This section describes the experimental design, study configurations, participants, dataset, and analysis methods adopted to investigate the impact of different prompt engineering strategies on the construction of data visualizations using LLMs.

4.1 Methodology

This work adopts a repeated-measures experimental design, divided into two stages that involve data-visualization construction tasks supported by the ChatGPT Go 5.2 tool.

In the first stage of the experiment, participants were introduced to the experimental sessions, the Informed Consent Form (ICF), and the presence of a supervisor throughout the procedure, who would clarify any doubts and provide guidance at any time. Subsequently, participants received detailed instructions on the experiment's objectives, the dataset used, the adopted methodology, and the expected results.

All instructions, interactions, and prompt construction were conducted in Portuguese to ensure linguistic consistency across participants. Participants were also informed that there was no time limit for completing the experiments. Finally, they signed the ICF, formally agreeing to participate in the study.

The second stage consisted of participants performing data visualization construction and analysis tasks using the ChatGPT Go 5.2 tool. This stage was structured into four experimental sessions corresponding to the following conditions: (i) no prompt engineering, in which participants freely described prompts to construct the visualizations; (ii) few-shot prompt engineering, in which the tool was configured with 15 predefined visual examples as references; (iii) VQL-based engineering, in which participants specified prompts using a formal structure inspired by query languages; and (iv) hybrid engineering (few-shot + VQL),

which combined visual reference examples with a formal specification structure. The order of the four conditions was defined using a balanced randomization for each participant.

Sessions could be completed either continuously or at different times, depending on participant availability. For each new experimental condition, the environment configuration was previously adjusted by the experiment supervisor. Completion of each experimental condition occurred when the participant considered the result obtained through the applied prompts satisfactory relative to the expected outcomes. The second stage was completed once all four experimental conditions had been performed.

All configurations and expected results for the proposed experimental conditions are available in [1] and are further described in Sect. 4.2.

4.2 Experimental Configurations

To support the experimental conditions, four main elements were defined: (i) a dataset, (ii) visual configuration files, (iii) configuration prompts, organized into two experimental configuration sets, and (iv) a set of six reference images.

Two sets of experimental configurations were defined. The first set consists of a dataset and a configuration file, named *visualconfig*, which specifies the experiment's visual parameters, such as the color palette, typography, and visualization area dimensions (1200 × 800 pixels). This set was used in conditions that did not involve visual examples as references, specifically, the prompt without engineering and VQL-based approaches.

The second set consists of the dataset, the *visualconfig* file, and 15 additional reference images, organized into three visualization techniques, with five examples per technique. This set was used in conditions adopting the few-shot approach and the hybrid condition (few-shot + VQL).

In addition, base prompts were defined to configure the experimental environment across all conditions. These prompts specified that visualizations should be implemented in Python using the Matplotlib library, based on the provided dataset, while respecting the parameters defined in the *visualconfig* file. Furthermore, they instructed ChatGPT Go to exclusively use the information provided in the experiment, disregarding any context from previous interactions.

In the conditions adopting the few-shot approach, the prompts explicitly indicated the presence of visual reference examples, used to guide the structure and format of the generated visualizations. In contrast, in the VQL-based conditions, the prompts incorporated a formal specification structure, enabling the explicit description of visualization elements.

The reference images used as expected outputs for the three visualization techniques explored are presented in Fig. 1, which includes bar charts, line charts, and scatterplots, with two images for each technique corresponding to value comparison and extreme value identification tasks. All images were generated using ChatGPT Go 5.2. In Fig. 1 (A) value comparison by sector (CNAE) and Fig. 1 (B) distribution of operation value by BNDES sector, Fig. 1 (C) temporal evolution of operation and disbursement values and Fig. 1 (D) variation of

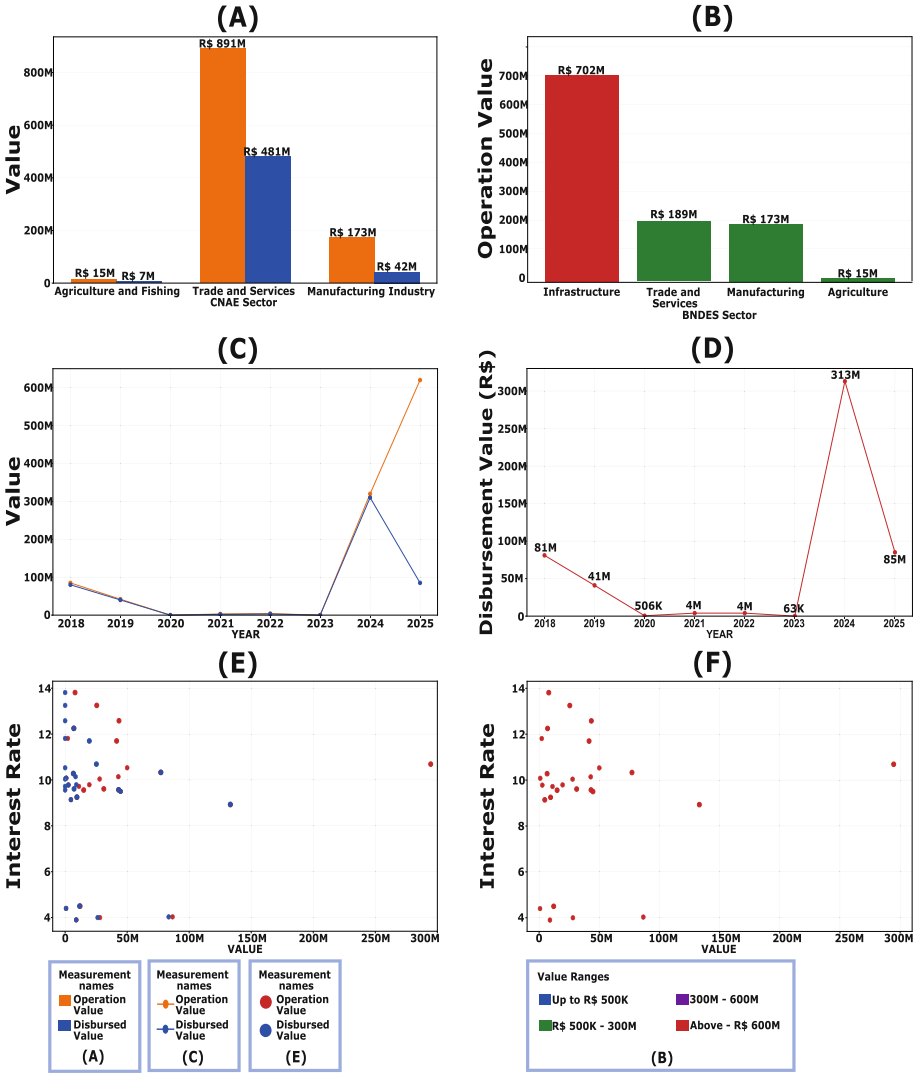


Fig. 1. Visualizations used as references during the experiment.

disbursement value over time, Fig. 1 (E) relationship between value and interest rate considering operation and disbursement values, and Fig. 1 (F) relationship between operation value and interest rate.

4.3 Participants

A total of 16 volunteers (10 male and 6 female) participated in the study, with ages ranging from 18 to 55 years (Mean = 30.5, SD = 8.9). All participants were

undergraduate or graduate students who volunteered for the study. The number of participants was defined based on experimental studies in the fields of Human–Computer Interaction and Information Visualization [13–15].

4.4 Dataset

The dataset used in this study is based on operations of the Climate Fund, which finances projects related to sustainable development, climate change, and environmental protection [2]. The dataset was released in the context of the 2025 United Nations Climate Change Conference (COP30).

The dataset contains 394 records and 19 dimensions, including 11 categorical variables (state, month, CNAE sector, client size, municipality, CNAE subsector, grouped CNAE subsector, BNDES sector, BNDES subsector, contract date, and financial institution) and 8 numerical variables (operation ID, year, municipality code, operation value, disbursed amount, interest rate, amortization period, and grace period).

4.5 Quantitative Analysis

The quantitative analysis used descriptive statistics to characterize participants' behavior in terms of effort, structure, and efficiency in prompt construction. We considered three dependent variables: Completion Time, Prompt Length, and Prompt Complexity.

Completion Time represents the total time taken to complete each task, measured from the start to the completion of prompt creation. This variable estimates the time effort involved in the prompt construction process.

Prompt Length corresponds to the size of the prompt relative to the total number of tokens, calculated using the platform [12]. This metric allows the analysis of the level of detail in the instructions and its relationship with both effort and process efficiency.

Prompt Complexity evaluates the level of instruction structuring based on the Directed Content Analysis methodology [6]. For this analysis, a category scheme was developed to reflect the fundamental requirements for proper data visualization specification, including: (i) chart type, (ii) axis mapping (x or y), (iii) aggregation operations, (iv) analytical task, (v) informative elements (titles, legends, and labels), and (vi) data formatting.

Based on these categories, we constructed a codebook with operational definitions and representative examples. The codebook definitions are presented in Table 1.

Each category was coded according to its presence or absence in the instructions, enabling the calculation of quantitative scores for structural completeness. This enabled a comparison of the level of specification between the prompts created by each participant.

Table 1. Codebook used for prompt content analysis

Category	Definition	Example
Chart type specification	Explicit mention of the type of visualization to be generated by the system.	"I want a grouped bar chart."
Axis mapping	Explicit association between dataset variables and the axes.	"The X-axis represents the year and the Y-axis represents the amount."
Data aggregation operations	Indication of aggregation operations, such as sum or average.	"Perform the summation for the BNDES sector."
Analytical task	Specification of the analytical operation, such as comparison or identification of extremes.	"Create a comparison line chart."
Informational elements	Inclusion of descriptive elements such as title, legend, axis labels, or annotations.	"Add a title and legend to the chart."
Data formatting	Specification of how data values should be presented, such as scientific notation, full values, or abbreviations.	"Use abbreviations such as 'K' for thousands and 'M' for millions."

A second evaluator conducted an independent analysis to ensure reliability. The level of agreement achieved was **83%** ($P_o = 0.830$), with a Cohen's Kappa coefficient of $\kappa = 0.71$, indicating substantial agreement [4].

In addition to the structural dimensions considered in the Prompt Complexity metric, two additional categories related to the semantic quality of instructions were included: ambiguity and redundancy. Ambiguity refers to vague or incomplete instructions that fail to clearly specify essential visualization elements. Redundancy corresponds to the unnecessary repetition of information or overly verbose descriptions that do not contribute to the structural clarity of the prompt.

5 Results and Discussion

5.1 RQ1: Efficiency in the Prompt Construction Process

To answer the first research question (RQ1), we analyzed efficiency using two variables: Completion Time and Instruction Length. To the first variable, the distribution for experimental condition, is shown in Fig. 2(A) distribution of completion time by condition, Fig. 2(B) completion time distribution with individual data points, Fig. 2(C) completion time by condition and visualization technique, and Fig. 2(D) completion time by condition and task type.

The analysis of mean completion times by condition, Fig. 2(A), was similar across the four conditions analyzed, ranging from 307.27 s (VQL) to 370.95 s

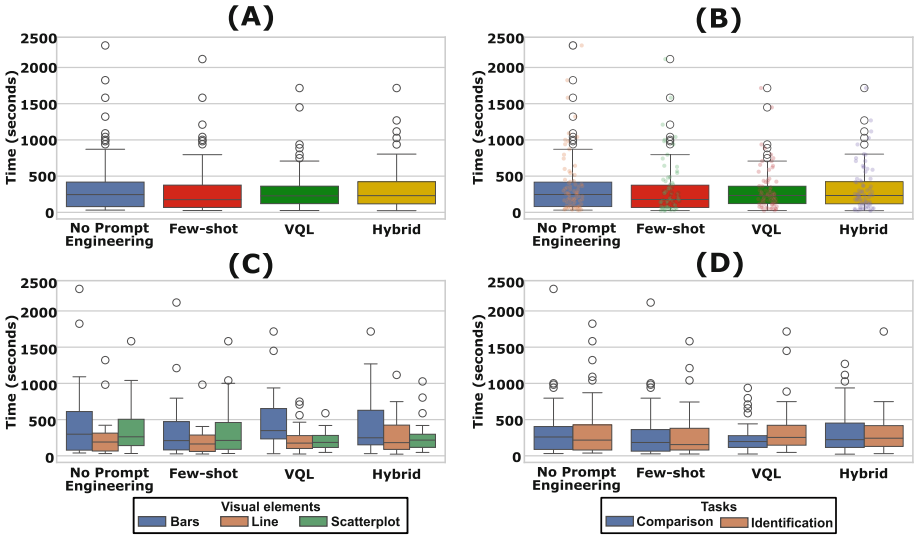


Fig. 2. Distribution of completion time variable by experimental condition.

(few-shot). The Kruskal-Wallis test was applied to evaluate the medians of the groups and did not identify statistically significant differences ($H = 1.88$,p = 0.597). This result suggests that the choice of engineering strategy does not significantly affect the mean time to develop instructions.

For the distribution of a participant's completion time, Fig. 2(B), the values are concentrated up to 1000 s across all conditions. For the no prompt engineering condition, there is greater dispersion of points, followed by the few-shot and hybrid conditions, while the VQL condition shows greater clustering. The largest outliers were evaluated in the no prompt engineering condition.

In Fig. 2(C), the completion time by condition for the visualization technique, each point corresponds to an individual participant's execution time under a specific prompting technique. For no prompt engineering, the largest distribution was shown for bars chart and scatterplot, the line chart shows a more concentrated distribution.

In the few-shot condition, the largest distribution was for the scatterplot technique, followed by the bar chart. The Line showed the smallest distribution. For the VQL condition, the largest distribution was for the bar chart. The line chart and scatterplot showed the smallest distribution. In the hybrid Condition, the largest distribution was for the scatterplot technique, followed by the line chart, and the scatterplot showed the smallest distribution.

For the distribution of task types comparison and identification, Fig. 2(D), the values are concentrated below 1000 s across all conditions, although outliers extend this range. For no prompt engineering, there is the largest dispersion for the identification task. The few-shot condition showed slight dispersion in task

comparison. The VQL condition showed a wider spread in the Identification task, and the hybrid Condition showed the largest distribution in the comparison task.

Based on the results, it is possible to understand that the prompt engineering strategy does not affect the mean instruction construction time, although small variations may occur across conditions. Regarding the dispersion of participant data, no prompt engineering is the least consistent (highest dispersion), while VQL is the most stable.

Considering visualization techniques, the scatterplot is the most unstable, and the line chart is the most consistent, regardless of the engineering technique. It is important to note that the VQL technique reduced variability but did not eliminate the effect of the visualization technique. However, the VQL technique showed improved performance stability, especially in the comparison task.

5.2 RQ2: Prompt Structure and Complexity

The analysis of prompt structure was conducted using the Prompt Complexity variable, through the Directed Content Analysis methodology, based on the codebook presented in Table 1, Sect. 4.5. Figure 3 shows the distribution of the analysis of prompt complexity, score, across different conditions: No Prompting Engineering, Few-shot, VQL, and Hybrid, visualization types, and tasks (Comparison (C) and Identification (I)). Figure 3 (A) Performance comparing prompt strategies, Fig. 3 (B) Performance by task, for bar chart, Fig. 3 (C) Performance by task, for line chart, Fig. 3 (D) Performance by task, for scatterplot.

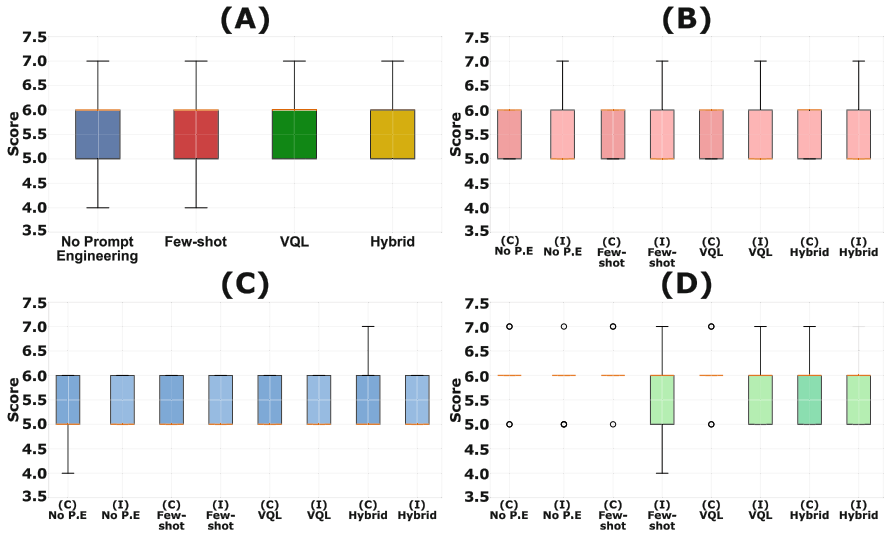


Fig. 3. Performance comparison between prompt strategies

The results for prompt complexity, showed in Fig. 3 (A), exhibited limited variation across conditions, with mean values ranging from 5.5 to 6.0. This pat-

tern suggests that, for all conditions, participants produced descriptions with similar levels of structure. The 1st quartile was flattened in the VQL and hybrid conditions, indicating that at least 50% of the data lie at or above the median.

In the bar chart, stimulus complexity, Fig. 3 (B), showed stable performance scores between the comparison (C) and identification (I) tasks, for the four conditions analyzed. However, the comparison (C) task was more stable than the identification (I) task in all conditions. This condition is characterized by the absence of the 1st and 3rd quartiles in the comparison (C) tasks in all cases, whereas, for the identification task, only the 1st quartile was flattened, remaining very close to the median.

In the line chart, Fig. 3 (C), prompt complexity showed a stable performance between tasks comparison (C) and identification (I). For the task comparison in conditions: No prompting engineering and hybrid, the 3rd and 1st quartiles were flattened, respectively.

For the scatter plot, prompt complexity, Fig. 3 (D), the tasks: comparison (C) and identification (I), for the conditions: No prompt engineering, few-shot and VQL are the same value (6.0), some outliers showed scores of 5.0 or 7.0, for the prompt complexity (score). The Identification (I) task in the few-shot condition was the most unstable condition, with scores ranging from 4.0 to 7.0. Furthermore, the Identification Task for the VQL and hybrid condition, and the Comparison for the hybrid Condition, showed the absence of the 1st quartile.

In general, the complexity of the prompts remains stable across all conditions, with limited variations related to visualization technique and task type. The greatest variability was observed in the scatterplot technique and was influenced by comparison (C) and identification (I) tasks associated with no prompt Engineering condition, suggesting that complexity is also influenced by cognitive factors in prompt construction.

5.3 RQ3: Performance Variability and Consistency

Performance variability was analyzed using dispersion measures, including standard deviation (SD), interquartile range (IQR), and maximum value (Max), considering the Completion Time variable. The Table 2 presents the main descriptive measures of variability by experimental condition.

Table 2. Variability measures of Completion Time by experimental condition.

Condition	Mean Time(s)	Standard Deviation	IQR	Max
No Prompt Engineering	370.95	426.26	338.75	2304
Few-shot	309.67	370.76	308.00	2116
VQL	307.27	298.51	241.50	1716
Hybrid	325.78	317.17	305.50	1716

The results reveal relevant differences in data dispersion across conditions. The no prompt engineering condition exhibited the highest variability, with a high standard deviation (426.26), the largest interquartile range (IQR = 338.75), and the highest observed maximum value (2304 s), indicating a strong influence of outliers and high inconsistency across participants.

The few-shot prompting showed increased variability (SD = 370.76; IQR = 308.00) in relation to VQL, demonstrating more consistent performance. Hybrid engineering showed moderate variability, thus achieving intermediate behavior.

This suggests that even if prompt engineering does not have a strong impact on completion time (RQ1), it does influence performance consistency. In this sense, structured engineering, in this case VQL and hybrid approaches, can promote more consistent results among participants and reduce potential uncertainties.

This analysis complements the results from RQ1 and RQ2, suggesting diversity in the prompt construction process, in terms of prompt complexity and prompt time, there were no significant differences between the methodologies. As shown in Fig. 2 (Sect. 5.1), there is substantial dispersion in execution times across all experimental conditions, with the presence of outliers and wide variation among participants.

5.4 RQ4: Influence of Visualization Technique and Analytical Task

The cross-analysis between visualization techniques (bar charts, line charts, and scatterplots) and analytical tasks (comparison and identification) revealed distinct effects on participant performance and prompt structure.

Regarding visualization techniques, we observed significant variation in completion time across different chart types. As shown in Table 3, bar charts exhibited the highest mean completion time (440.61 s) and the greatest dispersion (SD = 456.44), being significantly slower than line charts and scatterplots.

Table 3. Mean prompt completion time by visualization technique

Technique	Mean Time(s)	Standard Deviation
Bar Charts	440.61	456.44
Line Charts	243.72	243.41
Scatterplots	300.92	306.75

These results suggest that bar charts require greater effort during prompt construction, possibly because they require explicit specification of multiple categories and relationships between discrete variables.

To better investigate this interpretation, a multidimensional analysis was conducted. Figure 4 shows a heatmap of normalized metrics by experimental condition and variables: mean completion time (a), coefficient of variation of

completion time (b), mean prompt length (c), coefficient of variation of prompt length (d), axis mapping (e), aggregation (f), ambiguity (g), and redundancy (h).

Fig. 4. Synthesis of dimensions by experimental condition.

The analysis used a normalized quantitative metrics derived from three main groups: (i) experimental performance metrics, such as mean completion time and coefficient of variation of completion time (RQ1); (ii) structural prompt metrics, such as mean prompt length and coefficient of variation of prompt length (RQ2); and (iii) qualitative metrics related to prompt content, including axis mapping, aggregation, ambiguity, and redundancy.

The results indicate that structured approaches, such as VQL and hybrid, present more balanced profiles, with higher levels of axis mapping and aggregation combined with lower levels of ambiguity and redundancy. In contrast, the no prompt engineering condition shows higher execution time, greater ambiguity, and lower structural quality of prompts, whereas the few-shot approach exhibits high variability, indicating lower consistency.

This pattern highlights that structured methodologies concentrate higher values in quality-related metrics (axis mapping and aggregation) and lower levels of ambiguity and redundancy. The hybrid methodology shows the best overall performance, while VQL combines strong structural quality with greater temporal efficiency. In contrast, the no prompt engineering condition exhibits high ambiguity and low structural performance, whereas the few-shot approach shows higher redundancy and variability.

Regarding analytical tasks, the Mann–Whitney test [10] did not indicate a statistically significant difference between comparison and identification tasks ($U = 11559.5$, $p = 0.445$). This result suggests that there is insufficient evidence to conclude that task type influences participant performance. In contrast, the observed effects across visualization techniques indicate that these constitute a more decisive factor in prompt construction effort.

The integrated analysis of the results from research questions RQ1, RQ2, and RQ4 indicated that, although the visualization technique influences the effort

required to construct the prompts, the final performance is mainly determined by the level of prompt structuring. In particular, structured engineering reduces ambiguity, increases consistency, and mitigates the negative effects associated with techniques such as bar charts.

When comparing prompt engineering methods, we observed that hybrid engineering and VQL performed best (scores of 0.858 and 0.739, respectively), combined with high axis mapping rates, greater use of aggregation, and lower redundancy, demonstrating the stability that a structured engineering approach generates. In contrast, the zero-shot (0.331) and few-shot (0.213) methods performed worst, with greater variability and lower structural quality.

Therefore, we observed that the performance of the prompts was related to the complexity of the visualization technique used and the level of structuring of the prompts.

5.5 Design Approach for Prompt Engineering in LLM-Based Data Visualization

Based on the findings of this study, we derived a set of practical approaches to support the design and use of prompt engineering strategies for LLMs performing data visualization tasks.

Structured prompt-based engineering tools like VQL should be prioritized when the output demands precision and strict control over visualization specifications. These specifications include the need for multiple variables, aggregation within the visualization, and other factors. Furthermore, we believe VQL is best suited for expert users who require reproducibility and consistency.

The few-shot approach, on the other hand, is more suitable for beginner users because it provides visual examples, helping to establish patterns in expected output formats and reducing a possible initial cognitive barrier. However, since this engineering is based only on implicit guidance, it cannot be used alone. Thus, it must be used in a hybrid manner for tasks requiring high rigor.

Structured prompts are a type of engineering that places the entire responsibility for structuring the task on the user, increasing cognitive load and the likelihood of incomplete or ambiguous prompts. Therefore, it should be applied in scenarios that involve datasets with different structures, multiple variables, or specific analytical requirements.

The results show that using unstructured prompts produces the greatest ambiguity, the highest variability, and the lowest overall performance. While acceptable for exploratory tasks, it is not suitable for the reliable generation of visualizations.

Hybrid engineering, combining the VQL structure and visual examples from a few-shot examples, provided the necessary balance to assist participants, enabling them to achieve the best overall performance in the study, as observed. Therefore, this engineering approach is suitable for user groups with varying levels of experience.

These results indicate that there is no prompt engineering strategy that is most useful. The effectiveness depends on the alignment across task complex-

ity, user experience, and the level of structural support provided. Among the approaches evaluated, the hybrid strategy emerges as a robust solution, combining structured specifications with example-based guidance to mitigate limitations.

6 Final Remarks

The results indicate that prompt engineering does not affect average user performance but does influence performance variability. Engineering strategies, such as VQL and hybrid conditions, were more effective. This result may be related to the specification of structural elements, such as axis mapping and data aggregation.

Furthermore, we found that bar charts and scatterplots require more structural specifications than line charts, influencing the level of precision needed in the instructions.

Although the quality of the resulting visualization techniques was not evaluated, the findings suggest that prompt engineering with clearer structures and less ambiguity could yield more consistent results. This interpretation should be considered with caution. These findings reinforce the importance of structuring content visualization and task categories in the prompts.

7 Limitations and Future Work

This study presents some limitations that should be considered when interpreting the results. The number of participants selected was consistent with some studies analyzed. Furthermore, the sample consisted only of students, which does not reflect the behavior of experts in the field.

Another limitation lies in the number of tasks and types of visualization. We conducted the experiment using a single language model, which may have generated the observed patterns. Different language models may exhibit distinct behaviors in the interpretation of instructions, in the structuring of responses, and in sensitivity to linguistic variations.

This study offers opportunities for future research, including qualitative analyses of the prompts constructed by participants, expanding the sample, diversifying tasks and visualization techniques, and including participants with different experience levels.

Furthermore, comparative studies across different LLMs to understand how prompt engineering approach can be applied to achieve better outputs. This support can reduce ambiguity and improve the quality of user-created instructions.

Acknowledgments. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES), Finance Code 001. The authors also acknowledge the support provided by the Dean of Research and Graduate Studies (PROPESP) of the Federal University of Pará (UFPA) through the PIAPA 2026 grant.

References

1. Aguiar, C.: Prompt-engineering-vis-data: Data for prompt engineering in AI-based data visualization (2025). https://github.com/AguiarCamile/prompt-engineering-vis-data. Accessed 14 Mar 2026
2. BNDES: BNDES data - dados e séries temporais (2025). https://www.bndes.gov.br/wps/portal/site/home/bndes-data/bndes-data-dados-series-temporais/. Accessed 14 Mar 2025
3. Chen, N., Zhang, Y., Xu, J., Ren, K., Yang, Y.: VisEval: a benchmark for data visualization in the era of large language models (2025). https://doi.org/10.1109/TVCG.2024.3456320
4. Cohen, J.: Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. Psychol. Bull. **70**(4), 213 (1968). https://doi.org/10.1037/h0026256
5. Ford, J., Zhao, X., Schumacher, D., Rios, A.: Charting the future: using chart question-answering for scalable evaluation of LLM-driven data visualizations. In: Proceedings of the 31st International Conference on Computational Linguistics, pp. 7497–7510 (2025). https://aclanthology.org/2025.coling-main.501/
6. Hsieh, H.F., Shannon, S.E.: Three approaches to qualitative content analysis. Qual. Health Res. **15**(9), 1277–1288 (2005). https://doi.org/10.1177/1049732305276687
7. Kim, N.W., Ahn, Y., Myers, G., Bach, B.: How good is ChatGPT in giving advice on your visualization design? ACM Trans. Compu.-Hum. Interact. **32**(5), 1–33 (2025). https://doi.org/10.1145/3745768
8. Li, G., et al.: Visualization generation with large language models: an evaluation. arXiv preprint arXiv:2401.11255 (2024), https://arxiv.org/abs/2401.11255
9. Ma, X., Wan, G., Yu, R., Fang, G., Wang, X.: CoT-valve: length-compressible chain-of-thought tuning. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 6025–6035 (2025). https://doi.org/10.18653/v1/2025.acl-long.300
10. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. Ann. Math. Stat. 50–60 (1947). https://doi.org/10.1214/aoms/1177730491
11. Monteiro, W.C., Dos Santos, D.H., De Sousa, T.A.S., Queiroz, V.F., De Araújo, T.D.O., Meiguins, B.S.: Workload evaluation to create data visualization using ChatGPT. In: 2023 27th International Conference Information Visualisation (IV), pp. 136–141 (2023). https://doi.org/10.1109/IV60283.2023.00032
12. OpenAI: OpenAI tokenizer (n.d.). https://platform.openai.com/tokenizer. Accessed 14 Mar 2026
13. Qiu, T., et al.: SmartMLVs: LLM-enabled multiple linked views generation for interactive visualization. In: 2025 IEEE 18th Pacific Visualization Conference (PacificVis), pp. 58–68 (2025). https://doi.org/10.1109/PacificVis64226.2025.00012
14. Shi, D., Xu, X., Sun, F., Shi, Y., Cao, N.: Calliope: automatic visual data story generation from a spreadsheet. IEEE Trans. Vis. Comput. Graph. **27**(2), 453–463 (2020). https://doi.org/10.1109/TVCG.2020.3030403
15. Siu, A., Kim, S.-H.G., O'Modhrain, S., Follmer, S.: Supporting accessible data visualization through audio data narratives. In: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems. CHI '22. Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3491102.3517678

16. Tian, Y., et al.: ChartGPT: leveraging LLMs to generate charts from abstract natural language. IEEE Trans. Vis. Comput. Graphics **31**(3), 1731–1745 (2024). https://doi.org/10.1109/TVCG.2024.3368621
17. Uusnäkki[1], J., Ihantola, P., Mikkonen, T.: Exploring prompt engineering with large language models. In: Arai, K. (ed.) CompCom 2025. LNNS, vol. 1423, pp. 166–177. Springer, Cham (2025). https://doi.org/10.1007/978-3-031-92602-0_10
18. Wang, C., Lee, B., Drucker, S.M., Marshall, D., Gao, J.: Data formulator 2: iterative creation of data visualizations, with AI transforming data along the way. In: Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems. CHI '25. Association for Computing Machinery, New York, NY, USA (2025). https://doi.org/10.1145/3706598.3713296
19. Wang, L., Zhang, S., Wang, Y., Lim, E.P., Wang, Y.: LLM4Vis: explainable visualization recommendation using ChatGPT. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, pp. 675–692 (2023). https://doi.org/10.18653/v1/2023.emnlp-industry.64
20. Wang, Q., Zhu-Tian, C., Wang, Y., Qu, H.: A survey on ML4VIS: applying machine learning advances to data visualization. IEEE Trans. Vis. Comput. Graph. **28**(12), 5134–5153 (2022). https://doi.org/10.1109/tvcg.2021.3106142
21. Wu, Y., et al.: Automated data visualization from natural language via large language models: an exploratory study. Proc. ACM Manag. Data **2**(3), 1–28 (2024). https://doi.org/10.1145/3654992
22. Yu, H.: Leveraging prompt engineering to facilitate AI-driven chart generation. In: Proceedings of the 2024 6th International Conference on Computer Information Science and Artificial Intelligence. CISAI '24, pp. 261–268. Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3703187.3703231
23. Zhang, W., et al.: Natural language interfaces for tabular data querying and visualization: a survey. IEEE Trans. Knowl. Data Eng. **36**(11), 6699–6718 (2024). https://doi.org/10.1109/TKDE.2024.3400824
24. Zhao, Y., et al.: LightVA: lightweight visual analytics with LLM agent-based task planning and execution. IEEE Trans. Vis. Comput. Graph. **31**(9), 6162–6177 (2024). https://doi.org/10.1109/TVCG.2024.3496112

Detection of Cow Face and Subregions Using a New Annotated Dataset

Mathews Edwirds Gomes Almeida[1], Pedro de Araújo Bhering Bittencourt[1], Brian Luís Coimbra Maia[1], Lucas Silva Santana[1], João Vítor de Castro Martins Ferreira Nogueira[1], Daniel Muller Rezende[1], Gabriel Rezende da Silva[1], Thais de Souza Marins[1], Luiz Maurílio da Silva Maciel[1(✉)], Marcelo Bernardes Vieira[1], Saulo Moraes Villela[1], and Bruno Campos de Carvalho[2]

[1] Federal University of Juiz de Fora, Juiz de Fora, MG, Brazil
{mathews.edwirds,pedro.bittencourt,brian.maia,lucas.santana,joao.nogueira,daniel.rezende,rezende.gabriel,thais.marins}@estudante.ufjf.br,
{luiz.maciel,marcelo.bernardes,saulo.moraes}@ufjf.br
[2] Embrapa Dairy Cattle,Juiz de Fora, MG, Brazil
bruno.carvalho@embrapa.br

Abstract. This study proposes a novel methodology for detecting cattle faces and their key facial subregions – ears, eyes, and muzzle – through a two-stage object detection pipeline based on deep convolutional neural networks. To address the lack of annotated datasets in this domain, we introduce the Annotated Cattle Images (ACI) dataset, composed of 7,146 images and 54,111 labeled instances, collected from both public repositories and real-world farming environments. This dataset was essential for training our detection models, and we made it publicly available for future research. Our detection approach employs a two-stage detection strategy: initially locating the bovine face in the entire image and subsequently detecting subregions within the localized area, effectively reducing background noise and improving accuracy for small features. Experiments using YOLOv8-X, YOLOv9-E, and YOLOv10-X show that the YOLOv9-E model achieved the best performance. Comparative analyses confirm the advantage of the two-stage approach, which improved subregion detection over the single-stage baseline. We also implemented a dedicated post-processing module to remove duplicate predictions and further enhance the reliability of the models. These results show the viability of the proposed solution for real-time, non-invasive livestock monitoring, advancing the applicability of computer vision in precision livestock farming.

Keywords: Object Detection · Dataset · Computer Vision · Precision Livestock Farming

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 561–579, 2026.
https://doi.org/10.1007/978-3-032-30491-9_35

1 Introduction

Livestock farming constitutes a pivotal component of the global agricultural economy, serving as a primary source of food and animal-derived raw materials. Globally, livestock farming is a key element of economic development, with the cattle herd reaching approximately 1.5 billion in recent years [1]. The expansion of cattle herds reflects the growing global appetite for meat and animal products, despite ongoing challenges related to environmental sustainability. This emphasizes the critical need to enhance the efficiency and sustainability of livestock production, especially in the face of growing environmental challenges and increasing social demands for ethical animal husbandry practices.

In response to these ecological and livestock welfare concerns, the concept of precision livestock farming has gained traction as a transformative approach. This approach integrates advanced Information and Communication Technologies (ICTs) to optimize livestock production through the continuous and detailed monitoring of animals health, welfare, and environmental conditions. The deployment of sensors, cameras, and mobile technologies facilitates real-time data acquisition and analysis, thereby enabling data-driven decision-making to enhance productivity and sustainability [2].

In the context of precision livestock farming, computer vision allows to recognize and analyze patterns in images or videos. This technology offers a wide range of applications, from animal detection to monitoring livestock health and welfare. Specifically, object detection has shown substantial potential in identifying and classifying specific animal features, such as the location of the face, eyes, ears, and muzzle. These detections are particularly valuable for developing automated monitoring systems that operate in real time, reducing the need for human intervention and increasing the accuracy of animal health assessments. For instance, analyzing the position and movement of bovine ears, eyes, and muzzles can provide valuable insights into their emotional states and levels of discomfort, facilitating early intervention and enhanced welfare management [3].

Despite numerous fields have being benefited from extensive research and abundant datasets, the detection of facial subregions in cattle presents significant challenges due to the limited availability of high-quality annotated datasets in the literature. The scarcity of representative datasets that encompass diverse environments, variability in animal characteristics, and varying image capture conditions makes the development of effective precision livestock systems in uncontrolled settings difficult [4]. This disparity underscores the need for specialized resources tailored to livestock. To address these challenges and improve model performance in varied conditions, one of the main contributions of this study is the creation of a dataset with annotations dedicated to detecting the face and its facial subregions of interest in cattle images.

The present study aims to develop object detection models tailored to identify key facial subregions in cattle within non-standardized, real-world environments. Thus, our proposal explores a two-stage detection approach: we first detect the cattle faces and then detect the subregions, which effectively eliminates background noise and improves model results. We employed our proposed dataset

to train these models to deliver precise and reliable detection performance even under varying conditions of illumination, angle, and background complexity. The integration of such models into real-time monitoring systems has the potential to significantly enhance the assessment of animal health and welfare in precision livestock farming, thereby contributing to the broader objectives of improving productivity, sustainability, and animal well-being in the sector.

2 Related Work

In recent years, some studies have addressed the problem of facial subregions detection in animals for different applications. Billah et al. [5] conducted a facial subregions detection process in goats using convolutional neural networks (CNNs) to improve and facilitate the animal management. Furthermore, the work of Jaddoa et al. [6] proposed a method to monitor the body temperature of cattle to identify certain risks to the animal. They analyze the animal's body temperature by identifying the eye region in infrared thermal images using the Infrared Thermography Technique (IRT). However, we note a scarcity of works addressing facial subregions detection. Particularly, to the best of our knowledge, we did not find works that propose methods to detect the facial subregions.

Some works propose to detect the entire animal face, without separating the subregions. A primary objective in many studies involving face detection is the recognition of individuals [7–9]. Kawagoe et al. [10], in addition to individual recognition, conducted a feeding time estimation. Another common motivation is facial expression recognition [11]. Mao and Liu [11] focused on pet dogs, in contrast to other related works that studied livestock animals. The YOLO series algorithms and their variations are the most common methods for solving face detection problems.

The detection of other parts of the animal's body for individual identification has also been studied in several works. Tassinari et al. [12] conducted a real-time identification of cows based on coat pattern, to monitor the position, actions, and movements of each cow. Smink et al. [13] focused on detecting, tracking, and reading ear tags in videos to identify animals with diseases on large-scale cattle farms. Some studies conducted the detection of the animal's entire body. For instance, Gupta et al. [14] detected and classified cows into eight different breeds.

Despite the related works presented, in this study, we introduce a methodology for detecting faces and subregions (ears, eyes, and muzzle) in cattle. We propose an annotated dataset specifically for this task, which was used to train and evaluate the performance of detection models.

3 Deep Convolutional Neural Networks for Object Detection

Modern object detection methods are predominantly based on CNNs. These approaches follow either a two-stage strategy – where region proposals are generated before classification – or a one-stage strategy, which performs detection

and classification in a single step. One-stage detectors, despite a slight reduction in accuracy, offer significantly better computational performance and are thus more suitable for real-time applications [15]. In this work, we adopted one-stage models for detecting cattle facial regions, with a focus on different versions of the You Only Look Once (YOLO) architecture.

YOLO was introduced by Redmon [16] as a unified, real-time object detection framework. Since then, successive versions have improved performance, robustness, and usability. YOLO's core principle is to treat detection as a regression problem: it divides the input image into an $S \times S$ grid, where each cell predicts bounding boxes, class probabilities, and objectness scores. The original architecture comprised 24 convolutional layers and 2 fully connected layers, using 1×1 convolutions to reduce feature dimensions. Outputs are refined using Non-Maximum Suppression (NMS) to eliminate redundant predictions.

Recent YOLO versions have adopted significant structural and algorithmic innovations to enhance detection accuracy and computational efficiency. YOLOv8, released by Ultralytics in 2023, adopts an anchor-free approach for bounding box prediction and integrates techniques like mosaic augmentation and an improved neck for better feature fusion. This version introduced a suite of pre-trained models with varying sizes to balance accuracy and speed. YOLOv9, proposed in 2024 by an independent team, incorporates Programmable Gradient Information (PGI) and Generalized Efficient Layer Aggregation Network (GELAN), addressing the information bottleneck and improving gradient flow. These features enhance generalization, particularly in low-data regimes and multitask scenarios. YOLOv10, released in mid-2024, eliminates the need for NMS via consistent double assignments and leverages large-kernel convolutions and partial self-attention to improve accuracy with minimal computational cost [17].

In addition to the different versions of YOLO, there are variations in the sizes of each model based on the parameter count and complexity. In our experiments, we employed the extra-large version (X) of YOLOv8, which has 68.2 million parameters. We also use the extended version (E) of YOLOv9, with 58.1 million parameters and the extra-large version (X) of YOLOv10, with 29.5 million parameters. These are the biggest versions of each network available. We intend to evaluate the performance of our detection approach using different network versions.

4 ACI Dataset

Due to the lack of available datasets including bovine facial subregions, we created a new dataset to meet the specific needs of this study, named "Annotated Cattle Images" (ACI) dataset. Its construction involved several critical steps: data collection, filtering, and labeling. Data collection employed specific techniques to ensure the dataset's diversity and representativeness. Filtering was applied to remove irrelevant and low-quality data. Labeling was essential for identifying and categorizing key features, such as facial subregions, to train and evaluate models effectively. These steps were crucial for building a reliable dataset for bovine facial detection models.

To support further research and development in the area, we made the training dataset publicly available at Almeida et al. [18]. This provision allows multiple studies and institutions to utilize the data for further research and model development. In the following sections, we describe the steps performed to create the proposed dataset.

4.1 Data Collection

To develop a dataset for advancing bovine detection and classification methodologies, we implemented a data collection strategy that combined curated images from the internet with captured images. Initially, we employed a tool to assist in downloading a diverse batch of bovine images from selected open-access platforms, including Pexels,[1] Pixabay,[2] Creative Commons' Openverse,[3] and Unsplash.[4] These platforms align with our non-commercial research objectives and guarantee adherence to ethical guidelines concerning copyrighted materials. We meticulously filtered the images to include only those explicitly designated as public domain or available under licenses that do not require specific attribution to the original authors.

To capture a broad range of bovine breeds, poses, and environmental contexts, we utilized a variety of search terms in Portuguese (e.g., *vaca*, *bezerro*, *novilho*, *boi*, *touro*, *bovino*, *gado*), Spanish (e.g., *vaca*, *becerro*, *buey*, *toro*, *bovino*, *vacuno*, *ganado*), and English (e.g., *cow*, *calf*, *bull*, *taurus*, *bovine*, *cattle*, *livestock*). This multilingual approach enriched the diversity of our dataset by incorporating images from various regional sources.

Furthermore, manual curation was an integral part of our data collection process, as we did not use any scripts or web scraping libraries to filter the images before downloading them from open-access websites. Given that many downloaded images lacked complete metadata, manual verification was essential to assess their suitability for inclusion in our dataset.

After compiling the initial set of images, we performed a post-filtering process to improve the general consistency of the dataset. Firstly, we eliminated images in which the bovine's face occupied less than 10% of the image area, ensuring that the faces were sufficiently prominent for facial analysis. In addition, we removed images that were excessively blurry, distorted, or had severe lighting problems, as these challenges are not in the scope of our work. Images that did not contain bovines or showed significant occlusions were also excluded to maintain the relevance and reliability of the dataset.

Finally, we complemented our dataset through direct image capture conducted in collaboration with Embrapa Dairy Cattle. This approach involved photographing bovines in their natural farm environments, providing practical and realistic examples of appearances under real-world conditions. By incorporating images of diverse breeds, poses, and environmental conditions through

[1] https://www.pexels.com.
[2] https://www.pixabay.com.
[3] https://openverse.org.
[4] https://www.unsplash.com.

direct capture, we significantly improved the dataset's applicability for detection systems and practical deployment scenarios.

4.2 Annotation

An annotation process was implemented to label bovine facial subregions meticulously. This methodical approach aimed to achieve both precision and consistency in labeling. The annotation adhered to the following principles:

1. **Bounding Box Accuracy**: Bounding boxes were meticulously adjusted to fit tightly around each object of interest, ensuring central placement and minimizing background inclusion to enhance the focus on relevant facial subregions and features.
2. **Labeling Key Regions**: Each subregion of interest was tagged with specific labels such as "cow" for the bovine face, "left ear," "right ear," "left eye," "right eye," and "muzzle" to accurately categorize facial components.
3. **Perspective Consideration**: Labels were assigned carefully to the animal's perspective, ensuring that "left" and "right" features were represented correctly based on the animal's viewpoint.
4. **Partial Occlusions**: All visible regions of interest were annotated, including those partially occluded.
5. **Special Cases**: In instances where we observed specific features, such as closed eyes or an open jaw, annotations were made approximately around the region.
6. **Image Quality Considerations**: Instances where objects were difficult to recognize due to blurriness or small dimensions were ignored.
7. **Avoiding Mislabeling**: We avoided overly tight bounding boxes around features like eyes, which could lead to confusion with background patterns. We adjusted the boxes to ensure a clear distinction from the surrounding area.

The annotation process was performed using LabelImg (version 1.8.6),[5] a tool that offers an intuitive interface and supports various output formats, such as PASCAL VOC and YOLO. The tool's features, including simultaneous image and label visualization, facilitated error detection and correction.

4.3 Dataset Overview and Statistics

The ACI dataset consists of bovine facial images collected from multiple sources, ensuring a diverse representation of breeds, poses, and environmental conditions. However, the images exhibit a wide range of resolutions, with many occupying excessive storage space. This variability not only increased storage demands but also slowed down the preprocessing stage before model training.

[5] https://github.com/heartexlabs/labelImg.

To address this, we applied a dataset adaptation strategy aimed at getting image sizes more compatible with deep learning models while preserving annotation integrity and image diversity. As part of this strategy, we established a target resolution A_{target} of 15 megapixels and resized the images that exceeds this threshold. Specifically, given an image with area $A = H \times W$ (in pixels), if $A > A_{\text{target}}$, the scaling factor $s = \sqrt{A_{\text{target}}/A}$ was applied uniformly to both dimensions. This ensures that the resized image approximates the target resolution while preserving the original aspect ratio. This dynamic resizing approach effectively reduced storage demands while maintaining visual and annotation consistency. Importantly, this resizing procedure was applied exclusively to the training set, leaving the test set untouched.

Although very high-resolution images do not negatively impact training, they also do not provide significant benefits for standard object detection tasks. All images were resized to align with the typical input dimensions expected by YOLO and other modern detection architectures, which commonly operate on relatively low-resolution inputs (e.g., 640×640). This resizing process was not primarily intended to reduce storage usage – although that became a positive side effect – but rather to ensure architectural compatibility and consistency throughout the dataset. Importantly, this adjustment did not alter the dataset's distribution or introduce any biases that could affect model performance. In fact, most detection models in literature use input sizes smaller than those resulting from our resizing procedure, and only a small fraction of the original images had very high resolution. By implementing this adjustment, we obtained lower storage space and computational feasibility.

The dataset final structure is summarized in Table 1, which presents key attributes of the training and test sets, including the number of images, resolution details, and overall dataset size. Specifically, images sourced from the internet were used for the training set, while images collected by our team were designated for the test set. Notably, the training images were sourced from the internet without prior labeling, making it challenging to determine the exact number of individuals and breeds within this subset. Additionally, the average resolution of images in the training set is lower than in the test set. The image distribution between the sets is approximately 86.51% for training and 13.49% for testing.

The analysis of the images from both the training and test sets reveals significant variability in many aspects, including proximity, angle, background, lighting, and sharpness. This variability is crucial for understanding the dataset's scope and challenges. For instance, the presence of occlusions, multiple faces, diverse categories (such as calf, cow, or other bovine classifications), poses, and breeds adds complexity to detecting cow facial subregions. Moreover, some images feature multiple facial subregions, even when the bovine face is not fully visible.

The diversity of environments in which the images were captured further highlights the dataset's variability. The training set includes bovines captured in various scenarios, such as urban areas, which contrasts with the test set, which is

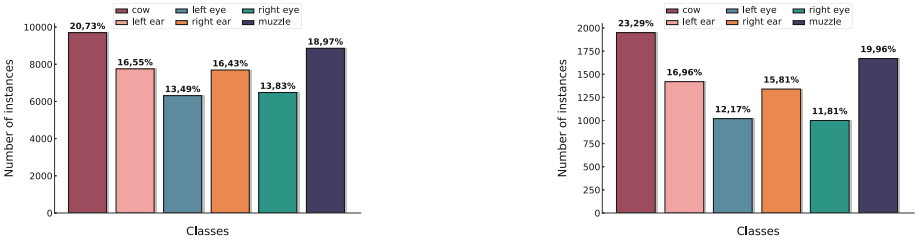
Table 1. Detailed information on the training and test sets.

Attribute	Training set	Test set
Number of Images	6182	964
Minimum Resolution	136 × 235	2974 × 2918
Maximum Resolution	7360 × 4912	4624 × 3468
Average Resolution	1982 × 1694	4317 × 3243
Minimum Size (MB)	0.009	1.82
Maximum Size (MB)	9.77	25.72
Average Size (MB)	1.24	13.33
Total Size (GB)	7.45	12.5
Number of Individuals	-	165
Breeds	-	Girolando and Holstein
Colored	Yes	Yes
Format	JPG	JPG

composed exclusively of images from rural production environments, including pens, calving areas, pastures, and confinement systems. This distinction is visually represented in Fig. 1, which showcases representative examples from each dataset.

Fig. 1. Examples of images from the proposed dataset. Fist row: training set. Second row: test set.

Figure 2 presents the class distribution of the annotated objects in both the training and test sets. Larger facial regions, such as the full cow head and the muzzle, dominate the distribution. In contrast, smaller object classes, particularly ears and eyes, are consistently underrepresented—a disparity likely caused by the inherent difficulty in annotating small-scale features and by occlusions. This imbalance not only affects annotation efforts but may also hinder the detection performance for these classes.



(a) Training set. (b) Test set.

Fig. 2. Distribution of labels by class and by set.

5 Proposed Detection Method

We used the proposed dataset to train a system that consists of two detection models for distinct purposes, as illustrated in Fig. 3. The first model is designed to detect the animal's face, isolating this area from the image background. Next, the second model only detects the facial subregions within the facial area. This two-stage approach aims to obtain precise results for larger, more easily identifiable objects, such as faces, and for identifying smaller objects, such as cattle ears, eyes, and muzzles.

To implement the proposed approach, it was essential to prepare the dataset described in Sect. 4.3 using the ground truth labels. At this stage, specific classes in the original dataset were selected, creating two complementary subdatasets for training two independent models. In the first subdataset, named "Annotated Cattle Face Images" (ACFI) subdataset, we kept the entire images but only the annotations related to the animal face. In the second subdataset, designated "Annotated Cattle Facial Subregions Images" (ACFSI) subdataset, faces were ignored and only the annotations of ears, eyes, and muzzle were kept. The original image was cropped to contain only these subregions. Initially, ground truth annotations guided the cropping process, after which a careful manual review was conducted to verify that each crop contained the correct number of features, avoiding any inconsistencies (e.g., no crop showing more than two ears or missing critical features). This thorough filtering process was essential to exclude non-target regions or background elements, focusing solely on the relevant subregions for facial recognition. Additionally, each cropped face was adjusted to maintain a 1:1 aspect ratio, achieved through resizing or padding as necessary. This standardization ensured uniform input dimensions, avoiding image distortion for the model training and evaluation.

Each one of the subdatasets was subdivided into training and test sets based on the original dataset structure to facilitate the performance evaluation. This organization of the original dataset in two new subdatasets is essential to enable specialized training, which addresses the detection of complete faces and the identification of specific features within those faces.

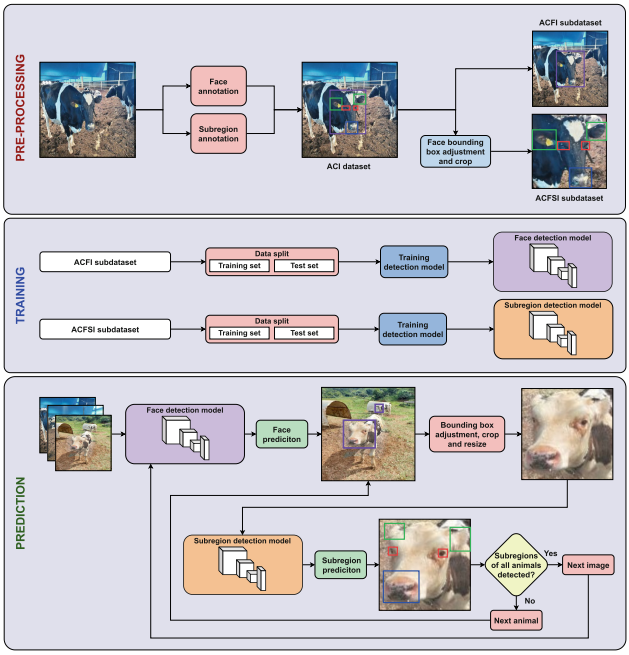


Fig. 3. Proposed detection method overview.

In the next step, model training, two different models were produced. The first one, responsible for face detection, was trained using the ACFI subdataset, enabling it to learn how to automatically identify the presence and location of bovine faces in any cattle images. In parallel, a second model, focused on subregion detection, was trained using the ACFSI subdataset. This second model was trained to detect specific subregions within the face area but with a reduced influence of background noise.

With both models trained, we can apply them to perform the prediction of new images. The prediction system receives an input image and uses the face detection model to locate all faces on the complete image. Given a detected face, its bounding box is adjusted to keep a 1:1 aspect ratio. Then, the face is cropped and resized to the input size of the subregion detection model. This model perform the detection of the subregions on the face image. If the face detection model identifies multiple faces on an image, the system extracts the next face to be processed by the subregion detection model. Otherwise, the face detection model process the following image.

6 Experiments and Results

6.1 Experimental Setup

In this subsection, we present the results of the experiments conducted to evaluate the proposed methods for the problem of detecting the faces of cattle and

their subregions. We conducted four different types of training, where two of them are integral to our proposed method: one dedicated to faces detection in the entire image and another focused on detecting subregions within the cropped face image, i.e. with the background removed, as detailed in the previous section. Additionally, two other approaches were trained for comparison and validation purposes: one to detect both the face and subregions in the entire image, and another to detect only the subregions in the entire image.

The training step were performed in an Intel(R) Core(TM) i7 CPU 870 @ 2.93 GHz processor, 8 GB of single-channel RAM, a Quadro M5000 GPU, and running Ubuntu 22.04 LTS as the operating system. Furthermore, we utilized the implementations provided by Ultralytics for the YOLO models, ensuring that the dependencies required for their usage are consistent with the official releases.

We trained all networks with an input resolution of 640 $\times$ 640 pixels, a batch size of 2, for 40 epochs and with the SGD optimizer in all cases. These settings were determined by the limitations of the available hardware resources. Additionally, we kept the default data augmentation for all networks, except for the horizontal flip. This operation was disabled (set to 0.0) because it inverts the orientation of the face. For the YOLOv8-X we set, in all cases, the initial learning rate at 10^{-3} and linearly decay it to 10^{-4}.

Furthermore, for the all classes, detection problem and for the face detection problem, we set the initial learning rate to 10^{-3} and decay it to 10^{-4} for YOLOv9-E network. For the subregions detection problem with the entire image we set the initial learning rate to 10^{-3} and decay it to 10^{-5} and for the subregions detection problem with cropped image we set the initial learning rate to 10^{-4} and decay it to 10^{-5}.

For YOLOv10-X we set initial learning rate to 10^{-3} and decay it to 10^{-5} in all problems. Furthermore, we enabled the cosine learning rate scheduler, ensuring a smooth and gradual reduction of the learning rate throughout the training process. Additionally, we employed a dropout regularization of 0.5 to help prevent overfitting by randomly deactivating half of the units in each layer. The minimum confidence threshold was set to 0.05, which is less than the default value of 0.25. This adjustment is particularly effective in YOLOv10 since, unlike previous versions such as YOLOv8, it does not use non-maximum suppression (NMS). As a result, lowering the confidence threshold does not negatively affect inference performance, enabling the detection of smaller or more distant objects, contributing, in our case, to the detection of eyes for example. The half-precision floating-point option was enabled to improve computational efficiency.

6.2 Training Results

The results are computed on the test set for each architecture, based on the model of the last epoch of each training. We evaluated the results for precision, recall, mAP50, and mAP50-95. The precision represents the fraction of correctly identified positive instances, called True Positives (TP), out of the total number of predicted positive instances, that is, the sum of TP and False Positives (FP). In contrast, recall represents the proportion of positive instances correctly identified

relative to the total number of actual positive instances, including both TP and False Negatives (FN). The mean Average Precision (mAP) is calculated as follows:

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i, \tag{1}$$

that is, for each class i out of a total of N classes, the area under the Precision-Recall curve is computed. Therefore, mAP50 corresponds to the average of the AP values using an IoU threshold of 0.5, while mAP50-95 is computed using multiple IoU thresholds ranging from 0.5 to 0.95 in increments of 0.05.

We divided the results into four tables, one for each problem. Table 2 refers to the problem of detecting all classes in the entire image, Table 3 presents the results for the problem of detecting only the face in the entire image, Table 4 refers to the problem of detecting subregions in the entire image, and finally, Table 5 shows the values achieved for the problem of detecting subregions in cropped images. The best results in each table are highlighted in bold.

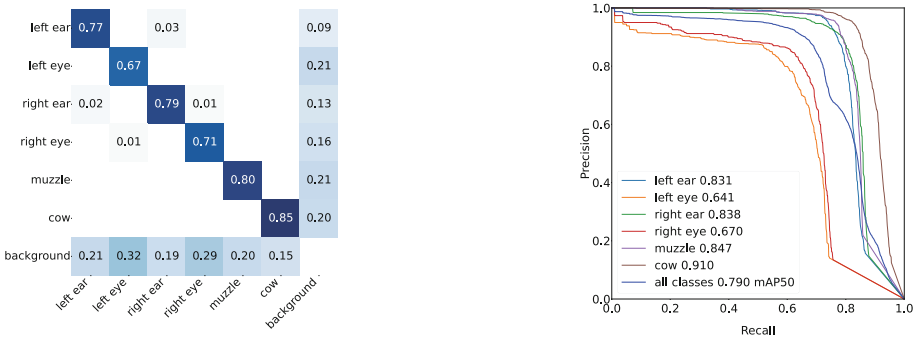
For each of these scenarios, right after the respective table, we present a confusion matrix and a precision-recall (PR) curve corresponding to the model that achieved the best results (except for the face-detection case, which is single-class. In that case, we report only the numerical metrics (Table 3) without confusion matrix or PR curve). These are shown in the following figures: Fig. 4 for the all-classes detection problem, Fig. 5 for subregions detection in entire images, and Fig. 6 for subregions detection in cropped images.

Table 2. Results on the test set for each CNN for the problem of all classes detection in the entire image.

Model	Precision	Recall	mAP50	mAP50-95
YOLOv8-X	0.862	0.689	0.743	0.434
YOLOv9-E	**0.882**	**0.734**	**0.790**	**0.466**
YOLOv10-X	0.864	0.693	0.751	0.429

Table 2 shows that the YOLOv9-E model obtained the best results for the problem of all classes detection in the entire image. In Fig. 4a, we observe that the class cow achieved the best detection accuracy. This result indicates that the face is more easily detected in the entire image compared to the subregions. On the other hand, the classes of eyes present lower accuracy values, evidencing the difficulty of detecting these subregions in the entire image. The behavior of the PR curves in Fig. 4b reinforce these conclusions; the class cow presents high precision values even for high recall values, while the eyes classes curves show a decrease of precision when the recall increases.

In Table 3, we note that YOLOv9-E achieved the best results, except for precision. Compared to Table 2, we observe greater values for all criteria. This fact suggests that face detection is a relatively more straightforward problem.



(a) Confusion matrix. (b) Precision-Recall curve.

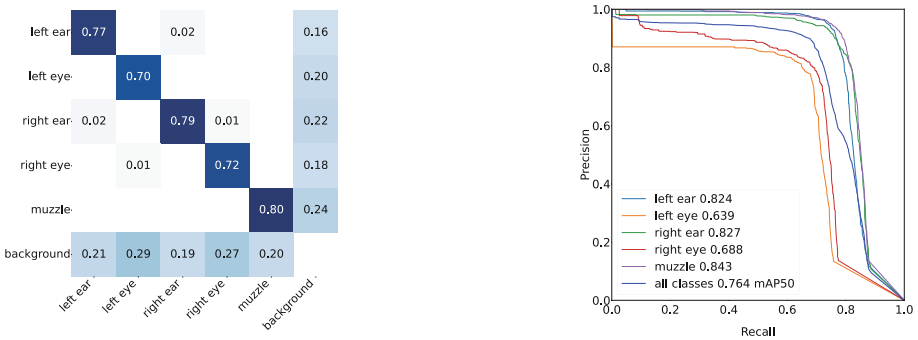
Fig. 4. Detection of all classes in entire images.

Table 3. Results on the test set for each CNN for the problem of face detection.

Model	Precision	Recall	mAP50	mAP50-95
YOLOv8-X	**0.928**	0.795	0.885	0.748
YOLOv9-E	0.919	**0.820**	**0.894**	**0.755**
YOLOv10-X	0.926	0.771	0.869	0.722

Table 4. Results on the test set for each CNN for the problem of subregions detection with entire images.

Model	Precision	Recall	mAP50	mAP50-95
YOLOv8-X	0.855	0.617	0.736	0.372
YOLOv9-E	**0.867**	**0.728**	**0.764**	**0.388**
YOLOv10-X	0.838	0.688	0.732	0.372



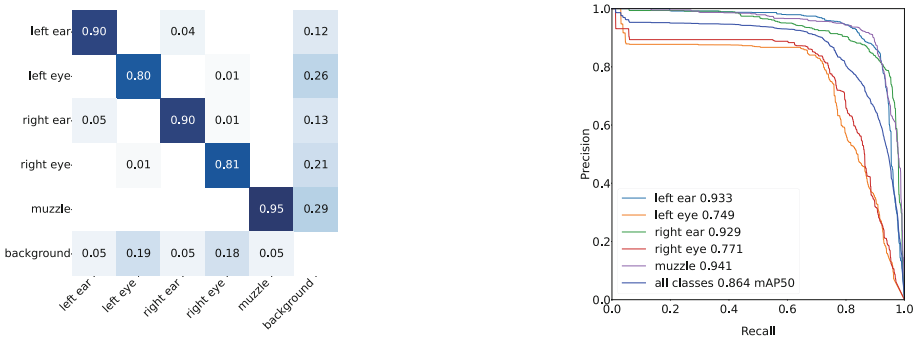
(a) Confusion matrix. (b) Precision-Recall curve.

Fig. 5. Detection of the subregions in entire images.

By the analysis of Table 4, we observe that YOLOv9-E model also presents the best results. We note a decrease in the values compared to Table 2, which highlights the influence of the class cow. The confusion matrix in Fig. 5a also reveals worse results for the eyes classes since these are the smallest subregions. Figure 5b presents the same behavior of Fig. 4b, with the eye classes presenting a sustantial decrease of precision when the recall increases.

Table 5. Results on the test set for each CNN for the problem of subregions detection with cropped images.

Model	Precision	Recall	mAP50	mAP50-95
YOLOv8-X	0.833	0.825	0.842	0.449
YOLOv9-E	**0.847**	**0.826**	**0.864**	**0.472**
YOLOv10-X	0.843	0.816	0.851	0.440



(a) Confusion matrix.

(b) Precision-Recall curve.

Fig. 6. Detection of the subregions in cropped images.

Similar to the other problems, Table 5 shows that YOLOv9-E provided the best results for the problem of detecting subregions in images cropped only with the animal face. We note that the values are better than those in Table 4, which confirms that detecting subregions after cropping the animal face is an advantageous strategy. In Fig. 6a, we note an increase in the accuracy of all classes, although the eyes remain the more difficult classes. The PR curve (Fig. 6b) presents a better behavior, with the recall values presenting a decay only for high precision values.

Thus, through the analysis of the four result tables, it is evident that YOLOv9-E achieved the best mAP50-95 results, which is the most challenging criterion. For comparison, Table 6 presents the results obtained by YOLOv9-E

model across all the addressed problems. We notice that the inclusion of face detection in the problem of all classes detection had a significant impact on the results. The results dropped considerably when this step is removed in the problem of detecting subregions in the entire image. We argue that our two-stage method tends to provide superior results by separating the face detection and the subregions detection.

Table 6. Results on the test set of YOLOv9-E model for all problems described.

Problem	Precision	Recall	mAP50	mAP50-95
All classes	0.882	0.734	0.790	0.466
Face	0.919	0.820	0.894	0.755
Subregions (entire images)	0.867	0.728	0.764	0.388
Subregions (cropped images)	0.847	0.826	0.864	0.472

6.3 Evaluation and Analysis of the Facial and Subregion Detection Method

We developed an application that performs detections according to the method described in Sect. 5. It is applied on the test set. Initially, the algorithm identifies faces using the model trained with the ACFI subdataset. Then, detected faces are first expanded and centered by adding uniform margins so that each region becomes a square (1:1 aspect ratio). Next, these square regions are cropped, resized to 640 $\times$ 640 pixels, and saved. Thus, the face images are delivered to the model trained on the ACFSI subdataset, which detects the subregions. To evaluate the method, we compared the generated annotations with those available in the test set. The comparison is carried out with the Intersection over Union (IoU), where each detected subregion is compared with its corresponding annotated subregion for a given image.

The application operates in two detection modes:

1. **Most Prominent Face:** This mode detects only the largest (most prominent) face in the image along with its subregions. In this mode, 964 faces were detected in 964 images, meaning that one face was successfully detected in every image.
2. **All Faces:** In this mode, the application detects all faces in the image and the subregions within each face. A total of 1,730 faces were detected from 964 images and the dataset annotations include 1,934 faces. Consequently, some faces (typically those less prominent and more challenging for the network to detect) were missed.

We also observed cases in which the same subregion appeared multiple times within a single face. This phenomenon can occur when animals are close to

each other and the cropped face inadvertently includes a subregion belonging to another animal, resulting in duplicate detections of the same subregion. In the **Most Prominent Face** mode, we found 76 duplicate subregion occurrences in 70 images, while in the **All Faces** mode, 116 duplicate subregions were found in 110 images.

We developed two approaches to eliminate these duplicated detections:

1. The more straightforward approach named **Max Confidence** retains only the instance with the highest confidence among the repeated detections.
2. The second approach named **Advanced Filter** verifies for each subregion the intersection with the detected face with a tolerance of 0.8. This allows to identify and eliminate cases where the subregions actually belong to other animals adjacent to the target one. If elimination does not occur through this method, we check whether these detections overlap another subregion with an IoU threshold of 0.8, for every part with multiple detections. This additional step helps to remove further erroneous detections. Finally, if more than one instance remains for the same class, we retain only the detection with the highest confidence.

Table 7 presents the accuracies for each class using the most prominent face and the all faces detection modes. Two result sets are presented: one uses the max confidence strategy to remove duplicates and the other uses the advanced duplication filter. We calculated the Intersection over Union (IoU) between the detected regions and the respective ground truth. We considered a detection as corrected if the IoU was greater than the threshold of 0.5. The proposed pipeline operates with an inference time of approximately 500 ms per stage, amounting to about 1 s per image.

Table 7. Accuracy (%) by subregion and approach.

Subregion	Max Confidence		Advanced Filter	
	Most Prominent	All Faces	Most Prominent	All Faces
Muzzle	94.11	83.60	94.32	83.99
Left Ear	89.50	79.00	89.74	79.22
Right Ear	88.49	83.26	88.49	83.26
Left Eye	73.76	70.86	73.76	70.86
Right Eye	77.39	73.22	77.39	73.22
Overall	85.67	78.93	85.77	79.02

The results show that the Advanced Filter proved advantageous both for the most prominent individual and for the detection of all faces. Additionally, the detection results for the most prominent individual performed better. One explanation is that, in the case of detecting all faces, there are several instances where the network had difficulty in identifying the subregions, as these faces are

relatively smaller and less prominent in the images. Besides, we observe that the "Muzzle" class has higher accuracy, while "Left Eye" and "Right Eye" classes show poorer performances. This may be due to the difficulty in detecting these subregions, as they are less salient compared to the others.

7 Conclusion

This work aimed to apply neural networks on images to detect the faces of cattle as well as their regions of interest, such as eyes, ears, and muzzle, under various conditions of lighting, angle, position, and others. To this end, we proposed a dataset comprising images of cattle faces and their corresponding subregions of interest. We have made this dataset publicly available, which is a contribution of our work. This dataset may be used for future research that requires cattle images with annotations of subregions. Based on this dataset, we trained and evaluated neural networks for object detection.

To avoid the influence of the background in the detection, we developed a two-stage approach. First, we detect the animal faces in the entire image. Then, the face is cropped and the subregions are detected. Through a quantitative analysis of the models, we observed that YOLOv9-E achieved the best performance according to the mAP50-95 evaluation criterion. The network showed satisfactory results for the classes cow, muzzle, and ears (left and right), likely due to their greater salience and larger number of training samples. However, for the eye classes, the overall performance was lower, possibly due to underrepresentation in the dataset and smaller object size compared to other classes.

To validate our strategy, we developed an application that performs facial and subregion detection using the proposed method. The evaluation was conducted on a test set, using IoU. Two detection modes were explored: detecting only the most prominent face or detecting all faces in the image. Although the "All Faces" mode resulted in a larger number of detections, it also introduced more erroneous responses, particularly due to missed detections of smaller or less prominent faces. Furthermore, this mode increased instances of duplicated subregions per face.

To address the face duplication issue, we implemented two filtering approaches: Max Confidence and Advanced Filter. The latter, which incorporates geometric constraints and IoU-based filtering, achieved better overall accuracy by effectively removing erroneously duplicated subregions that were detected. Among the subregions, the "muzzle" class consistently presented the highest detection accuracy, while the "eyes" classes performed less accurately, likely due to their size and lower prominence in the images. The results reinforce the effectiveness of the two-stage detection strategy.

In summary, the results emphasize the strong potential of deep learning-based object detection networks in identifying cattle faces and their key subregions under real-world conditions. As future works we may improve the detection of the eyes by using models for Tiny Object Detection (TOD). Another possibility is to combine different detectors in an ensemble to improve the detection performance.

Acknowledgments. We thank the Minas Gerais State Research Foundation - FAPEMIG (APQ-02230-23) for the financial support in this research.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Food and Agriculture Organization of the United Nations (FAO): Global cattle distribution and production trends (2022). https://www.fao.org/livestock-systems/global-distributions/cattle/en/. Accessed 02 Sept 2024
2. Monteiro, A., Santos, S., Gonçalves, P.: Precision agriculture for crop and livestock farming–brief review. Animals **11**(8), 2345 (2021)
3. Ginger, L., et al.: A six-step process to explore facial expressions performances to detect pain in dairy cows with lipopolysaccharide-induced clinical mastitis. Appl. Animal Behav. Sci. **264**, 105951 (2023)
4. Kumar, P., Luo, S., Shaukat, K.: A comprehensive review of deep learning approaches for animal detection on video data. Int. J. Adv. Comput. Sci. Appl. **14**(11) (2023)
5. Billah, M., Wang, X., Yu, J., Jiang, Y.: Real-time goat face recognition using convolutional neural network. Comput. Electron. Agric. **194**, 106730 (2022)
6. Jaddoa, M.A., Gonzalez, L., Cuthbertson, H., Al-Jumaily, A.: Multiview eye localisation to measure cattle body temperature based on automated thermal image processing and computer vision. Infrared Phys. Technol. **119**, 103932 (2021)
7. Xu, B., et al.: Cattlefacenet: a cattle face identification approach based on retinaface and arcface loss. Comput. Electron. Agric. **193**, 106675 (2022)
8. Bergman, N., Yitzhaky, Y., Halachmi, I.: Biometric identification of dairy cows via real-time facial recognition. Animal **18**(3), 101079 (2024)
9. Zhang, X., Xuan, C., Xue, J., Chen, B., Ma, Y.: LSR-YOLO: a high-precision, lightweight model for sheep face recognition on the mobile end. Animals **13**(11), 1824 (2023)
10. Kawagoe, Y., Kobayashi, I., Zin, T.T.: Facial region analysis for individual identification of cows and feeding time estimation. Agriculture **13**(5), 1016 (2023)
11. Mao, Y., Liu, Y.: Pet dog facial expression recognition based on convolutional neural network and improved whale optimization algorithm. Sci. Rep. **13**(1), 3314 (2023)
12. Tassinari, P., et al.: A computer vision approach based on deep learning for the detection of dairy cows in free stall barn. Comput. Electron. Agric. **182**, 106030 (2021)
13. Smink, M., Liu, H., Döpfer, D., Lee, Y.J.: Computer vision on the edge: individual cattle identification in real-time with readmycow system. In: IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 7056–7065 (2024)
14. Gupta, H., et al.: Computer vision-based approach for automatic detection of dairy cow breed. Electronics **11**(22), 3791 (2022)
15. Zou, Z., Chen, K., Shi, Z., Guo, Y., Ye, J.: Object detection in 20 years: a survey. Proc. IEEE **111**(3), 257–276 (2023)
16. Redmon, J.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)

17. Hussain, M., Khanam, R.: In-depth review of yolov1 to YOLOv10 variants for enhanced photovoltaic defect detection. In: Solar, vol. 4, pp. 351–386. MDPI (2024)
18. Almeida, M.E.G., de Souza Marins, T., da Silva Maciel, L.M., Villela, S.M., Vieira, M.B.: Annotated Cattle Images (ACI) dataset (2025). https://doi.org/10.17632/rzyc5ssvzx.2, https://data.mendeley.com/datasets/rzyc5ssvzx/2

Enhanced Architecture for Non-destructive Leaf Area Estimation Based on a Semantic Segmentation Network

Caio Seixas Duarte, Luiz Maurilio da Silva Maciel, Saulo Moraes Villela, and Marcelo Bernardes Vieira(✉)

Department of Computer Science, Federal University of Juiz de Fora, Juiz de Fora, Brazil
caioseixas.duarte@estudante.ufjf.br,
{luiz.maciel,saulo.moraes,marcelo.bernardes}@ufjf.br

Abstract. The leaf area is a fundamental parameter for evaluating plant growth, development, and physiological status, playing a central role in agricultural and ecophysiological studies. Conventional measurement methods based on physical contact are often destructive, time-consuming, and inefficient, motivating non-destructive approaches based on computer vision and deep learning. However, the high computational cost and architectural limitations of convolutional-based models still represent significant challenges. In this study, architectural improvements are proposed for a deep convolutional neural network derived from an adapted version of DeepLabv3+, aiming to enhance non-destructive leaf area estimation in common bean (Phaseolus vulgaris) using a public dataset. The proposed architecture employs an encoder and two decoders for semantic segmentation and pixel-wise area estimation. The main contributions include the design of a dual-decoder DeepLabv3+-based architecture with a dedicated area estimation branch enhanced by CBAM and C3K2 blocks, along with an analysis through cross-validation and ablation studies to evaluate the impact of architectural design choices. Structural improvements were applied to the area decoder to enhance feature representation. The model was evaluated using the Relative Error Rate (RER) and compared with a reference architecture. Results show a consistent reduction in RER for leaf area estimation. Ablation studies indicate that increasing architectural complexity does not necessarily lead to performance gains. Overall, the proposed improvements increase estimation accuracy, highlighting the potential of efficient deep learning models for precision agriculture.

Keywords: leaf area estimation · deep learning · semantic segmentation

1 Introduction

The leaf is the plant organ responsible for photosynthesis and plays an essential role from both physiological and ecophysiological points of view [1]. Several

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 580–596, 2026.
https://doi.org/10.1007/978-3-032-30491-9_36

parameters associated with this organ are directly linked to plant growth and development, including size, shape, thickness, area, and leaf amount [2]. Among these attributes, leaf area stands out as one of the most relevant indicators for assessing plant growth, productivity, and physiological status [3].

The measurement of leaf area can be performed using contact and non-contact methods. Contact-based methods typically require physical manipulation of plant material, which may damage plant development, demand longer execution times, and are often destructive [4,5]. In contrast, non-contact methods are recognized for their higher speed, while also enabling reductions in resource waste and physical damage during analysis [5,6].

In this context, image-based techniques have been increasingly adopted as non-contact measurement approaches, as they allow the extraction of relevant information without interfering with plant growth and development [5]. Some studies estimate the leaf area using pixel-based methods [7], whereas others exploit images acquired by unmanned aerial vehicles to extract the leaf area index of plant canopies [8].

With the advancement of computer vision, a wide range of methods have been employed for image analysis and processing, enabling greater accuracy in tasks such as semantic segmentation [9]. In this context, Convolutional Neural Networks have stood out in performing these tasks due to their superior learning capability and performance, which is attributed to their hierarchical architecture [10]. However, there are still several relevant challenges, particularly those related to feature refinement and feature representation [11,12]. Therefore, attention mechanisms such as CBAM and convolutional structures such as C3K2 have demonstrated the ability to improve the model's ability to discriminate important features and information and improve the capture of fine details, respectively [12,13].

In this scenario, the present work proposes architectural enhancements to a deep convolutional neural network derived from a previously adapted version of DeepLabv3+ [14], aiming to improve the non-destructive estimation of the leaf area of common beans using a public dataset [15]. To this end, our main contributions comprise new structural modifications introduced into the DeepLabv3+ based architecture, including the incorporation of novel blocks such as CBAM and C3K2 to analyze their impact on model performance, and a five-fold cross-validation for a more comprehensive analysis of the results, along with an ablation study varying parameters and the internal structure of the incorporated blocks. Furthermore, we compared the performance of the proposed model with the results obtained from the reference architecture and performed an analysis of predictions by image.

2 Background

In recent years, several studies have been developed that focus on estimating leaf area through computational models [16]. This interest arises from the relevance of this metric for monitoring plant physiological parameters, such as photosynthetic

capacity, respiration, and evapotranspiration [17], as well as its importance for tracking plant growth [18] and for prediction of agricultural yield [19].

The methods used for measurement of the leaf area can be classified as direct (contact-based) and indirect (non-contact) approaches [20]. Direct methods rely on physical measurements made in the field, requiring substantial time and labor, in addition to potentially causing damage to plant development. In contrast, indirect methods are known to offer greater speed, accuracy, and flexibility, while also reducing damage to analyzed resources [4–6].

Among the tools used to implement indirect methods, Computer Vision techniques, particularly those based on semantic segmentation, have demonstrated promising results in this context [21]. A relevant example is the work proposed by Silva et al. [14], which presented an adaptation of the DeepLabv3+ network to estimate the are of bean leaves using images containing a leaf and a fiducial marker, achieving satisfactory results.

Despite its strong performance in semantic segmentation tasks, the DeepLabv3+ architecture presents important limitations, such as the high computational cost resulting from its complex feature extractor and the large number of parameters [22,23]. Furthermore, a progressive reduction of spatial resolution in the encoder can lead to a loss of information and fine details in the decoder [22].

In light of these challenges, the search for architectural adaptations that incorporate more efficient mechanisms with lower computational costs has become increasingly common [22,24]. In this context, modules such as the Convolutional Block Attention Module (CBAM) and efficient convolutional blocks, such as C3K2, have been employed to enhance performance in various Computer Vision tasks, including semantic segmentation, by promoting the prioritization of visually relevant regions and the efficient refinement of extracted features [13,22].

The C3K2 block was introduced in YOLOv11, an architecture designed for object detection and instance segmentation, with the objective of improving feature extraction while maintaining low structural complexity. Thus, it seeks to balance the representational capacity of the model without incurring a high computational cost [25]. Additionally, this block contributes to more effective feature refinement, resulting in more discriminative visual representations [13].

In terms of its internal structure, the C3K2 module extends the conventional C3 structure by incorporating a more flexible and efficient feature processing strategy. Instead of employing large convolutional kernels, it leverages parallel convolutional paths that approximate a wider receptive field while keeping the computational cost low. This module begins with a convolution that expands the input feature channels, followed by a split operation that separates the feature map into two distinct paths. One branch bypasses most of the transformations, preserving original information, while the other is processed through a sequence of residual bottleneck units, which may incorporate the C3K structure based on the iterative configuration. Within this processing path, parallel convolutional operations are employed to approximate a larger receptive field, and their out-

puts are subsequently merged to enrich the feature representation. After these transformations, the different branches are concatenated and passed through a final convolutional layer to fuse the aggregated information and produce the output feature map. This design, combining some operations with residual processing, enables efficient feature extraction and representation while maintaining a reduced computational overhead [26].

In turn, CBAM is a lightweight attention mechanism proposed to enhance the capability of convolutional neural networks with reduced computational overhead [27]. This module sequentially applies channel and spatial attention mechanisms, allowing the network to emphasize relevant features while suppressing less informative ones [22]. By operating directly on intermediate feature maps, CBAM performs adaptive feature refinement and can be easily integrated into different convolutional architectures without requiring significant structural modifications [22,27].

Based on these considerations, this work proposes the adaptation of a DeepLabv3+ based deep neural network through modifications to the decoder responsible for estimating leaf area and the fiducial marker. These modifications aim to reduce information loss and preserve fine details, thereby increasing the efficiency of the proposed network compared to the reference model.

3 Proposed Method

For this work, a neural network based on DeepLabv3+ was employed, implemented in TensorFlow 1.15.0. The architecture was modified from the reference model proposed by Silva et al. [14] to improve the prediction of the pixels areas corresponding to leaves. Figure 1 presents an overview of the proposed architecture, highlighting the new added blocks.

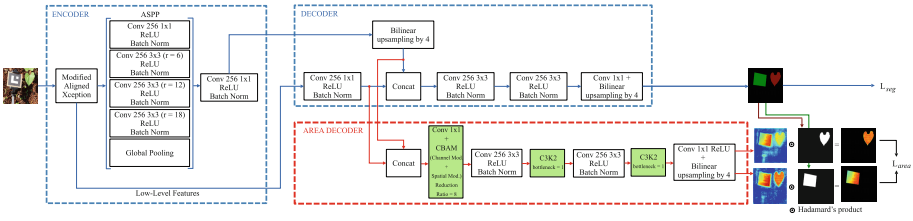


Fig. 1. Proposed model architecture. The blocks highlighted in green indicate the Convolution 1×1, CBAM and C3K2 introduced to the Area Decoder in the architecture proposed by Silva et al. [14]. (Color figure online)

3.1 Adapted Deeplabv3+

For leaf area estimation, we used a network based on the reference model. This network simultaneously performs semantic segmentation and estimation of the

per-pixel area of objects present in the input image, as illustrated in Fig. 2. The architecture is composed of an encoder and two distinct decoders, with the first responsible for semantic segmentation and the second responsible for per-pixel area regression.

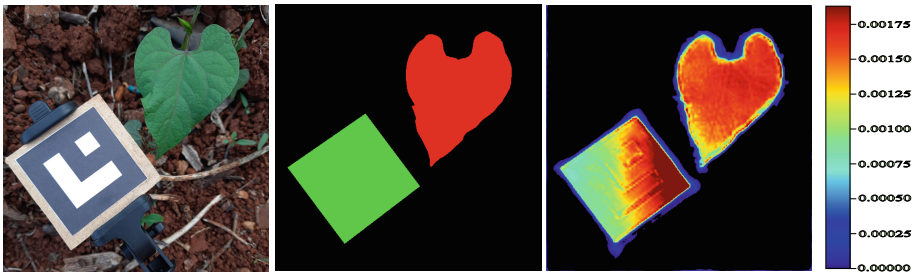


Fig. 2. From the left to right, the input sample used, the segmentation output and the area output.

The encoder consists of a convolutional backbone network, followed by an Atrous Spatial Pyramid Pooling (ASPP) module, formed by parallel separable convolutions and finalized by a 1×1 convolution with 256 channels. As in the reference work [14], we adopted the Aligned Xception as the backbone. The ASPP module is used to capture contextual information at multiple scales, while the 1×1 convolution acts to fuse information across channels, contributing to the preservation of the spatial resolution of the feature maps.

From the encoder, the feature maps are directed to the original DeepLabv3+ decoder, which is responsible for semantic segmentation. At this stage, low-level feature maps extracted directly from the backbone and processed by a 1×1 convolution are concatenated with the maps originating from the ASPP, which were previously resized through bilinear upsampling by a factor of 4. Subsequently, two separable 3×3 convolutions are applied to refine the features, and a 1×1 convolution, followed by another bilinear upsampling with the same factor, finalizes the process of generating the segmentation masks and logits.

In parallel with the segmentation decoder, the architecture incorporates a decoder dedicated to area estimation. Similarly to the original DeepLabv3+ decoder, this structure performs the concatenation of low-level features with the maps produced by the ASPP, followed by the application of two 3×3 separable convolutions and a 1×1 convolution, accompanied by bilinear upsampling. However, this decoder differs by employing the ReLU activation function, ensuring the generation of positive area values, as well as by producing maps that represent the per-pixel estimated area of the objects [14]. Finally, a Hadamard product is performed between the segmentation masks provided by the segmentation decoder and the per-pixel area maps, enabling the separate estimation of the areas of the leaf and the fiducial marker, culminating in the generation of area logits.

3.2 Modified Decoder for Leaf Area Estimation

Based on the reference architecture, with the objective of improving the estimation of leaf area on a per-pixel basis, we implemented relevant modifications in the area decoder architecture. Starting from the premise that the depth of the network can be explored, blocks such as C3K2 and CBAM were added. These modifications are highlighted in green in Fig. 1.

We added a 1×1 convolution to the area decoder after the concatenation stage, with the purpose of performing channel fusion between low-level feature maps and those originating from the ASPP module. Following this, we added a CBAM block that is responsible for applying channel and spatial attention. This strategy aims to enable the network to emphasize the features and regions of the image that are relevant for the area estimation. Additionally, a C3K2 block was inserted after each 3×3 separable convolution, which improves the refinement of the extracted features.

The added blocks have internally adjustable parameters. The CBAM's channel attention module has a parameter called the reduction rate, which regulates the number of input tensor channels. The C3K2 block, in turn, allows for variation in the number of internal modules, which are named bottlenecks. We performed an ablation study of these parameters, described in Sect. 4.2.

The loss function used in the proposed model was maintained as defined by Silva et al. [14], which is composed of the sum of the semantic segmentation loss and the area estimation loss.

Furthermore, an ablation study of the proposed architecture was conducted, in which different configurations of the added blocks were evaluated. This study included variations of the reduction ratio in the CBAM, the number of bottlenecks in the C3K2, as well as the position of internal modules within the CBAM. These experiments were performed using fold 1 of the first cross-validation of the dataset. Additionally, complete experiments were conducted considering all five folds of the same cross-validation. In all cases, the results were compared with those obtained using the network proposed by Silva et al. [14].

4 Experiments and Results

To evaluate our method, we performed a comparative analysis with the reference network. We preserved the main training hyperparameters, including *atrous* rates, weight decay, learning policy, learning power, optimizer, and momentum to ensure a fair comparison between the approaches, as detailed on Table 1. Likewise, the weight initialization and pre-training strategies were also maintained. As performed on the network of baseline, the encoder and decoder networks were initialized with weights pre-trained on the PASCAL VOC 2012 dataset [28], with the exception of the logits layer. Separable convolution layers were initialized from a truncated normal distribution (mean = 0.0, standard deviation = 0.33). For the remaining convolutional layers, Xavier initialization [29] was adopted, drawing weights from a uniform distribution. Only the hyperparameters learning rate and batch size differed from those defined in the reference work. The batch

size was set to 8. We conducted the experiments using one NVIDIA A30 GPU with 24 GB of VRAM.

Table 1. Main hyperparameters used on the present work.

Parameter	Values
Atrous rates	[6, 12, 18]
Weight decay	0.00004
Learning policy	Polynomial
Learning power	0.9
Optimizer	Momentum
Momentum (m)	0.9

For model training, a maximum number of 150,000 steps was defined. Performance evaluation was carried out using 51 checkpoints saved at regular intervals of 100 steps, ranging from step 145,000 to step 150,000. After training the models, the checkpoints saved during this stage were evaluated in the training set, and based on these results, the best step was selected. Subsequently, during the testing phase, this selected model was evaluated in the testing set. As a performance metric, the Relative Error Rate (RER) was adopted, as defined in Eq. 1, which corresponds to the percentage ratio between the absolute difference between the estimated area (a_{est}) and the real area (a_{real}) relative to the real area [14]:

$$\text{RER}(\%) = \left| \frac{a_{\text{est}} - a_{\text{real}}}{a_{\text{real}}} \right| \cdot 100 \tag{1}$$

The selection of the best model considers the sum of the mean and standard deviation of the RER for both the leaf and the fiducial marker. Thus, the selected model was the one that presented the lowest value of $\mu_l + \sigma_l + \mu_m + \sigma_m$.

4.1 Dataset

The dataset used in this study consists of images of common bean leaves in the vegetative stage, captured together with a custom 5 × 5 cm fiducial marker from the ArUco library [30]. The dataset contains 6,981 images of 512 × 512 pixels from 612 distinct leaves [15]. The dataset is organized into a 5-5-fold distribution. Each leaf appears exclusively in the training or test set, preventing any overlap within the folds. We used the first cross-validation for our experiments. For some ablation tests, we employed only the first split. For our final configuration, all five splits were evaluated, enabling a complete analysis of the dataset. Since Silva et al. [14] used only a partial version of the dataset, our "baseline" model is an evaluation of the reference model on the complete dataset.

Table 2. Ablation study varying the CBAM reduction ratio parameter. The values of Avg RER and Std RER are shown in percentage.

Description	Train				Test			
	Leaf		Marker		Leaf		Marker	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Baseline	1.582	0.866	1.480	0.809	6.310	5.267	1.708	1.290
Proposed (Reduction Ratio = 2)	**1.016**	0.869	0.655	0.362	6.119	4.961	1.318	1.046
Proposed (Reduction Ratio = 8)	1.084	0.766	**0.321**	**0.226**	**5.989**	**4.937**	**1.147**	**0.949**
Proposed (Reduction Ratio = 16)	1.108	**0.709**	0.508	0.307	6.543	5.411	1.246	1.009

4.2 Ablation Study

To evaluate the individual contribution of the mechanisms incorporated into the proposed architecture, an ablation study was conducted based on variations in parameters and the internal arrangement of the added structures, while keeping the positioning of these structures within the architecture unchanged. In this way, different variations of the model were analyzed to investigate the influence of the CBAM and C3K2 modules on the performance of the leaf area estimation task. For the ablation experiments, we utilized only the first fold of the first cross-validation of the dataset.

Initially, we evaluated variations of the reduction ratio parameter of the CBAM channel attention module to analyze the impact of this module on network performance. For this purpose, we considered the values 2, 8, and 16, keeping the sequential arrangement of the channel and spatial attention modules in the CBAM, as well as the number of bottlenecks of the C3K2 set to 1. Table 2 presents the corresponding results.

Subsequently, we analyzed the effects of varying the number of bottlenecks in the C3K2 module, to evaluate the influence of this parameter on the refinement of features in the estimation of leaf and marker areas. For this experiment, we adopted the number of bottlenecks 1, 2, and 6, keeping the reduction ratio of the CBAM fixed at 8 and maintaining the same structural arrangement of this attention module adopted in the previous experiment. Table 3 shows the results.

Finally, we performed experiments by varying the internal arrangement of the CBAM modules in a manner analogous to the procedure described in the original work [12]. We evaluated sequential configurations of the channel and spatial attention modules in both orders, as well as a parallel configuration. In these experiments, we kept 1 bottleneck for C3K2 and set the reduction ratio in the attention module equal to 8. Table 4 displays the results achieved.

4.3 Results with Five-Fold Cross-Validation

To validate the results achieved in the ablation study and to verify the absence of bias associated with the use of a single cross-validation fold, we conducted

Table 3. Ablation study varying the number of bottlenecks of C3K2. The values of Avg RER and Std RER are shown in percentage.

Description	Train				Test			
	Leaf		Marker		Leaf		Marker	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Baseline	1.582	0.866	1.480	0.809	6.310	5.267	1.708	1.290
Proposed (Bottleneck = 1)	**1.084**	**0.766**	**0.321**	**0.226**	**5.989**	**4.937**	1.147	0.949
Proposed (Bottleneck = 2)	1.104	0.814	0.470	0.292	6.135	4.959	1.241	1.019
Proposed (Bottleneck = 6)	1.254	0.782	0.626	0.411	6.867	5.731	**1.144**	**0.897**

Table 4. Ablation study varying the arrangement of internal structures of the CBAM. The values of Avg RER and Std RER are shown in percentage.

Description	Train				Test			
	Leaf		Marker		Leaf		Marker	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Baseline	1.582	0.866	1.480	0.809	6.310	5.267	1.708	1.290
Channel → Spatial	**1.084**	0.766	**0.321**	**0.226**	**5.989**	**4.937**	1.147	0.949
Spatial → Channel	1.255	0.825	0.496	0.314	6.443	5.412	**1.136**	**0.941**
Channel || Spatial	1.261	**0.760**	0.905	0.574	6.194	5.077	1.403	1.031

experiments using both the baseline network and the model proposed in this work, considering all five folds of the first cross-validation of the dataset. For this evaluation, we computed the RER individually for each test image across all folds, accumulated the obtained values and subsequently divided by the total number of images in the dataset. Thus, the final RER values for the leaf and the marker represent the average error considering all 6,981 images in the dataset, ensuring a global and unbiased evaluation of the model. Table 5 shows the results.

4.4 Prediction by Image

In this section, we present the results of image-level predictions generated by the adapted network proposed in this study, considering the five folds of the first cross-validation procedure.

Figure 3 illustrates the relationship between the predicted leaf areas and the reference areas through scatter plots generated for each fold. In these plots, the estimated values are compared with the ground truth, and a fitted linear regression line is also presented, along with the corresponding coefficient of determination (R^2). Across all folds, the predictions follow a consistent linear trend, with R^2 values of 0.858, 0.811, 0.820, 0.803, and 0.747, indicating strong agreement between the estimated and actual areas in most cases, particularly on fold 1, which achieved the best performance. However, the presence of vertical group-

Table 5. RER per image of dataset for leaf and marker. The values of Avg RER and Std RER are shown in percentage.

Description	Leaf		Marker	
	Avg	Std	Avg	Std
Baseline	6.787	5.547	1.547	1.215
Proposed model	**6.578**	**5.516**	**1.155**	**0.941**

ings of points can be observed, suggesting variability in the predictions for the same leaf under different conditions.

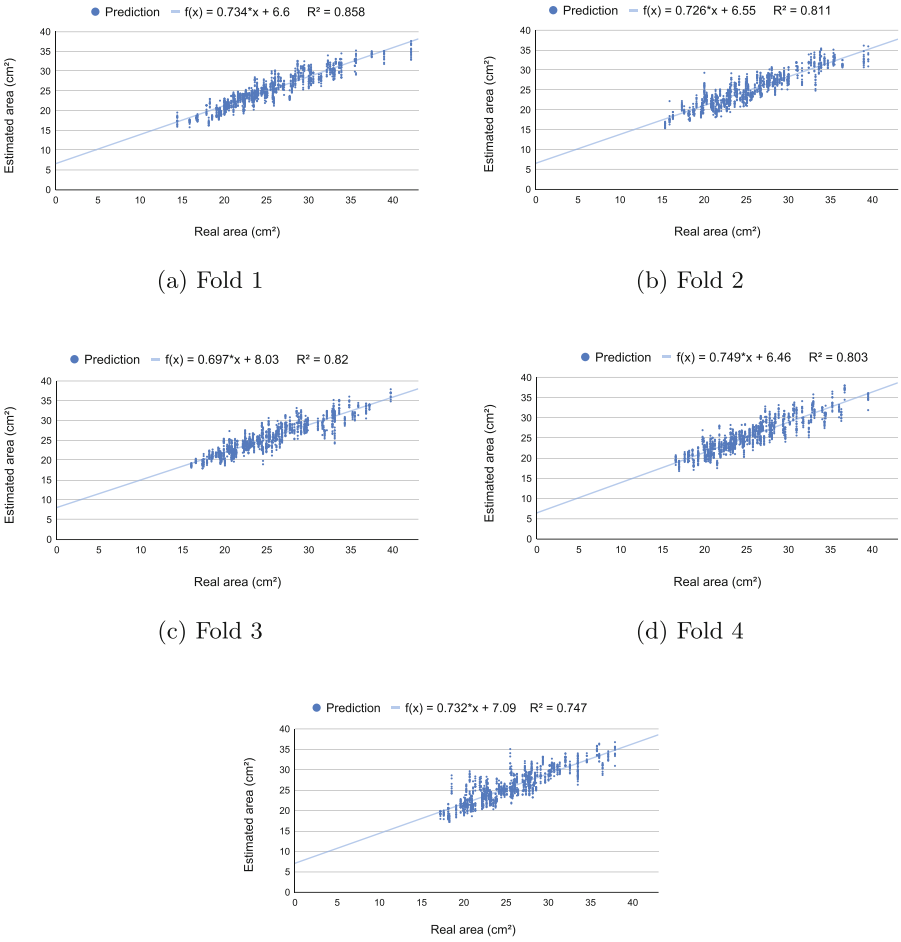
The behavior of the prediction errors is illustrated in Fig. 4, which present the error distributions for each fold. The errors, calculated as the difference between the predicted and reference values, reveal a bias toward underestimation in most folds, with the exception of folds 2 and 4, which present a greater balance between positive and negative prediction errors. In fold 1, the error values for leaf range from -5.50 to $9.50\,\mathrm{cm}^2$, with a higher concentration in the interval between -1.04 and $-0.64\,\mathrm{cm}^2$, corresponding to 139 out of 1387 images. In the remaining folds, an increase in the error range is observed, with intervals of -9.50 to $9.00\,\mathrm{cm}^2$, -7.00 to $9.00\,\mathrm{cm}^2$, -7.50 to $8.00\,\mathrm{cm}^2$, and -10.50 to $8.00\,\mathrm{cm}^2$, respectively, indicating greater dispersion in the achieved errors.

4.5 Qualitative Results

Figure 5 presents qualitative results obtained by the proposed model for representative leaf samples. Each example includes the input image, the segmentation output, the estimated area output, and the corresponding ground truth. These visual results allow a qualitative evaluation of the behavior of the proposed architecture under different leaf and marker perspectives, geometries, and positions.

The qualitative results indicate that the proposed architecture was able to accurately segment both the reference marker and the leaf region under different acquisition conditions, while also producing area estimations visually consistent with the ground truth maps. Among the evaluated samples, Fig. 5(a) achieved the best overall correspondence, with predicted area maps presenting homogeneous distributions and well-defined object boundaries for both the marker and the leaf. Figure 5(b) also presented strong agreement with the ground truth, although with slightly smoother transitions near the object contours. These results suggest that the model preserves geometric consistency even under moderate variations in object orientation and illumination.

In contrast, Fig. 5(c) exhibited larger discrepancies between the estimated area maps and the ground truth, particularly in regions affected by perspective distortion and non-uniform illumination. Although the segmentation masks remained coherent, the estimated area distribution became less homogeneous, especially in the marker regions. Nevertheless, the proposed method preserved the overall object shape and spatial correspondence, indicating robustness and consistency in leaf area estimation.



(a) Fold 1

(b) Fold 2

(c) Fold 3

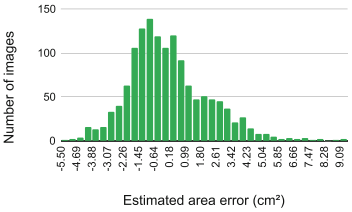
(d) Fold 4

(e) Fold 5

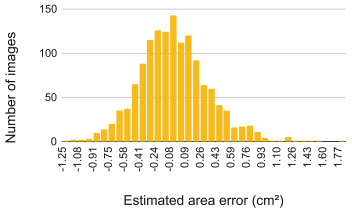
Fig. 3. Scatter plots of the estimated leaf areas for all folds in Cross-Validation 1.

4.6 Discussion

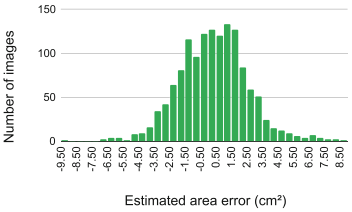
In this subsection, we discuss the contribution and impact of the mechanisms incorporated into the proposed architecture, as well as the influence of their parameters on the task of estimating the leaf area, compared to the baseline network. Based on the results obtained, it is observed that the modifications introduced in the model, in most of the evaluated configurations, consistently contributed to increasing the network accuracy. In particular, considering the results obtained on the leaf test set, the best configuration of the proposed model was composed of the CBAM with a reduction ratio equal to 8 and with a sequential internal arrangement with a channel attention module and a spatial



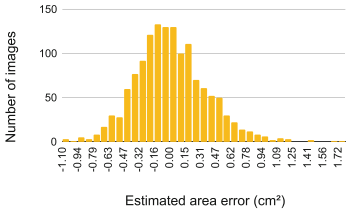
(a) Leaf prediction errors for fold 1.



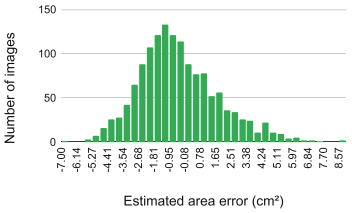
(b) Marker prediction errors for fold 1.



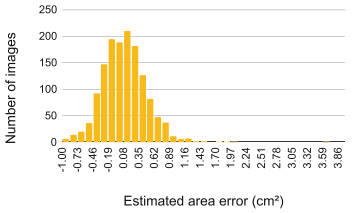
(c) Leaf prediction errors for fold 2.



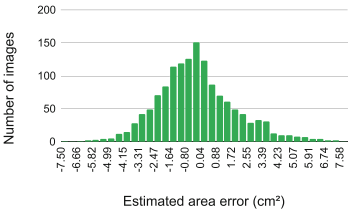
(d) Marker prediction errors for fold 2.



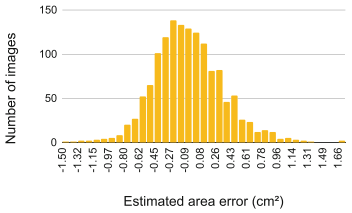
(e) Leaf prediction errors for fold 3.



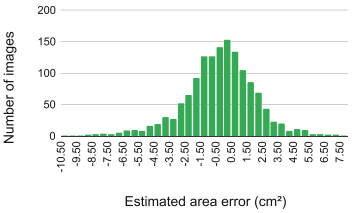
(f) Marker prediction errors for fold 3.



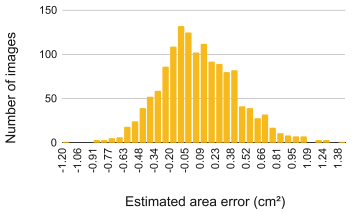
(g) Leaf prediction errors for fold 4.



(h) Marker prediction errors for fold 4.

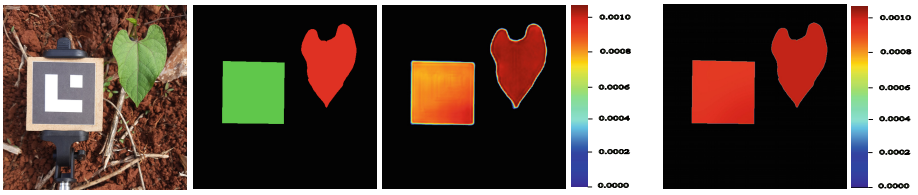


(i) Leaf prediction errors for fold 5.

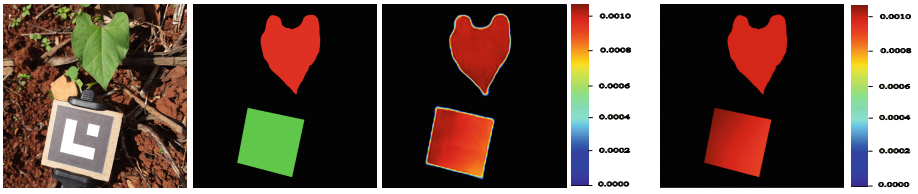


(j) Marker prediction errors for fold 5.

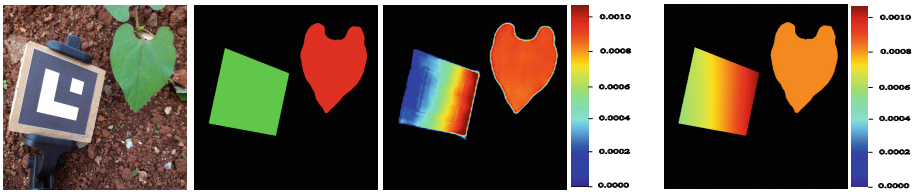
Fig. 4. Histogram of leaf and marker prediction errors achieved by the best model in the test set across all folds.



(a) First example.



(b) Second example.



(c) Third example.

Fig. 5. Qualitative results for some leaf samples. From left to right: input sample, segmentation output, area output and ground truth area image.

module, and with a bottleneck quantity equal to 1 in the C3K2. This architecture presented significant reductions compared to the baseline, both in mean and standard deviation of the RER, in the training and test sets, for the estimation of leaf area, which constitutes the main objective of this work, as well as for fiducial marker area estimation, although, in other configurations implemented in the ablation study, the network presented slightly superior performance in the training for the leaf and in the marker test.

It should be noted that the marker exhibited a more expressive gain compared to the leaf when comparing the results of the proposed model with those obtained using the baseline. This behavior can be attributed to the more uniform and well-defined geometry of the marker (plane), which tends to facilitate the network's learning process. However, since the results were consistent for both objects, it is possible to infer that the proposed model is capable of learning diverse representations, regardless of the geometric complexity of the region of interest.

The analysis of the results presented in Table 3 shows that the C3K2 module achieves better performance when used in a more compact manner, that is, with fewer bottlenecks. This behavior suggests that increasing the depth of the

module does not necessarily lead to improvements in feature refinement for the area estimation task. In contrast, deeper configurations tend to introduce greater complexity into the model, which can hinder the learning process and result in marginal gains or even performance degradation. Therefore, the results indicate that C3K2 operates more effectively when used as a lightweight refinement mechanism, highlighting the importance of balancing representational capacity and computational cost.

In a complementary manner, the results obtained from the CBAM ablation study indicate that the inclusion of this module tends to contribute positively to the performance of the proposed model, especially when smaller values are adopted for the reduction ratio parameter. As shown in Table 2, configurations with reduced values of this parameter exhibited superior performance, possibly preserving a greater amount of information within the channel attention module. This behavior may be related to a more adequate representation of features relevant to the leaf area estimation task, which depends on details associated with texture and edges.

Additionally, the results show that variations in the arrangement of the internal CBAM modules, different from the sequential configuration composed of the channel attention module followed by the spatial attention module, did not result in significant performance improvements. This finding is in agreement with the arrangement originally proposed by Woo et al. [12]. Thus, the results reinforce that the appropriate choice of parameters and the internal organization of the CBAM modules exert a direct influence on the gains provided by the attention mechanisms in the proposed architecture.

The results presented in Table 5 indicate that the proposed model shows consistent performance in estimating the area of both leaf and the fiducial marker. The analysis of the image-level RER across the five folds suggests good generalization capability, without dependence on a specific portion of the data. In addition, the proposed architecture outperformed the reference network presented by Silva [14], highlighting the effectiveness of the architectural modifications adopted.

The image-level prediction analysis also reinforces the consistency of the proposed model across the evaluated folds. The scatter plots demonstrated a tendency of linear relationship between the predicted and reference leaf areas, with satisfactory R^2 values in all folds, indicating that the model presents high correlation between the estimated and real values. In addition, although the error histograms revealed a slight tendency toward underestimation in most folds, the error distributions remained concentrated around small values, suggesting that the proposed architecture maintains consistent predictions even under different image conditions.

It is possible to observe that the obtained results demonstrate the practical relevance of incorporating lightweight attention and refinement mechanisms into architectures dedicated to agricultural computer vision tasks. By improving estimation accuracy with a moderate increase in computational cost, the proposed approach may contribute to the development of more reliable systems for leaf

area estimation. This is particularly important in real-world scenarios, in which a robust area estimation may assist in crop monitoring, plant growth analysis, and precision agriculture applications. Furthermore, the results suggest that the proposed architectural modifications favor the performance and generalization capability of the model.

5 Conclusion

In this work, we introduced CBAM and C3K2 blocks into a semantic segmentation based network for leaf area estimation. The experiments conducted indicate that the integration of these blocks enhances the performance of semantic segmentation networks for leaf area estimation. The results demonstrate that the modified network outperformed the reference model based on DeepLabv3+, achieving higher accuracy in estimating the area of both leaf and fiducial marker. However, it was observed that the gain was more expressive for the marker, which can be attributed to its more uniform and well-defined geometry. These results highlight the potential of deep neural networks for agricultural applications, particularly in the non-destructive analysis of plant morphological features. The incorporation of C3K2 and CBAM modules significantly contributed to both improved performance, an aspect often cited as a challenge in the adoption of deep models in practical scenarios. Despite these promising results, future studies could explore different hyperparameter configurations to further optimize model performance, as well as strategies to reduce the difference in gain between the analyzed objects. Additionally, as a way to address a limitation of this work, it may be interesting to use the proposed model on more than one dataset, aiming to evaluate the generalization potential of the proposed network.

Acknowledgments. The authors would like to thank the Minas Gerais State Research Foundation - FAPEMIG, Brazil. This study was financed in part by the Federal University of Juiz de Fora - UFJF. This work has been supported by UFJF's High-Speed Integrated Research Network (RePesq). https://www.repesq.ufjf.br/.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. White, A.C., Rogers, A., Rees, M., Osborne, C.P.: How can we make plants grow faster? A source-sink perspective on growth rate. J. Exp. Bot. **67**(1), 31–45 (2016). https://doi.org/10.1093/jxb/erv447
2. Huang, W., Ratkowsky, D.A., Hui, C., Wang, P., Su, J., Shi, P.: Leaf fresh weight versus dry weight: which is better for describing the scaling relationship between leaf biomass and leaf area for broad-leaved plants? Forests **10**(3), 256 (2019). https://doi.org/10.3390/f10030256
3. Fang, H., Baret, F., Plummer, S., Schaepman-Strub, G.: An overview of global leaf area index (LAI): methods, products, validation, and applications. Rev. Geophys. **57**(3), 739–799 (2019). https://doi.org/10.1029/2018RG000608

4. Kim, D.C., Anderson, N.O.: Comparative analysis of laboratory freezing methods to establish cold tolerance of detached rhizomes and intact crowns in garden chrysanthemums (dendranthema× grandiflora tzvelv.). Scientia horticulturae **109**(4), 345–352 (2006). https://doi.org/10.1016/j.scienta.2006.05.014
5. Mohammadi, V., Minaei, S., Mahdavian, A.R., Khoshtaghaza, M.H., Gouton, P.: Estimation of leaf area in bell pepper plant using image processing techniques and artificial neural networks. In: 2021 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), pp. 173–178. IEEE (2021). https://doi.org/10.1109/ICSIPA52582.2021.9576778
6. Xing, M., et al.: Physiological alterations and nondestructive test methods of crop seed vigor: a comprehensive review. Agriculture **13**(3), 527 (2023). https://doi.org/10.3390/agriculture13030527
7. Doğan, A., et al.: Grapevine leaf area measurements by using pixel values. Comptes rendus de l'Académie bulgare des sciences: sciences mathématiques et naturelles **71** (2018). https://doi.org/10.7546/crabs.2018.06.07
8. Roth, L., Aasen, H., Walter, A., Liebisch, F.: Extracting leaf area index using viewing geometry effects—a new perspective on high-resolution unmanned aerial system photography. ISPRS J. Photogramm. Remote. Sens. **141**, 161–175 (2018). https://doi.org/10.1016/j.isprsjprs.2018.04.012
9. Lei, L., Yang, Q., Yang, L., Shen, T., Wang, R., Fu, C.: Deep learning implementation of image segmentation in agricultural applications: a comprehensive review. Artif. Intell. Rev. **57**(6), 149 (2024). https://doi.org/10.1007/s10462-024-10775-6
10. Kamilaris, A., Prenafeta-Boldú, F.X.: Deep learning in agriculture: a survey. Comput. Electron. Agric. **147**, 70–90 (2018). https://doi.org/10.1016/j.compag.2018.02.016
11. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 833–851. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_49
12. Woo, S., Park, J., Lee, J.-Y., Kweon, I.S.: CBAM: convolutional block attention module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 3–19. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_1
13. He, L.h., Zhou, Y.z., Liu, L., Cao, W., Ma, J.h.: Research on object detection and recognition in remote sensing images based on YOLOv11. Sci. Rep. **15**(1), 14032 (2025). https://doi.org/10.1038/s41598-025-96314-x
14. Silva, K.G.F., Silva Maciel, L.M., Villela, S.M., Pedrini, H., Morais, L.E., Vieira, M.B.: Non-destructive leaf area estimation based on a semantic segmentation deep neural network. Comput. Electron. Agric. **237**, 110551 (2025). https://doi.org/10.1016/j.compag.2025.110551
15. Silva, K.G.F., et al.: Bean leaf image dataset annotated with leaf dimensions, segmentation masks, and camera calibration. Data Brief **59**, 111328 (2025). https://doi.org/10.1016/j.dib.2025.111328
16. Mthembu, N., Lottering, R., Kotze, H.: Forest, crop and grassland leaf area index estimation using remote sensing: a review of current research methods, sensors, estimation models and accomplishments. Appl. Sci. **13**(6), 4005 (2023). https://doi.org/10.3390/app13064005
17. Du, L., et al.: Estimating leaf area index of maize using UAV-based digital imagery and machine learning methods. Sci. Rep. **12**(1), 15937 (2022). https://doi.org/10.1038/s41598-022-20299-0

18. Dong, Y., et al.: A method for retrieving coarse-resolution leaf area index for mixed biomes using a mixed-pixel correction factor. IEEE Trans. Geosci. Remote Sens. **61**, 1–17 (2023). https://doi.org/10.1109/TGRS.2023.3235949
19. Gitelson, A.A., Viña, A., Arkebauer, T.J., Rundquist, D.C., Keydan, G., Leavitt, B.: Remote estimation of leaf area index and green leaf biomass in maize canopies. Geophys. Res. Lett. **30**(5) (2003). https://doi.org/10.1029/2002GL016450
20. Zheng, G., Moskal, L.M.: Retrieving leaf area index (LAI) using remote sensing: theories, methods and sensors. Sensors **9**(4), 2719–2745 (2009). https://doi.org/10.3390/s90402719
21. Luo, Z., Yang, W., Yuan, Y., Gou, R., Li, X.: Semantic segmentation of agricultural images: a survey. Inf. Process. Agric. **11**(2), 172–186 (2024). https://doi.org/10.1016/j.inpa.2023.02.001
22. Fu, H., et al.: DSC-deeplabv3+: a lightweight semantic segmentation model for weed identification in maize fields. Front. Plant Sci. **16**, 1647736 (2025). https://doi.org/10.3389/fpls.2025.1647736
23. Liu, Y., Bai, X., Wang, J., Li, G., Li, J., Lv, Z.: Image semantic segmentation approach based on deeplabv3 plus network with an attention mechanism. Eng. Appl. Artif. Intell. **127**, 107260 (2024). https://doi.org/10.1016/j.engappai.2023.107260
24. Chen, H., Qin, Y., Liu, X., Wang, H., Zhao, J.: An improved deeplabv3+ lightweight network for remote-sensing image semantic segmentation. Complex Intell. Syst. **10**(2), 2839–2849 (2024). https://doi.org/10.1007/s40747-023-01304-z
25. He, L., Zhou, Y., Liu, L., Zhang, Y., Ma, J.: Application of the YOLOv11-seg algorithm for AI-based landslide detection and recognition. Sci. Rep. **15**(1), 12421 (2025). https://doi.org/10.1038/s41598-025-95959-y
26. Rani, M.U., Rajesh, A., Rao, V.V., Prasanna, K.L., Vasanthi, P.: Extreme small object detection with transformer-based local and global context attention. Infrared Phys. Technol. 106251 (2025). https://doi.org/10.1016/j.infrared.2025.106251
27. Liu, L., et al.: A lightweight deeplab v3+ network integrating deep transitive transfer learning and attention mechanism for burned area identification. Sci. Rep. **15**(1), 15969 (2025). https://doi.org/10.1038/s41598-024-66060-7
28. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal visual object classes challenge: a retrospective. Int. J. Comput. Vision **111**(1), 98–136 (2014). https://doi.org/10.1007/s11263-014-0733-5
29. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256. JMLR Workshop and Conference Proceedings (2010)
30. Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F.J., Marín-Jiménez, M.J.: Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recogn. **47**(6), 2280–2292 (2014). https://doi.org/10.1016/j.patcog.2014.01.005

Advanced and Emerging Applications

Beam Forming and Super-Grouping for Seismic Data Enhancement in 5D Domain

Maxim Protasov[1(✉)], Maxim Dmitriev[2], Young Seo Kim[2], and Denis Sabitov[3]

[1] Institute of Petroleum Geology and Geophysics, 3, Koptuga st, Novosibirsk 630090, Russia
protasovmi@ipgg.sbras.ru
[2] EXPEC Advanced Research Center, Saudi Aramco, Dhahran 31311, Saudi Arabia
[3] Aramco Innovations LLC, Saudi Aramco, Moscow 117105, Russia

Abstract. In this study, we have developed a novel seismic data enhancement approach along with its corresponding software implementation that enables comprehensive processing of data in the 5D domain. The primary objective of this development is to establish a robust and computationally efficient methodology capable of producing optimal data enhancement results. Our approach is fundamentally based on the integration of two key techniques: seismic data super-grouping and non-linear beamforming. While each of these techniques individually operates using two spatial coordinates and one time coordinate, it is important to note that complete seismic datasets actually contain four spatial coordinates. To address this dimensionality challenge, we have developed an innovative integration of these techniques that effectively accounts for all four spatial coordinates simultaneously, thereby providing a more comprehensive and robust solution. A significant portion of our research has focused on systematically investigating the most suitable data domains for optimal performance when applying super-grouping and beamforming in combination. Through this investigation, we have identified and characterized several effective domain configuration options. The primary domain configuration employs the cross-spread domain for beamforming operations while utilizing its orthogonal counterpart for super-grouping. An alternative important configuration option utilizes the offset domain for beamforming while employing the central point domain for super-grouping operations. Furthermore, we have developed optimal data sampling strategies that are specifically tailored to different data geometries. Additionally, we have implemented several significant optimizations that enhance the standard application of non-linear beamforming, improving both its efficiency and effectiveness. To complete this comprehensive development, we have created specialized software that implements our enhanced methodology. The performance and capabilities of both our algorithmic approach and its software implementation have been rigorously evaluated through extensive testing. These evaluations have included both synthetic datasets and real seismic data acquired from the Arabian Plate. The results of these tests conclusively demonstrate the proposed methodology's capabilities and effectiveness, validating its practical utility for seismic data enhancement applications.

Keywords: Beam forming · Super-grouping · 5D seismic data enhancement

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 599–612, 2026.
https://doi.org/10.1007/978-3-032-30491-9_37

1 Introduction

In land seismic exploration, strong scattering caused by small- to medium-scale heterogeneities in the near-surface greatly degrades the quality of seismic images [1, 2]. Standard seismic time processing procedures—such as surface-consistent deconvolution, statics estimation, and velocity analysis—require reliable prestack signal data. However, their application in areas with complex near-surface conditions often yields unreliable results because the derived parameters are based on noise rather than signal. To address this issue, it is essential to significantly improve the signal in the prestack data before evaluating any processing parameters [3, 4].

Local stacking is one of the most powerful methods for data enhancement. For example, supergrouping adjacent traces in different domains after applying normal moveout (NMO) correction [5] has proven highly effective for enhancing reflections. Data-driven local stacking of neighboring traces can substantially improve signals of varying complexity. Existing methods based on this technique include common reflecting surface (CRS) stacking [6–10] and multifocusing [11–14]. One of the most effective data enhancement procedures involves local summation of the initial data along surfaces defined by locally coherent seismic events [15].

A developed version of this method employs an operator-oriented scheme to enhance the useful signal while suppressing regular and irregular noise [15]. In this approach, the wavefront of the useful signal is locally approximated by a second-order surface, which can be viewed as a Taylor series expansion of the seismic event arrival time around an arbitrary point. The expansion coefficients (local kinematic attributes) are estimated by maximizing the coherence measure (semblance) calculated along the second-order surface. A key feature of this scheme is that the most computationally intensive step—estimating these parameters—is performed on a sparse grid. Once the parameters are determined, the "signal gain" operator is defined, which involves summing adjacent traces along a two-dimensional hodograph.

Bakulin et al. [16] proposed nonlinear beamforming (NLBF) in the cross-spread domain—an approach for enhancing challenging prestack data acquired in modern 3D land seismic surveys. The advantages of the cross-spread domain for data enhancement are demonstrated with real-data examples, highlighting its signal-preservation and signal-enhancement properties.

In this study, we further develop the beamforming procedure by introducing a novel seismic data enhancement method and corresponding software designed to process data in the 5D domain. The primary goal is to deliver a robust and computationally efficient approach that enhances data with a high signal-to-noise ratio. We develop the corresponding software and evaluate the algorithm using both synthetic data and real seismic data from the Arabian Plate, demonstrating significant improvements.

2 5D Data Enhancement Method

The 5D seismic data enhancement method is founded upon two fundamental techniques: super-grouping of seismic data and beamforming (collectively referred to as SGBF 5D). While conventional seismic data contains four spatial dimensions (4D), these individual techniques each utilize only two spatial coordinates (2D) plus a time coordinate. Naturally, the number of available traces in the complete 4D domain significantly exceeds that in any 2D subset. To quantify this difference precisely, we have conducted a detailed investigation of typical acquisition geometries.

2.1 Trace Statistics Analysis

Our analysis examines trace number statistics for single-sensor real data geometry [17, 18] (Fig. 1). The acquisition parameters are as follows: source grid spacing: dx_s =75 m, dy_s =12.5 m; receiver grid spacing: dx_r =12.5 m, dy_r = 150 m. In our terminology 4D geometry employs all four coordinates (x_s, x_r, y_s, y_r), 2D geometry fixes two coordinates (x_r, y_s) while varying the remaining two (x_r, y_s). The data enhancement procedure can be implemented in two distinct approaches: 2D Case - varying only two coordinates (x_r, y_s) within the aperture, and this is the implementation used in our developed code; 4D case - varying all four coordinates (x_s, x_r, y_s, y_r) within the aperture. We have computed and compared the number of traces within the aperture for both cases, examining how these values scale with aperture size (Fig. 2). Key observations include: in 4D approach 1000 traces require 125 m aperture, 10000 traces require 150 m aperture, 600 m aperture yields 1000000 traces; in 2D approach: 1000 traces require 200 m aperture, 10000 traces require 600 m aperture, 600 m aperture yields 10000 traces.

Our trace statistics analysis demonstrates that the 4D spatial domain provides substantially more traces (over tenfold) within a given aperture compared to the 2D spatial domain. This finding strongly supports the superiority of the 5D approach, which incorporates all four spatial coordinates (source and receiver) plus time, for seismic data enhancement applications. The 5D methodology offers two key advantages: potential for significantly higher signal-to-noise ratio in processed data; more comprehensive utilization of available seismic information. To fully realize these benefits, we have developed an innovative solution that combines both 2D techniques (beamforming and super-grouping) in a manner that effectively accounts for all four spatial coordinates. This integrated approach provides: more robust noise attenuation; better preservation of signal fidelity; improved resolution in the final processed data. The implementation maintains computational efficiency while delivering superior results compared to conventional 2D processing methods, particularly in challenging acquisition environments or when working with noisy datasets.

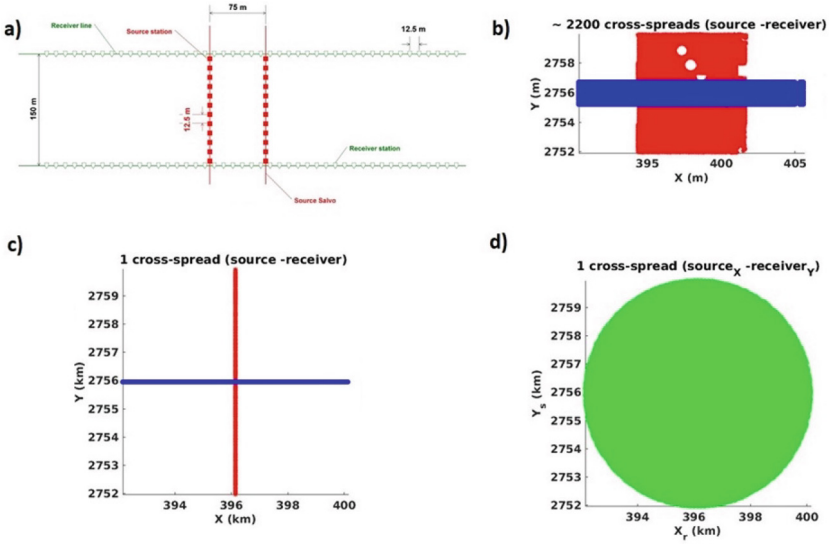


Fig. 1. Single sensor real data geometry: a) acquisition geometry; b) 4D source–receiver coordinates for ~2200 cross-spread gathers; c) 4D source–receiver coordinates for one cross-spread gather; d) 2D source_x–receiver_y coordinates for one cross-spread gather.

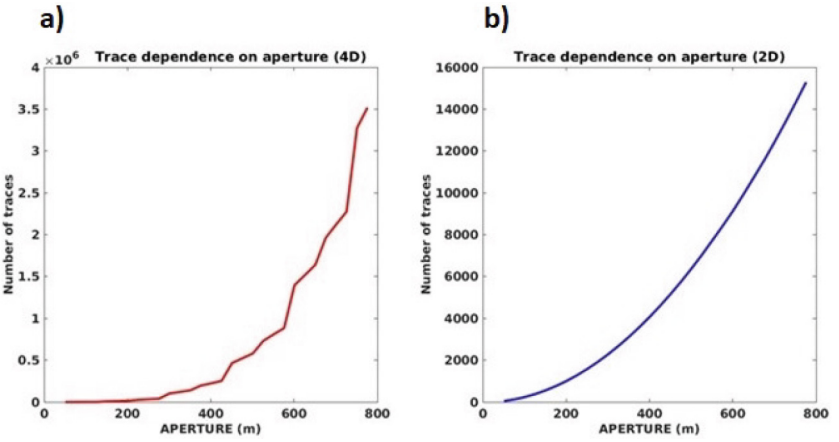


Fig. 2. Number of traces within aperture with respect to aperture size: a) 4D geometry; b) 2D geometry.

2.2 Super-Grouping and Beam Forming as a 5D Data Enhancement

From a technical perspective, 5D data enhancement can be effectively achieved through the implementation of super gathers. A super gather is created by combining several Common Depth Point (CDP) gathers into a single, more comprehensive gather. In our research, we have thoroughly investigated the most suitable data domains for the combined application of super-grouping and beamforming, presenting several viable options.

One primary configuration utilizes the cross-spread domain for beamforming operations while employing its orthogonal counterpart for super-grouping. Within the beamforming procedure in the cross-spread domain, we consider a second-order travel-time approximation in the 2D plane ($\boldsymbol{x_s}, \boldsymbol{y_r}$) around point ($\boldsymbol{x_{0s}}, \boldsymbol{y_{0r}}$):

$$\begin{aligned}\Delta t = t(x_s, x_r, y_s, y_r) - t(x_{0s}, x_{0r}, y_{0s}, y_{0r}) = A(x_s - x_{0s}) + B(y_r - y_{0r}) + \\ + C(x_s - x_{0s})(y_r - y_{0r}) + D(x_s - x_{0s})^2 + E(y_r - y_{0r})^2\end{aligned} \tag{1}$$

where coefficients ***A,B*** represent the wavefront dips, while coefficients ***C,D,E*** characterize the wavefront curvatures. However, when working with a CDP gather or a super-gather composed of multiple CDPs, the trace points ($\boldsymbol{x_r}, \boldsymbol{y_s}$) within such gathers are neither identical to each other nor equal to the parameter trace point ($\boldsymbol{x_{0r}}, \boldsymbol{y_{0s}}$). This observation implies that approximation (1) is incomplete for this scenario and should instead be considered in a 4D plane, thereby elevating the data enhancement process from 3D to 5D. To address this, we introduce the missing terms to approximation (1):

$$\begin{aligned}\Delta t = t(x_s, x_r, y_s, y_r) - t(x_{0s}, x_{0r}, y_{0s}, y_{0r}) = A(x_s - x_{0s}) + B(y_r - y_{0r}) + \\ + C(x_s - x_{0s})(y_r - y_{0r}) + D(x_s - x_{0s})^2 + E(y_r - y_{0r})^2 + \\ + A1(x_r - x_0r) + B1(y_s - y_0s) + D1(x_r - x_0r)^2 + E?1(y_s - y_0s)^2 \\ + C1(x_r - x_{0r})(y_s - y_{0s}) + C2(x_r - x_{0r})(y_r - y_{0r}) + C3(x_s - x_{0s})(y_s - y_{0s}) \\ + C4(x_r - x_{0r})(x_s - x_{0s}) + C5(y_s - y_{0s})(y_r - y_{0r})\end{aligned} \tag{2}$$

where coefficients ***A1,B1*** represent the wavefront dips in the domain orthogonal to the cross-spread domain, and coefficients ***D1,E1,C1,C2,C3,C4,C5*** characterize the corresponding wavefront curvatures. Implementing this complete travel-time approximation requires estimating nine additional parameters, which demands substantial computational resources. Alternatively, we can apply beamforming in this context and examine its implications for the complete travel-time approximation (2) in the 5D domain. This approach effectively reduces approximation (2) back to approximation (1), implying that the additional parameters (***A1,B1***, ***D1,E1,C1*** through ***C5***) are set to zero. This reduction means we perform summation in the orthogonal plane ($\boldsymbol{x_r}, \boldsymbol{y_s}$) using a horizontally flat approximation. When we consider how super-grouping operates, we recognize that it implicitly performs this operation in the orthogonal plane ($\boldsymbol{x_s}, \boldsymbol{y_r}$). Therefore, applying beamforming to a super-gather consists of two components: beamforming in the cross-spread domain; super-grouping in the domain orthogonal to the super-gather. This combined operation functions as a 5D data enhancement according to travel-time approximation (2), with parameters (***A,B,C,D,E***) being estimated while parameters (***A1,B1***, ***D1,E1,C1*** through ***C5***) are set to zero. We designate this integrated approach as Super Grouping Beam Forming (SGBF) 5D data enhancement.

Recognizing that super-grouping is inherently less precise than beamforming and is best suited for small apertures, we must consider alternative, more suitable operating domains for super-grouping. For super-gathers created from the nearest CDP gathers, the CDP domain emerges as the optimal choice for super-grouping due to the naturally small apertures resulting from the super-gather construction. This domain offers an additional advantage for horizontally layered geology, as the geological context perfectly justifies setting the additional parameters to zero. To demonstrate this mathematically,

let us examine 5D enhancement in the CDP-offset domain, defined by the following coordinates:

$$x_m = (x_r + x_s)/2, y_m = (y_r + y_s)/2; .h_x = (x_r - x_s), h_y = (y_r - y_s) \tag{3}$$

Analogous to our previous approach, we consider a second-order travel-time approximation in the 2D plane ($\boldsymbol{h_x}, \boldsymbol{h_y}$) around point ($\boldsymbol{h_{0x}}, \boldsymbol{h_{0y}}$):

$$\begin{aligned} &\Delta t = t\big(h_x, h_y, x_m, y_m\big) - t\big(h_{0x}, h_{0y}, x_{0m}, y_{0m}\big) = A(h_x - h_{0x}) + B\big(h_y - h_{0y}\big) + \\ &C(h_x - h_0x)\big(h_y - h_0y\big) + D(h_x - h_0x)^2 + E\big(h_y - h_0y\big)^2, \end{aligned} \tag{4}$$

where coefficients ***A,B*** represent wavefront dips, and ***C,D,E*** represent wavefront curvatures. Expanding this with the missing terms yields:

$$\begin{aligned} \Delta t = t\big(h_x, h_y, x_m, y_m\big) - t\big(h_{0x}, h_{0y}, x_{0m}, y_{0m}\big) = A(h_x - h_{0x}) + B\big(h_y - h_{0y}\big) + \\ + C(h_x - h_{0x})\big(h_y - h_{0y}\big) + D(h_x - h_{0x})^2 + E\big(h_y - h_{0y}\big)^2 + A1(x_m - x_{0m}) + \\ + B1(y_m - y_0m) + D1(x_m - x_0m)^2 + E1(y_m - y_0m)^2 + \\ + C1(x_m - x_{0m})(y_m - y_{0m}) + C2(x_m - x_{0m})\big(h_y - h_{0y}\big) + \\ + C3(h_m - h_{0m})(y_m - y_{0m}) + C4(x_m - x_{0m})\big(h_y - h_{0y}\big) + \\ + C5(y_m - y_{0m})\big(h_y - h_{0y}\big) \end{aligned} \tag{5}$$

where again the additional parameters (***A1,B1***, ***D1,E1,C1*** through ***C5***) are set to zero. This means we perform summation in the CDP plane ($\boldsymbol{x_m}, \boldsymbol{y_m}$) using a horizontally flat approximation. Since the super-gather is constructed from the nearest CDPs, this domain is particularly well-suited for super-grouping. Considering the geological context of the Arabian Plate, where reflectors are predominantly flat, this domain becomes an ideal candidate. Therefore, we also provide the option of SGBF 5D data enhancement in the CDP-offset domain.

2.3 Realization and Optimization of SGBF 5D Data Enhancement

When considering the theoretical context of SGBF 5D, there exists a clear and straightforward approach to implement the described method. The implementation requires taking the existing beamforming procedure and introducing specific modifications to enhance its functionality. First, we apply the beamforming procedure in both of the most effective domains for creating super-gathers, namely the cross-spread domain and the offset domain. To achieve this, we utilize the existing beamforming procedure in the cross-spread domain while simultaneously developing a new implementation for the offset domain. In these orthogonal domains, this approach results in simple stacking operations that do not require time corrections, which consequently means that super-grouping is automatically achieved within these domains. Additionally, as part of the SGBF 5D implementation, we introduce an optimization of the summation component within the beamforming procedure. This optimization involves eliminating unnecessary computations for new, unused traces within the enhanced super-gather, while retaining the computational processes only for the central CDP (Common Depth Point).

3 Numerical Investigations

3.1 Synthetic Data Set

In order to provide a feasibility study of the proposed and realized SGBF 5D, we create 3D synthetic data for realistic 3D acquisition geometry and for realistic 1D geological model, using analytic formulas and RMS velocities. The acquisition geometry simulates the standard acquisitions that are used in Saudi Aramco. The main acquisition parameters are taken with the following values: $\boldsymbol{dxs = 150\ m, dxr = 25\ m, dyr = 150\ m, dys = 25\ m}$. From that data we have created 1 central cdp gather, and 3 super gathers consisting of 3 by 3 cdp gathers, 5 by 5 cdp gathers, and 7 by 7 cdp gathers. At the end we add additive Gaussian noise with different level, i.e. 10% noise, and 100% noise.

Next, we start testing the developed code on the created data set. We choose rather small apertures for testing: $\mathbf{aper_sum_x = 150}\ \boldsymbol{m}$, $\mathbf{aper_sum_y = 150}\ \boldsymbol{m}$. We apply SGBF 5D using for the data with different noise level: 0%, 10%, 100%, and different number of gathers in super-gather: 1, 3 by 3, 5 by 5, 7 by 7, in the cross-spread domain (Fig. 3). Also, we apply SGBF 5D for the same data but in the offset domain (Fig. 4).

First of all, we should examine the results of data enhancement application when there is no noise present. Visual observation shows that for all super-gathers and in both domains (cross-spread and offset), the results of data enhancement appear appropriate - that is, nearly identical to the input data. This indicates that the data enhancement process does not distort or destroy the useful reflected signal. However, some minor differences do exist, which we will discuss later in greater detail.

Furthermore, visual inspection clearly demonstrates an improvement in data quality when comparing the enhanced data to the original noisy input data. This improvement is evident in both domains and is observable across nearly all super-gathers. Additionally, the visual analysis reveals that the quality of noise reduction improves as the number of gathers within each super-gather increases. This relationship holds true for both the cross-spread and offset domains. It is also visually apparent that for particularly noisy datasets, a greater number of gathers per super-gather is required to achieve satisfactory enhancement results, again applying equally to both domains.

However, these visual observations only provide qualitative assessments. To make more precise comparisons, especially in cases where visual differentiation is challenging, we conducted quantitative analysis by computing correlation coefficients between

the ideal data (known in our synthetic case) and the enhanced data obtained under various conditions. These conditions included different domains, varying noise levels, and differing numbers of gathers per super-gather. The resulting correlation coefficients (Table 1) not only confirm our visual observations but also establish the maximum achievable enhancement values (those obtained with noise-free input data).

The table reveals several important findings: For smaller super-gather configurations (specifically 1 gather and 3×3 gathers), the cross-spread domain yields the best achievable results. However, for larger configurations (5×5 and 7×7 gathers), the offset domain produces superior outcomes. This pattern persists when processing noisy data, with one crucial exception: when noise levels become excessive, smaller super-gather

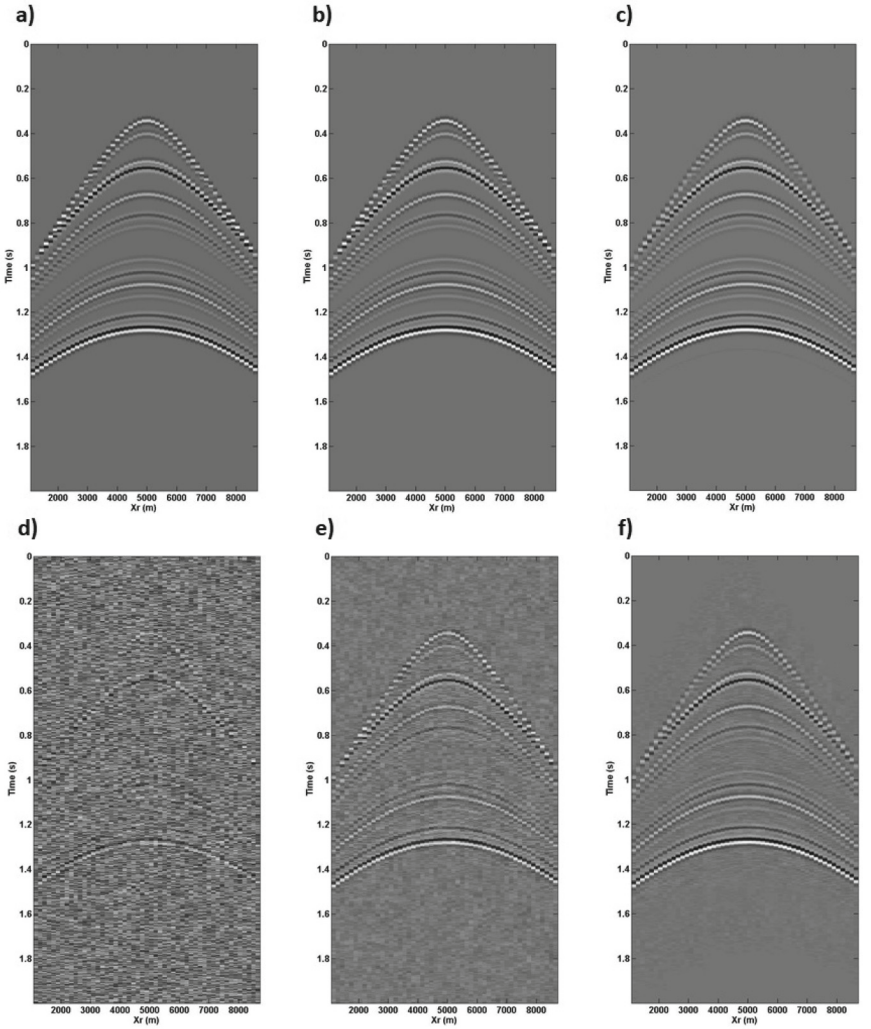


Fig. 3. (a) The input data without noise; (b) the data after 5D enhancement of 3 by 3 supergather in crosspread domain; (c) the data after 5D enhancement of 7 by 7 supergather in crosspread domain; (d) input data with 100% noise; (e) the noise data after 5D enhancement of 3 by 3 supergather in crosspread domain; (f) the noise data after 5D enhancement of 7 by 7 supergather in crosspread domain.

configurations prove inadequate, resulting in unacceptably low correlation coefficients. In such cases, larger super-gathers become necessary to achieve proper enhancement. Furthermore, when working with significantly noisy data, the offset domain - a new option in SGBF 5D - should be selected to obtain optimal results. These empirical findings align perfectly with the theoretical framework described earlier.

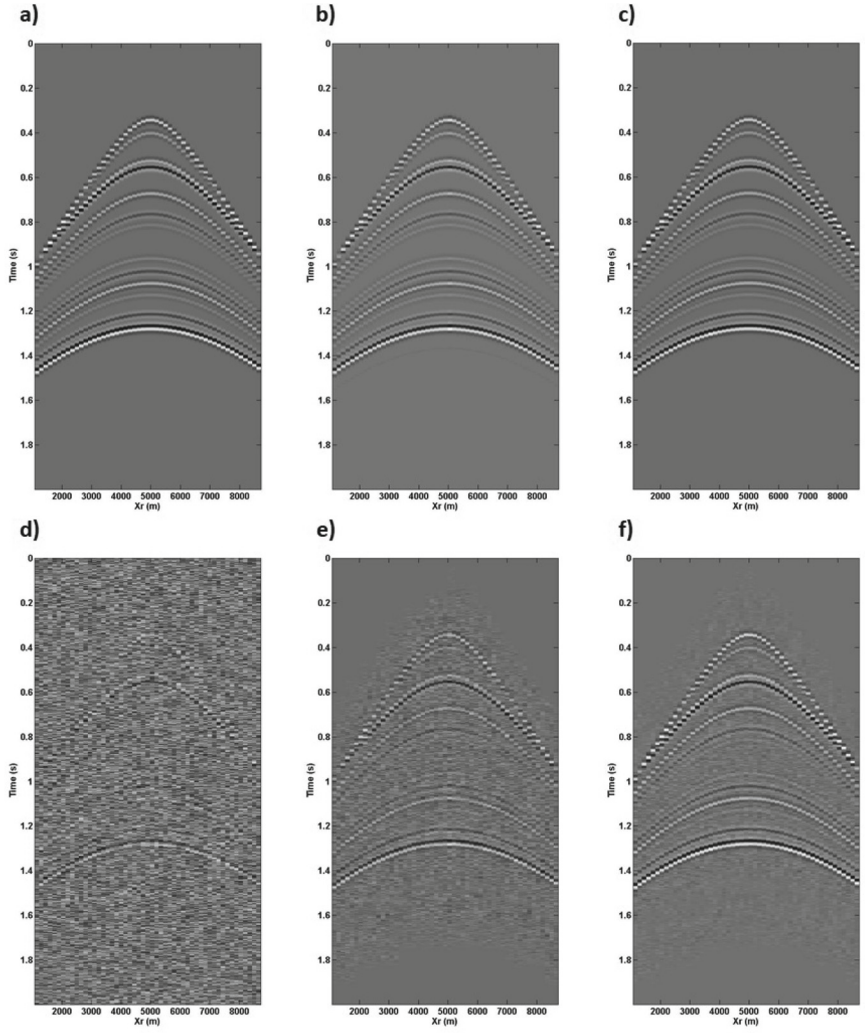


Fig. 4. (a) The input data without noise; (b) the data after 5D enhancement of 3 by 3 supergather in cdp-offset domain; (c) the data after 5D enhancement of 7 by 7 supergather in cdp-offset domain; (d) input data with 100% noise; (e) the noise data after 5D enhancement of 3 by 3 supergather in cdp-offset domain; (f) the noise data after 5D enhancement of 7 by 7 supergather in cdp-offset domain.

Additionally, we compared computational times across the various scenarios described above. The complete set of computational times for different scenarios is presented (Table 2). Of particular importance is the comparison between optimized SGBF 5D and non-optimized processing. The results demonstrate that the optimization provides computational acceleration of the summation process by a factor of 2 to 7, depending on the operation domain, number of OPM threads, and number of CDPs in

Table 1. Correlation coefficient between ideal data and data after SGBF 5D application with different noise level, different super-gathers and two different domains: cross-spread and offset.

	Noise 0%			Noise 10% 6.2 dB			Noise 100% -8.6 dB	
	Cross-spread	Offset		Cross-spread	Offset		Cross-spread	Offset
1 by 1	0.999	0.975	1 by 1	0.925	0.937	1 by 1	0.236	0.321
3 by 3	0.997	0.986	3 by 3	0.993	0.982	3 by 3	0.783	0.738
5 by 5	0.982	0.996	5 by 5	0.982	0.994	5 by 5	0.919	0.884
7 by 7	0.950	0.998	7 by 7	0.950	0.997	7 by 7	0.917	0.939

the super-gather. Another noteworthy observation is that when using the optimization, computational times become similar between the offset and cross-spread domains.

Table 2. Computational times of SGBF 5D application when optimization of summation is turned on and turned off with different super-gathers and two different domains: cross-spread and offset (38 OMP threads, different number of OMP provides different speed up, the same tests on 1 OMP threads provides speed up to 7 times).

	Without optimization			With optimization	
	Cross-spread	Offset		Cross-spread	Offset
1 by 1	1 s	3 s	1 by 1	1 s	3 s
3 by 3	9 s	6 s	3 by 3	3 s	5 s
5 by 5	36 s	20 s	5 by 5	9 s	9 s
7 by 7	108 s	48 s	7 by 7	26 s	18 s

3.2 Real Data Example

Finally, we present test results obtained from real seismic data acquired in the Arabian Plate, which demonstrate both the capabilities and effectiveness of our developed approach and its corresponding software implementation. The acquisition system parameters for this survey were as follows: shot spacing along the x-direction was 75 m; shot spacing along the y-direction was 25 m; receiver spacing along the x-direction was 25 m; receiver spacing along the y-direction was 75 m; with a CDP grid resolution of 12.5 m by 12.5 m. The dataset contains numerous different types of noise, making it particularly challenging to identify and trace reflected wave events (Fig. 5). We have applied our developed SGBF 5D methodology to process various super-gather configurations across different domains. In general terms, these real-data results provide clear confirmation of the conclusions we previously drew from synthetic data tests. More specifically, we have demonstrated significant improvements in signal-to-noise ratio through the application of SGBF 5D, even when using relatively small apertures in the CDP super-grouping domain (Fig. 5). These results strongly indicate the considerable

potential of our proposed approach for enhancing seismic data quality. Additionally, we would like to emphasize that this methodology shows promising potential for application in robust signal-to-noise ratio estimation [19, 20], particularly when dealing with challenging seismic datasets.

4 Conclusions

We have developed a novel seismic data enhancement approach along with its corresponding software implementation that enables comprehensive processing of data in the 5D domain. This innovative methodology combines two powerful techniques: seismic data super-grouping and non-linear beamforming. By integrating these complementary techniques, we have created a robust solution that effectively accounts for all four spatial coordinates simultaneously. Our research has systematically investigated the most suitable data domains for optimal performance when applying super-grouping and beamforming in combination.

The cross-spread domain has been identified as the optimal choice for beamforming operations due to its superior effectiveness when working with two spatial coordinates. However, it should be noted that super-grouping, being inherently less precise than beamforming, demonstrates better performance when applied with smaller apertures. This fundamental difference has led us to consider alternative domains that may be more appropriate for super-grouping applications. Specifically, when creating super-gathers from the nearest CDP gathers, the CDP domain emerges as clearly the most effective operating domain for super-grouping, primarily because the construction of super-gathers in this domain naturally results in very small apertures. Another significant configuration we have examined employs the offset domain for beamforming while utilizing the central point domain for super-grouping operations. Based on comprehensive analysis of data geometry, we have developed optimal data sampling strategies for various scenarios. Furthermore, we have implemented several key optimizations to enhance the standard application of non-linear beamforming.

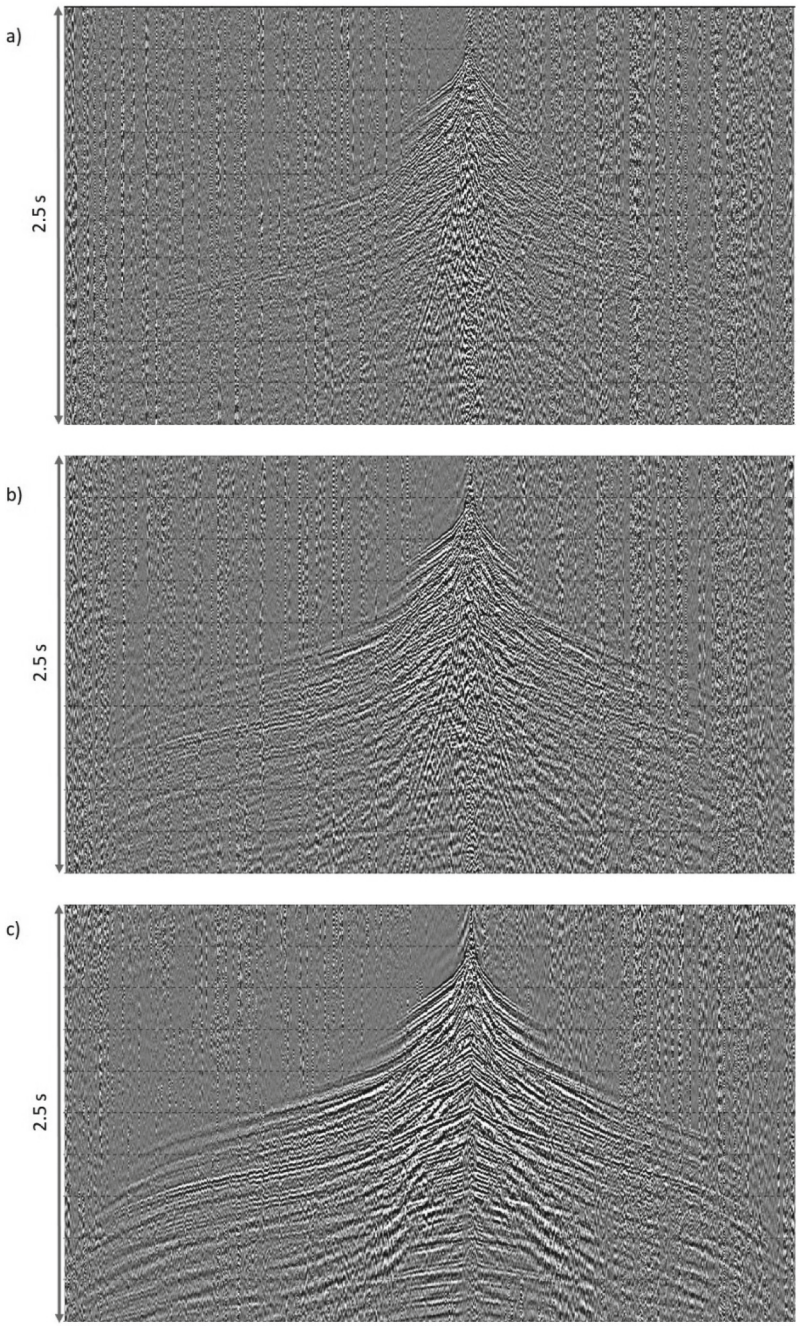


Fig. 5. CDP gather: (a) original data; (b) data after SGBF 5D, super-grouping in CDP domain (25 m by 25 m), beam forming in offset domain; (c) data after SGBF 5D, super-grouping in CDP domain (125 m by 125 m), beam forming in offset domain.

The SGBF 5D methodology has been thoroughly developed and rigorously tested using both synthetic gathers and real seismic data in both cross-spread and offset domains. Our optimized version of SGBF 5D demonstrates remarkable computational efficiency, achieving acceleration of the summation process by approximately 2–7 times, depending on three critical factors: the operation domain being used, the number of OPM threads employed, and the number of CDPs included in the super-gather. Extensive numerical testing has yielded several important findings regarding optimal configurations. The results indicate that the method potentially delivers the best performance when processing 3×3 super-gathers in the cross-spread domain, while for larger 5×5 and 7×7 super-gathers, the offset domain proves superior. More specifically, the numerical results demonstrate that appropriate enhancement results can be achieved for 3×3 and 5×5 super-gathers in both cross-spread and offset domains when processing data with low noise levels and using small apertures (150 m). Similarly, for datasets with moderate noise levels while maintaining small apertures (150 m), satisfactory results are obtained with 5×5 and 7×7 super-gathers in both domains. These empirically derived conclusions provide valuable practical guidance for initial implementation of the SGBF 5D methodology in real-world applications.

Based on the comprehensive results presented throughout this paper, we conclusively demonstrate that our developed approach, which synergistically combines beamforming and super-grouping techniques, represents a significant advancement in seismic data processing. This new 5D data enhancement tool successfully handles all spatial data coordinates while consistently delivering the best achievable processing results under various conditions.

Acknowledgments. Authors are grateful to Saudi Aramco for the support and for the permission to publish the results. This work is partially supported by the Russian Science Foundation within the framework of the project № 21-71-20002-П: Maxim Protasov provides investigations of the algorithm on seismic data.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Xie, X.-B., Ning, H., Chen, B.: How scatterings from small-scale near-surface heterogeneities affecting seismic data and the quality of depth image, analysis based on seismic resolution. In: SEG Technical Program Expanded Abstracts, pp. 4278–4282. SEG (2016)
2. Yilmaz, Ö.: Circumventing velocity uncertainty in imaging complex structures. Lead. Edge. **37**, 14–18 (2018)
3. Cordery, S.: An effective data processing workflow for broadband single-sensor single-source land seismic data. Lead. Edge. **39**, 401–410 (2020)
4. Bakulin, A., Neklyudov, D., Silvestrov, I.: Multiplicative seismic noise caused by small-scale near-surface scattering and its transformation during stacking. Geophysics. **87**(5), V419–V435 (2022)
5. Bakulin, A., Golikov, P., Dmitriev, M., Neklyudov, D., Leger, P., Dolgov, V.: Application of supergrouping to enhance 3D prestack seismic data from a desert environment. Lead. Edge. **37**, 306–313 (2018)

6. Jäger, R., Mann, J., Höcht, G., Hubral, P.: Common-reflection-surface stack: image and attributes. Geophysics. **66**, 97–109 (2001)
7. Baykulov, M., Gajewski, D.: Prestack seismic data enhancement with partial common-reflection-surface (CRS) stack. Geophysics. **74**, V49–V58 (2009)
8. Buzlukov, V., Baina, R., Landa, E.: Prestack data enhancement using local traveltime approximation. In: 72nd EAGE Conference and Exhibition Expanded Abstracts, pp. 372–376 (2010)
9. Buzlukov, V., Landa, E.: Imaging improvement by prestack signal enhancement. Geophys. Prospect. **61**, 1150–1158 (2013)
10. Xie, Y., Gajewski, D.: 5-D interpolation with wave-front attributes. Geophys. J. Int. **211**, 897–919 (2017)
11. Landa, E., Gurevich, B., Keydar, S., Trachtman, P.: Application of multifocusing method for subsurface imaging. J. Appl. Geophys. **42**, 283–300 (1999)
12. Landa, E., Keydar, S., Moser, T.J.: MultiFocusing revisited - inhomogeneous media and curved interfaces. Geophys. Prospect. **58**, 925–938 (2010)
13. Berkovitch, A., Deev, K., E.: Landa how non-hyperbolic multifocusing improves depth imaging. First Break. **27**, 95–103 (2011)
14. Rauch-Davies, M., Berkovitch, A., Deev, K., Landa, E.: Non-hyperbolic multifocusing imaging for pre-stack signal enhancement. In: SEG Technical Program Expanded Abstracts, pp. 4618–4622 (2013)
15. Hoecht, G., Ricarte, P., Bergler, S., Landa, E.: Operator oriented CRS interpolation. Geophys. Prospect. **57**, 957–981 (2009)
16. Bakulin, I., Silvestrov, M., Dmitriev, D., Neklyudov, M., Protasov, K.G., Dolgov, V.: Nonlinear beamforming for enhancement of 3D prestack land seismic data. Geophysics. **85**, V283–V296 (2020)
17. Cooper, N.: A world of reality—designing land 3D programs for signal, noise, and prestack migration. Lead. Edge. **23**, 1007–1014 (2004)
18. Vermeer, G.J.O.: 3D Seismic Survey Design. Society of Exploration Geophysicists (2012)
19. Neidell, N., Taner, M.: Semblance and other coherency measures for multichannel data. Geophysics. **36**, 482–497 (1971)
20. Liu, Y., Li, C.: Some methods for estimating the signal/noise ratio of seismic data. Oil Geophys. Prospect. **32**, 257–262 (1997)

Quantile Regression for Soil Organic Carbon: A Distributional Analysis of Soil Variability

Edson L. Almeida[1], Ian Alexander Zahner McIntosh[1], Joel André Ferreira dos Santos[1], Ana Carolina Ferreira[2], Angela Ferreira[3], Jorge Soares[1], Marcos Bacis Ceddia[2], and Diego Brandão[1(✉)]

[1] Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, Rio de Janeiro, Brazil
{edson.landim,ian.macintosh,joel.santos,jorge.soares, diego.brandao}@cefet-rj.br

[2] Universidade Federal Rural do Rio de Janeiro, Seropédica, Brazil

[3] Research Center in Digitalization and Intelligent Robotics, Instituto Politécnico de Bragança, Bragança, Portugal
apf@ipb.pt

Abstract. Climate change is affecting more places around the world, making it even more important to take new steps to stop it. In this context, soil organic carbon (SOC) not only plays a key role in biochemical cycles and soil fertility, but it also helps control the climate by taking in and/or releasing CO2 into the air. However, SOC data show a lot of variation, which makes it hard to estimate because of the complicated interactions between soil-forming factors. Standard regression models that use the mean usually don't show how complex SOC variability really is. This article examines the relationship between soil color and SOC through a distributional model employing quantile regression. This method helps in understanding how soil color affects different parts of the soil organic carbon (SOC) distribution, mainly its values along the distribution quartiles, allowing the identification of extreme values below 10% and above 90%. The Brazilian Agricultural Research Corporation (Embrapa) made the data available, which comes from different biomes and land use systems. This makes sure that the data is not all the same. The findings demonstrate that linear regression can only identify a general correlation between soil color and SOC, yielding a low R^2 value of approximately 0.53. This means limited explanatory power and the existence of heteroscedasticity, as corroborated by piecewise linear regression analysis. Quantile regression, on the other hand, proved more effective in identifying non-linear patterns and differences in distribution, obtaining an R^2 of 0.94 for the 90% quantile.

Keywords: Soil Organic Carbon · Quantile Regression · Environmental Modeling

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 613–627, 2026.
https://doi.org/10.1007/978-3-032-30491-9_38

1 Introduction

Soil is one of the largest carbon reservoirs on the Earth's surface, storing more carbon than the air and plants. Thus, proper soil management is an important part of policies to combat climate change. Soil organic carbon (SOC) is an essential component of the environment, as it influences biogeochemical cycles, soil fertility, and climate regulation. In turn, soil organic carbon has a direct effect on water retention, nutrient availability, biological activity, and soil structural stability, affecting agricultural productivity and ecosystem resilience [1,2].

Accurately monitoring soil organic carbon (SOC) is an important task for sustainability and mitigating the effects of climate change, being directly associated with the United Nations Sustainable Development Goals (SDGs), especially SDG 2 (Zero Hunger), SDG 13 (Climate Action), and SDG 15 (Life on Land) [3].

Despite its importance, common practices for estimating soil organic carbon (SOC) involve laboratory techniques, including dry combustion and wet oxidation, which are expensive, time-consuming, and logistically complex for extensive use [4].

To overcome these limitations, over the years, several researchers have investigated statistical and computational methods to estimate SOC from more accessible attributes, such as soil color, texture, and depth, in addition to environmental variables. Among the soil attributes associated with SOC, soil color is recognized as a good indicator, as it is closely linked to the accumulation of organic material and mineral composition.

Despite recent advances in machine learning methods, much of the research still uses mean-based regression models, ranging from simple linear regression to approaches with linear regression trees [5,6]. These models offer an initial approximation; however, they often do not adequately represent the complexity of pedological processes, since soil organic carbon (SOC) data typically exhibit non-linearity, asymmetry, and heteroscedasticity [8,10].

In this context, a single global relationship may be inadequate to characterize the variability of SOC under different environmental and soil conditions. In this context, quantile regression appears as an alternative method, evaluating several quantiles of the distribution, including the lower and upper extremes, instead of just the mean. This suggests an interesting alternative, especially in the context of SOC, since the extremes of the distribution are frequently associated with soils poor or rich in carbon [8–10].

Even though this seems like an advantage, soil scientists don't often use quantile regression, especially when they are looking into the link between soil color and soil organic carbon (SOC). Most current research emphasizes predictive performance or utilizes advanced modeling techniques rooted in machine learning, whereas the distributional characteristics of soil organic carbon (SOC) are inadequately investigated.

This study examines the correlation between soil color and SOC using quantile regression. The main objective is to examine variations in this relationship

across different regions of the SOC distribution, which may contribute to understanding its variability.

This work makes two important contributions. First, it offers an empirical examination of the soil color–SOC relationship from a distributional viewpoint. Second, it shows that quantile regression is a strong and easy-to-understand tool for analyzing soil data that can be used for digital soil mapping and environmental monitoring.

The remainder of this paper is organized as follows. Section 2 presents some aspects of quantile regression. Section 3 presents the related work on SOC estimation and soil color analysis. Section 4 describes the dataset and the methodological approach, including the regression models employed. Section 5 presents and discusses the experimental results. Finally, Sect. 6 concludes the paper and outlines directions for future research.

2 Theoretical Background

Modeling soil organic carbon (SOC) is quite hard because soil systems are naturally very different from each other. Climate, parent material, topography, organisms, time, and land use are all examples of soil-forming factors that interact in complicated ways to change SOC [7]. Consequently, SOC data frequently display nonlinearity, asymmetry, and heteroscedasticity, thereby constraining the efficacy of conventional regression methods predicated on the conditional mean.

Ordinary least squares (OLS) and other classical regression models try to find the conditional expectation $\mathbb{E}(Y \mid X)$ by minimizing the squared errors. Researchers like these models because they are simple to use and understand, but they rely on assumptions such as linearity and constant variance. In environmental and pedological datasets, these assumptions are often not met, which can lead to pictures of the processes that are going on that are biased or not complete.

Quantile regression extends classical regression by facilitating the estimation of conditional quantiles of the response variable. Quantile regression does not just model the mean; it also estimates the conditional quantile function $Q_Y(\tau \mid X)$ for a given $\tau \in (0, 1)$. This lets us look at the relationship between predictors and response in a more complete way.

Given a dataset $\{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^p$ denotes the explanatory variables and $y_i \in \mathbb{R}$ signifies the response variable, the quantile regression estimator is formally defined as:

$$\hat{\beta}_\tau = \arg\min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} \rho_\tau(y_i - x_i^\top \beta), \tag{1}$$

where $\rho_\tau(u)$ is the asymmetric loss function, known as the pinball loss:

$$\rho_\tau(u) = \begin{cases} \tau u, & \text{if } u \geq 0, \\ (\tau - 1)u, & \text{if } u < 0. \end{cases} \tag{2}$$

This loss function treats underestimating and overestimating differently, which lets the model focus on certain parts of the distribution. When $\tau = 0.5$, the model is median regression, which is better at dealing with outliers than mean-based methods.

One of the best things about quantile regression is that it can show how things affect different people in different ways. Classical regression assumes a constant relationship between predictors and response, while quantile regression allows model parameters to vary across quantiles. This feature makes it easier to find different regimes in the data, where explanatory variables may have different effects depending on the level of SOC.

Another important thing about quantile regression is that it can easily handle heteroscedasticity. The method doesn't assume that the variance is always the same. Instead, it models how the response variable's dispersion changes with the predictors. Diverging quantile curves show that things are becoming less stable, while converging curves show that things are becoming more stable.

These characteristics are crucial in soil science. SOC is known to change a lot based on the environment, and the processes that affect soils with low carbon may be different from those that affect soils with high carbon. Quantile regression provides a distributional perspective that goes beyond average trends by modeling several quantiles at once.

Classical quantile regression is generally characterized as a linear model; nonetheless, extensions such as polynomial formulations can proficiently depict nonlinear relationships between predictor and response variables. Quantile regression offers a robust and comprehensible method for analyzing complex environmental data, even in its simplest form.

In general, quantile regression is a way to model distributions that gives a better picture of how SOC behaves. It helps us understand soil processes better by letting us look at variability, asymmetry, and extreme values. It is also a useful alternative to traditional mean-based models.

3 Related Work

There have been a lot of different ways to study how to figure out how much soil organic carbon (SOC) there is. Some of these methods are pedological knowledge, field measurements, and computer-based methods. People have known for a long time that the color of soil can help you guess how much organic matter it has. Initial research showed that both field and lab tests of soil color can give very accurate estimates of SOC [18]. More research confirmed this link, showing that soil color can show how the environment is doing, such as where wetlands are located [11].

Better sensing technologies have made it possible to use a lot more data sources to figure out SOC. A lot of work has been done to figure out soil organic carbon (SOC) from soil reflectance characteristics using spectroscopy-based methods [14]. Satellite images, especially the Sentinel-2 time series, have also made it possible to map SOC and other soil properties in great detail [13,25].

Researchers have also looked into how unmanned aerial systems and multispectral images can help improve spatial resolution in landscapes that aren't all the same [16]. Low-cost sensing methods like mobile phone images and color sensors have also become useful for estimating SOC in places where there aren't many resources [19,20,27].

At the same time, machine learning methods have been increasingly used to predict SOC, often obtaining better results by finding non-linear relationships between environmental variables, especially when associated with spectral data and remote sensing data [12,15]. However, these methods often function as black boxes, which makes it difficult to understand the processes that are actually occurring and driving that soil state.

Numerous studies have shown that the correlation between soil color and soil organic carbon (SOC) is affected by different factors besides organic matter content. Changes in mineral composition, soil texture, and environmental variables can alter soil color without altering SOC, making modeling a somewhat more complicated process [23,26,30]. Geospatial analyses have demonstrated that the distribution of SOC is strongly influenced by land use and landscape characteristics, thus pointing to the need for methodologies capable of accommodating heterogeneous data [29].

A common limitation across much of the literature is the dependence on mean-based modeling methods, which assume a homogeneous relationship between predictors and SOC. However, SOC data are frequently characterized by asymmetry, nonlinearity, and heteroscedasticity, making average-based models insufficient to fully describe the underlying processes [7]. In this context, models that focus solely on the conditional mean may overlook important distributional patterns.

Quantile regression provides an alternative scheme to examine different regions of the conditional distribution of SOC. Through modeling multiple quantiles, this approach makes it possible to investigate how explanatory variables influence not only typical values as well as extreme conditions, which are particularly relevant in environmental applications [8–10]. In spite of these advantages, its application in soil science remains relatively limited, particularly in studies explicitly studying the relationship between soil color and SOC.

Consequently, the literature lacks a comprehensive distributional analysis of the soil color–SOC relationship. This paper aims to bridge the gap by using quantile regression to examine how the relationship varies across different parts of the SOC distribution. This gives a more detailed and understandable picture of soil variability.

4 Methodology

This section describes the dataset, preprocessing steps, and modeling strategies adopted to investigate the relationship between soil color and soil organic carbon (SOC). The proposed approach combines exploratory data analysis with mean-based and distributional regression models, enabling a comprehensive evaluation of how soil color influences SOC under varying conditions.

4.1 Dataset and Preprocessing

The dataset used in this study comprises soil profiles collected across Brazilian biomes, including agricultural, forest, and pasture areas. The data include measurements of soil organic carbon (SOC) and associated pedological attributes, with particular emphasis on soil color, which is used as the main explanatory variable in this study.

Due to the skewed nature of SOC data, a logarithmic transformation was applied to stabilize variance and reduce asymmetry. This transformation is commonly used in soil studies to improve the suitability of statistical modeling. After preprocessing, the dataset was analyzed to assess the distribution of SOC across land-use classes and to verify the absence of severe multicollinearity.

4.2 Exploratory Analysis

An exploratory data analysis was conducted to characterize the distribution of soil organic carbon (SOC) and investigate its relationship with available attributes, including soil color. Histograms were used to examine the distribution and variability of SOC across land-use classes. Furthermore, correlation coefficients were calculated to assess associations between SOC and other variables. This step aimed to support the selection of modeling approaches best suited to the data's characteristics.

4.3 Linear Regression Models

As a baseline approach, a simple linear regression model was fitted to describe the relationship between soil color index and SOC:

$$Y = \beta_0 + \beta_1 X + \varepsilon, \tag{3}$$

where Y represents the log-transformed SOC and X the soil color index.

To further investigate potential structural changes in this relationship, a segmented (piecewise) linear regression model was also considered. This approach allows different linear relationships to be estimated in distinct regions of the data, enabling the identification of possible regime shifts.

4.4 Quantile Regression

To capture the heterogeneous and distributional behavior of SOC, quantile regression was applied. Unlike classical regression, which estimates the conditional mean, quantile regression models the conditional quantile function:

$$Q_Y(\tau \mid X) = X^\top \beta_\tau, \tag{4}$$

where $\tau \in (0, 1)$ represents the quantile of interest.

The model parameters are estimated by minimizing the pinball loss function:

$$\hat{\beta}_\tau = \arg\min_{\beta} \sum_{i=1}^{n} \rho_\tau(y_i - x_i^\top \beta), \tag{5}$$

which allows different regions of the SOC distribution to be modeled independently.

In this study, quantile regression was used to analyze how the relationship between soil color and SOC varies across low-, median-, and high-carbon conditions. A polynomial specification was adopted to account for potential nonlinearities in the relationship.

4.5 Model Evaluation

We looked at the coefficient of fit (R^2) for linear models to see how well the model worked. This made it possible to see how well different specifications explained the data.

The focus of the quantile regression analysis was on how well the model could capture distributional patterns, including nonlinearity and heteroscedasticity. We checked the model by looking at the difference between the predicted and observed values and the shape of the estimated quantile curves.

5 Results and Discussion

This section presents the results obtained using the described methodology. Initially, we present a brief exploratory analysis of the data and examine the degree of asymmetry and kurtosis. In the following analysis, we evaluate the correlation matrix in the context of the three land uses: pasture, forest, and agriculture. The final step involved the use of regression models, namely linear, piecewise linear, and quantile. This allowed us to verify the relationship between soil color and soil organic carbon (SOC).

5.1 Exploratory Analysis

The exploratory analysis showed that SOC values varied significantly between land use classes, both in the different variables compared and after applying a logarithmic transformation. Figure 1 shows that the distributions present different patterns of asymmetry and dispersion. Forest areas tend to present higher SOC values, while agricultural and pasture systems present wider and more dispersed distributions. These patterns demonstrate that the soil is not homogeneous everywhere, which implies that modeling strategies must take this into account.

Table 1 presents the distribution of observations among biomes and land use types, demonstrating that the database has considerable diversity, despite a higher concentration of samples in the Atlantic Forest, Cerrado, and Caatinga biomes. This diversity contributes to the difficulty in determining the distribution of soil organic carbon (SOC), indicating heterogeneity in soil formation,

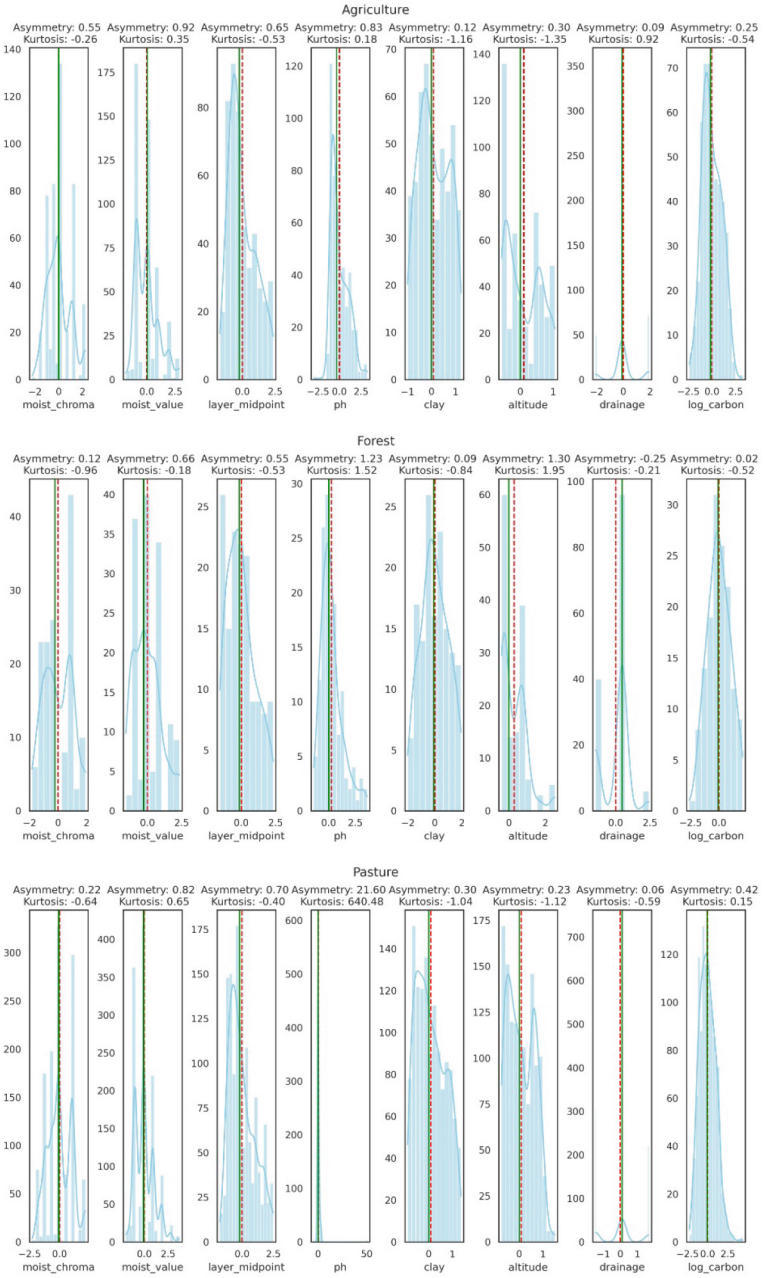


Fig. 1. Distribution of soil organic carbon content per land use.

which influences the relationship between the explanatory variables and our target variable.

Table 1. Number of records by biome and land use.

Biome	Agricultural	Forest	Pasture
Amazon	63	153	452
Caatinga	1050	38	1203
Cerrado	218	127	1681
Atlantic Forest	1136	405	1886
Pampa	39	12	96
Pantanal	0	5	0

The correlation analysis (Fig. 2) indicates that the relationships among the available attributes are generally moderate, suggesting the absence of severe multicollinearity. This supports the validity of the regression analysis and allows the use of soil color as a meaningful explanatory variable in the subsequent models.

5.2 Regression Models

As mentioned earlier, a simple linear regression model was fitted to describe the relationship between soil color index and soil organic carbon (SOC). Figure 3 shows the evaluation performed. The model demonstrates limited explanatory power, with a coefficient of determination of $R^2 = 0.53$. However, the pattern of residuals suggests heteroscedasticity, with increased dispersion at higher SOC values.

A complementary analysis was performed by individualizing the main color-related variables that influence SOC. Figure 4 shows that the model captures a general decreasing trend, indicating that darker soils tend to have a higher carbon content.

To investigate whether structural changes occur in the relationship between color and soil organic carbon, we used a piecewise linear regression model represented in Fig. 5. With this approach, we performed adjustments in different regions, aiming for the best linear relationship for each of them and thus trying to identify transition points where the system's behavior changes, or rather, where changes occur in the relationships between color and SOC.

As can be seen, the results indicate the presence of distinct regimes, characterized by different slopes along the intervals of the soil color index. This indicates that the influence of color on SOC is not uniform, but varies according to the range analyzed. In certain intervals, small variations in color are associated with more significant changes in SOC, while in others the relationship is weaker or more stable. This behavior shows that the relationship between

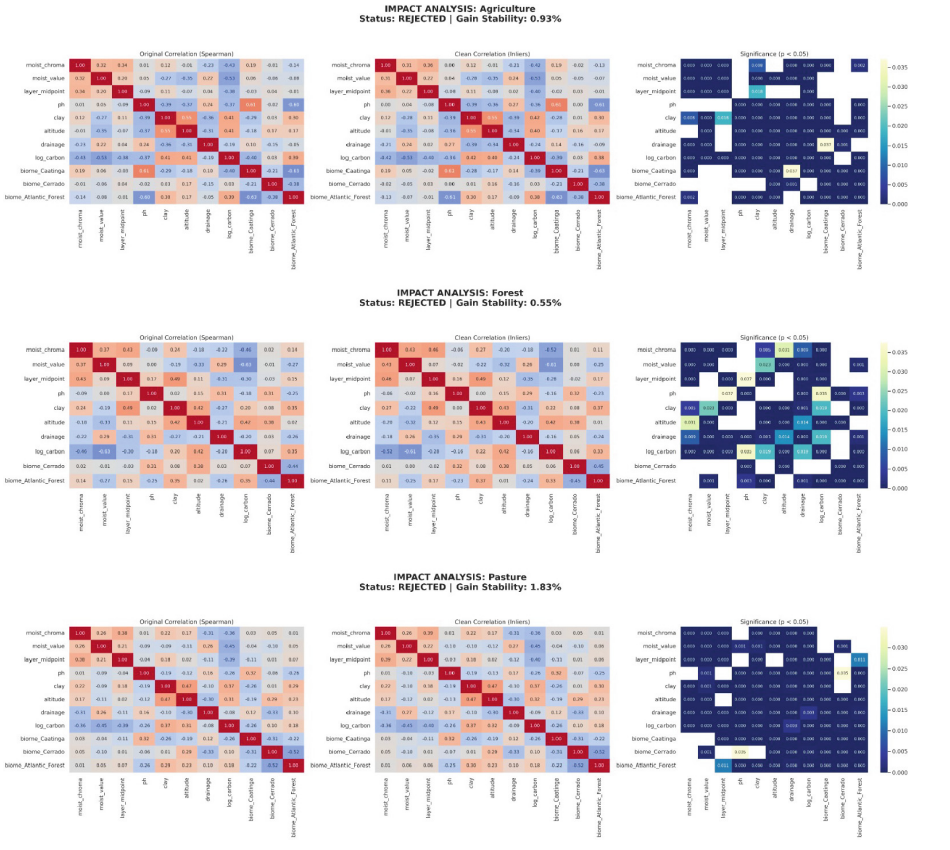


Fig. 2. Correlation matrix and statistical significance among the attributes.

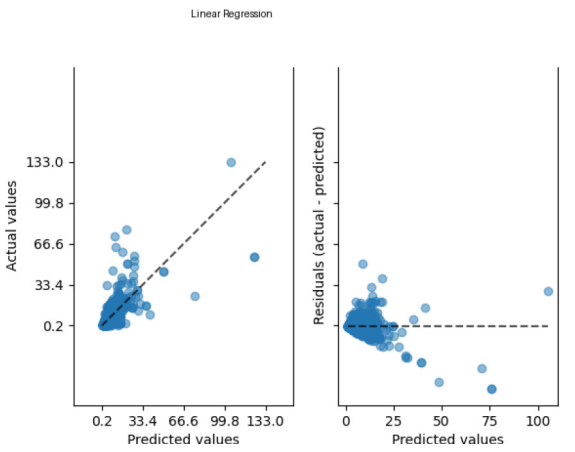


Fig. 3. Linear regression.

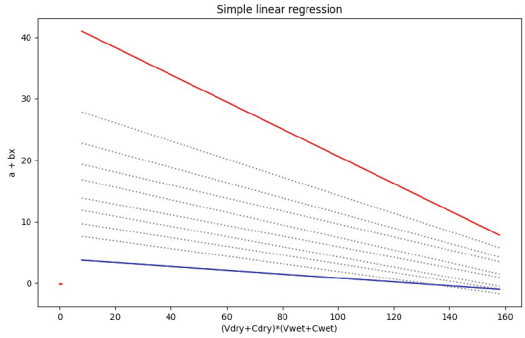


Fig. 4. Linear regression between soil color index and carbon content.

color and SOC is heterogeneous, probably due to other pedological and even environmental factors. For example, in some color ranges, the hue may be more directly associated with the accumulation of organic matter, while in others it may be strongly influenced by factors such as the presence of minerals, as already mentioned.

In order to better investigate this variation, quantile regression was employed. Figure 6 shows the result of the quantile regression adjustment; a coefficient of determination of $R^2 = 0.94$ was obtained. Furthermore, through the analysis of the quantile curves, it is observed that the relationship between soil color and SOC differs along the distribution.

In lower quantiles, the relationship seems to be more stable and less affected by changes in soil color. Higher quantiles, on the other hand, show more variability and stronger nonlinear patterns. This means that high SOC values are affected by more than just soil color. This behavior is in line with heteroscedasticity and backs up the idea that SOC is a distributional phenomenon.

Comparing the three approaches presented, important differences are observed in the ability of each to represent the relationship between soil color and SOC. Simple linear regression captures only a global trend, providing limited explanatory power, probably due to the presence of heteroscedasticity. In turn, the piecewise linear model achieves better results, as it can discern distinct regimes in the relationship between color and carbon, suggesting that the relationship is not uniform across all soil color possibilities. As already mentioned, not all dark soil is rich in carbon; it may have a high presence of minerals that give it a darker character. The problem is that this piecewise linear model still imposes linearity assumptions, limiting its flexibility. In contrast, quantile regression offers a more complete and robust structure, simultaneously modeling different regions of the SOC distribution. It obtained the best degree of fit, in addition to representing both the intensity and the variation of the relationship that differ between conditions through the divergence between the quantile curves. Thus, by comparing the three approaches, we get the impression that the dynamics of soil organic carbon (SOC) are so complex that they cannot

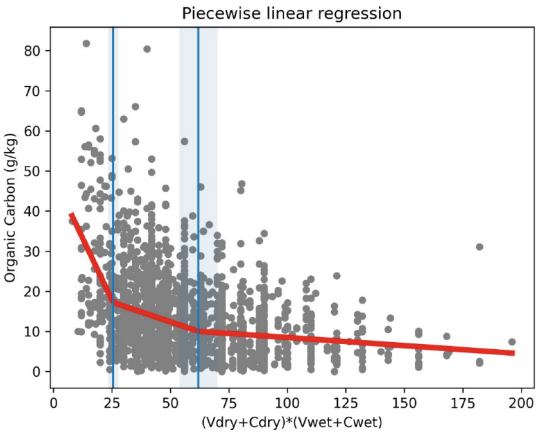


Fig. 5. Segmented linear regression between soil color index and carbon content.

be adequately described by a single functional relationship and, instead, must be interpreted from a viewpoint of distributions, in which different factors may prevail under conditions of low, medium, and high carbon content.

Overall, the results suggest that mean-based models provide only a partial description of the relationship between soil color and soil organic carbon (SOC). In contrast, quantile regression offers a more complete representation, being able to capture the variability, non-linearity, and even heterogeneity of the distribution.

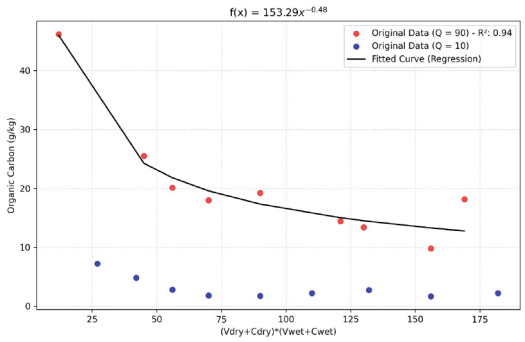


Fig. 6. Polynomial quantile regression between soil color index and carbon content.

6 Conclusion

Climate change has increasingly impacted the planet. Hurricanes, intense heat waves, and snowstorms have become more and more present in society's daily

life. Research that can assist in initiatives to help mitigate these climate effects is becoming increasingly important. In this context, soil plays a fundamental role, being able to help reduce polluting gases in the atmosphere.

This work examined the correlation between soil color and soil organic carbon (SOC) using approaches based on regression techniques focused on both the mean and the probability distribution itself. The results obtained indicate that linear regression identifies a general inverse correlation between soil color and SOC; however, its explanatory capacity is limited and insufficient to capture the heterogeneity of the data, due to the presence of heteroscedasticity. The more carbon in the soil, the greater the uncertainty of the relationship, which can be explained by factors that influence color, such as the presence of minerals that make the soil darker but without a significant presence of carbon.

The piecewise linear model improved our understanding of the relationship between color and carbon by defining specific regimes within this relationship. This result indicates that different processes can regulate the dynamics of soil organic carbon (SOC) in different soil color ranges. However, this approach relies on segmented assumptions, which compromises its ability to capture more complex patterns, especially in the presence of heteroscedasticity.

The use of quantile regression confirmed what the piecewise linear regression indicated about soil color not being uniform across all SOC levels. This quantile approach allowed modeling different parts of the conditional distribution; that is, the quantiles were able to capture the upper part of the distribution, where variability is greater and non-linearity is more prevalent, indicating that environmental and other pedological factors can affect SOC accumulation.

These results corroborate the literature, indicating that soil organic carbon (SOC) is a phenomenon of variability, and uncertainty is predominant. Another interesting result of this work is that it highlights the limitations of exclusively using mean-based models, emphasizing the effectiveness of quantile regression as a reliable and understandable alternative for soil data analysis. In this way, quantile regression proves to be a helpful technique for decision-makers in the agricultural, land management, and climate sectors, potentially providing a clearer view of soil organic composition (SOC) behavior.

As future work, we believe that incorporating other data, such as remote sensing data, in addition to climate data, could yield more accurate results for SOC estimation, especially with the use of quantile regression. Another possibility would be the use of quantile machine learning methods, which could provide greater explainability and robustness for understanding the problem.

Acknowledgments. The authors would like to thank the following Brazilian Agencies: FAPERJ, CAPES, and CNPq. DB thanks Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ) for the financial support through the grants E-26/210.798/2024 and E-26/204.262/2025.

Declaration of Generative AI in Scientific Writing. During the preparation of this work, the authors used Grammarly for English-language refinement. Following

the use of this tool, the authors reviewed and edited the manuscript as necessary and assume full responsibility for its content.

References

1. Lal, R.: Digging deeper: a holistic perspective of factors affecting soil organic carbon sequestration in agroecosystems. Glob. Change Biol. **24**(8), 3285–3301 (2018)
2. Lal, R.: Soil carbon management and climate change. Carbon Manag. **4**(4), 439–462 (2013)
3. Carlsen, L., Bruggemann, R.: The 17 United Nations' sustainable development goals: a status by 2020. Int. J. Sustain. Dev. World Ecol. **29**(3), 219–229 (2022)
4. Teixeira, P.C., Donagemma, G.K., Fontana, A., Teixeira, W.G.: Manual de métodos de análise de solo. Embrapa Solos (2017)
5. Pardelinha, A., et al.: Application of machine learning techniques in soil carbon stock estimation. In: SBBD (2024)
6. Ziviani, M.M., Pinheiro, É.F.M., Ceddia, M.B., Ferreira, A.C.S., Machado, F.S.: Carbon and nitrogen stocks and soil organic matter persistence. Soil Syst. **8**(2), 65 (2024)
7. Somarathna, P.D.S.N., Minasny, B., Malone, B.P.: More data or a better model? Soil Sci. Soc. Am. J. **81**(6), 1413–1426 (2017)
8. Nikou, M., Tziachris, P.: Prediction and uncertainty capabilities of quantile regression forests. ISPRS Int. J. Geo Inf. (2022)
9. Uduwerella, H., Iddawela, A.U., Kadupitiya, H.K.: Quantile regression for soil property mapping. Tropical Agriculture (2023)
10. Teleki, M.: Spatial distribution of soil organic carbon using quantile regression. Appl. Environ. Soil Sci. (2025)
11. Pretorius, M.L., Van Huyssteen, C.W., Brown, L.R.: Soil color indicates carbon and wetlands: developing a color-proxy for soil organic carbon and wetland boundaries on sandy coastal plains in South Africa. Environ. Monit. Assess. **189**(11), 1–18 (2017). https://doi.org/10.1007/s10661-017-6249-z
12. Salgado, L., Forján, R., Rodríguez-Pérez, J.R., et al.: Soil organic carbon fractionation assessment in areas with high fire activity using diffuse spectroscopy and tree-based machine learning algorithms. Earth Syst. Environ. (2025). https://doi.org/10.1007/s41748-024-00564-0
13. Castaldi, F., Koparan, M.H., Wetterlind, J., et al.: Assessing the capability of Sentinel-2 time-series to estimate soil organic carbon and clay content at local scale in croplands. ISPRS J. Photogramm. Remote. Sens. (2023). https://doi.org/10.1016/j.isprsjprs.2023.03.016
14. Amare, T., Hergarten, C., Hurni, H., et al.: Prediction of soil organic carbon for Ethiopian highlands using soil spectroscopy (2013). https://doi.org/10.1155/2013/720589
15. Das, A., Sinha, A., Dasgupta, S., et al.: Towards a two-stage Low-cost soil carbon estimation over large area using proximal sensing and digital soil mapping. Soil Adv. (2026). https://doi.org/10.1016/j.soilad.2026.100101
16. Wehrhan, M., Sommer, M.: A parsimonious approach to estimate soil organic carbon applying unmanned aerial system (UAS) multispectral imagery and the topographic position index in a heterogeneous soil landscape. Remote Sens. (2021). https://doi.org/10.3390/rs13183557

17. García-Ruiz, R., Bautista, F., Sánchez-Hernández, R., Goguitchaichvili, A.: Método colorimétrico para estimar la materia orgánica del suelo en áreas de karst. Ecosistemas y Recursos Agropecuarios (2022). https://doi.org/10.19136/era.a9n1.3189
18. Wills, S.A., Burras, C., Sandor, J.: Prediction of soil organic carbon content using field and laboratory measurements of soil color (2007). https://doi.org/10.2136/SSSAJ2005.0384
19. Fu, Y., Taneja, P., Lin, S., et al.: Predicting soil organic matter from cellular phone images under varying soil moisture. (2020). https://doi.org/10.1016/j.geoderma.2019.114020
20. Swetha, R., Chakraborty, S.: Combination of soil texture with Nix color sensor can improve soil organic carbon prediction. (2021). https://doi.org/10.1016/j.geoderma.2020.114775
21. Herranz-Luque, J., González-Canales, J., Martín-Sanz, J.P., et al.: Monitoring land management practices using Vis–NIR spectroscopy provides insights into predicting soil organic carbon and limestone levels in agricultural plots. Agronomy (2024). https://doi.org/10.3390/agronomy14061150
22. Zelenak, E.M.: Relationship between Munsell color value and organic carbon content in Montana soils (1995)
23. Chen, F., Kissel, D., West, L., Adkins, W.: Field-scale mapping of surface soil organic carbon using remotely sensed imagery (2000). https://doi.org/10.2136/SSSAJ2000.642746X
24. Jarmer, T., Hill, J., Mader, S.: The use of hyperspectral remote sensing data for the assessment of chemical properties of dryland soils in SE-Spain (2007)
25. Wetterlind, J., Simmler, M., Castaldi, F., et al.: Influence of Soil Texture on the Estimation of Soil Organic Carbon From Sentinel-2 Temporal Mosaics at 34 European Sites. Eur. J. Soil Sci. (2025). https://doi.org/10.1111/ejss.70054
26. Gorban', V., Huslystyi, A., Espejo, J., Bilova, N.: Prediction of SOC in calcic chernozem in the steppe zone of Ukraine using brightness and colour indicators. Ekológia (Bratislava) (2021). https://doi.org/10.2478/eko-2021-0034
27. Raeesi, M., Zolfaghari, A., Yazdani, M., et al.: Prediction of soil organic matter using an inexpensive colour sensor in arid and semiarid areas of Iran. Soil Res. (2019). https://doi.org/10.1071/SR18323
28. Zheng, F.: Using spectral reflection and multi-dye layer pixel values to quantify soil patterns for assessing field fertility conditions. (1986). https://doi.org/10.14288/1.0096890
29. Chig, L.A., Couto, E.G., Novaes Filho, J.P.: Distribuição espacial da granulometria, cor e carbono orgânico do solo ao longo de um transecto em microbacias na Amazônia meridional (2008). https://doi.org/10.1590/S0044-59672008000400015
30. Akbaş, F.: Spatial variability of soil color parameters and soil properties in an alluvial soil (2014). https://doi.org/10.5897/AJAR2013.8316
31. Fernandes, K., Marques Júnior, J., Ribon, A.A., et al.: Characterization and detailed mapping of C by spectral sensor for soils of the Western Plateau of São Paulo. Sci. Rep. (2024). https://doi.org/10.1038/s41598-024-66369-3

Comparative Evaluation of Filter and Wrapper Feature Selection Methods in Species Distribution Modeling

Yousra Cherif and Ali Idri(✉)

Software Project Management Research Team, ENSIAS, Mohammed V University in Rabat, Rabat, Morocco
ali.idri@um5.ac.ma

Abstract. Species Distribution Models (SDMs) have been extensively used as a tool for exploring the relationship between species occurrence and environmental factors. SDMs can be used for estimating the habitats where species can potentially exist and predicting their geographic distributions. In SDMs, feature selection (FS) is a crucial step in improving the accuracy of models and their interpretability by reducing the dimensionality of the feature space by discarding the least important features. In this study, SDMs were used to model the distributions of seven bird species, and a comparative analysis of FS techniques was conducted to identify the best approach. The FS methods used in this study include six different filter methods with two different thresholds each (40% and 50%). In addition, six wrapper techniques and two different multivariate methods have been used. The performance of the FS methods has been compared using four different classification methods: Extreme Gradient Boosting (XGB), Random Forest (RF), Decision Tree (DT), and Light Gradient Boosting Machine (LGBM). The results have shown that the Sequential Feature Selection (SFS) method has the highest performance. However, when the number of features is taken into consideration, it appears that Fisher Score with Random Forest is the best approach. Fisher Score with Random Forest has the highest performance with only nine features.

Keywords: Species distribution models · Feature selection · Environmental data · Machine learning · Classification

1 Introduction

Given the importance of SDMs in conservation practice, maintaining their accuracy and reliability is crucial. Feature Selection (FS) is commonly applied to improve model performance by selecting the most relevant predictors while removing redundant or irrelevant variables [6–8,13]. In addition to enhancing predictive accuracy [3], FS helps reduce data dimensionality, lower computational complexity, and increase model interpretability [14,19].

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 628–643, 2026.
https://doi.org/10.1007/978-3-032-30491-9_39

FS techniques are generally grouped into three main categories: filters, wrappers, and embedded approaches [15]. Filter methods rely on statistical relationships between features and the target variable, wrapper methods evaluate subsets of features using a predictive model, and embedded methods integrate selection directly into the training process. More recently, hybrid strategies have been proposed to combine the strengths of these approaches while reducing their individual limitations [24]. Filter techniques can further be divided into univariate methods, which rank features individually, and multivariate methods, which consider interactions among features.

For instance, the study [25] compared three filter methods with two embedded approaches using several selection thresholds on seven UCI Machine Learning Repository and FS datasets [17,18]. Their findings showed modest performance gains when ensemble rank aggregation was applied. Similarly, the study [9] evaluated multiple filters, wrappers, embedded, and ensemble strategies on environmental datasets using Random Forest and LightGBM classifiers. Shapley Additive Explanations and Permutation Importance were identified as strong individual methods, while Reciprocal Ranking performed best among ensemble approaches. Other works, such as the study [5], further emphasize the advantages of integrating filter and wrapper techniques, especially in medical applications. Nevertheless, significant gaps persist. Many studies depend on generic benchmark datasets instead of real-world SDM scenarios, which reduces their ecological relevance. Moreover, the interaction between FS methods and different classifiers, especially tree-based models commonly used in SDMs, is still underexplored. There is also a lack of systematic comparisons between univariate, multivariate, and wrapper approaches. These limitations motivate the present study, which aims to provide a more comprehensive and ecologically grounded evaluation of feature selection methods.

To the best of our knowledge, this paper is the first to concentrate on comparing univariate and multivariate filters, along with wrapper techniques. Furthermore, this study utilizes the Scott Knott (SK) statistical analysis, which has shown superior effectiveness compared to other statistical methods [2,6,8,16,29,30], such as those conducted by Calinski and Corsten [4] and Cox and Spjotvoll [28].

In this current paper, we have expanded our analysis by using: (1) six univariate filters (Linear Correlation, Mutual Information, Fisher Score, ANOVA F-value, minimum Redundancy Maximum Relevance, and t-test) with 40% and 50% thresholds as well as (2) two multivariate filters (Consistency-based subset selection and Correlation-based subset selection) and (3) six wrapper methods (Boruta, Permutation Importance, Shapley Additive Explanations, Genetic Algorithm, Recursive Feature Elimination with Cross-Validation, and Sequential Feature Selector), (4) seven bird species datasets (*P.moussieri, P.phoenicurus, P.ochruros, O.leucopyga, O.deserti, O.oenanthe, and O.leucura*) using the SK clustering test and Borda Count (BC) method to assess and contrast how FS affects classifier performance. In total, the current study evaluates 588 variants of

test set-ups: (4 classifiers) * (6 univariate filters * 2 thresholds + 2 multivariate filters + 6 wrapper methods + original feature set) * (7 datasets).

The paper is structured as follows: Sect. 2 describes the study area, detailing the species occurrence data, environmental variables, and the methodological steps followed to conduct the empirical analyses. Section 3 presents and discusses the main findings. Section 4 addresses the potential threats to validity. Finally, Sect. 5 concludes the paper by summarizing the key insights and contributions of the study.

2 Material and Methods

This study was conducted in Morocco using occurrence data from bird species belonging to the genera *Phoenicurus* and *Oenanthe*. Presence-only records were collected and complemented with pseudo-absence data to enable SDM. These datasets were then used to examine spatial patterns and build predictive models.

2.1 Study Area

Morocco, located in northwestern Africa, covers approximately 710,850 km^2. It borders Algeria and Mauritania and is bounded by the Mediterranean Sea to the north and the Atlantic Ocean to the west. The country presents substantial geographical diversity, including coastal areas, the Rif, Middle Atlas, High Atlas, and Anti-Atlas mountain ranges, as well as Saharan regions.

Morocco's climate is influenced by both maritime and desert conditions, resulting in marked spatial and seasonal variability. The average annual precipitation is approximately 318.8 mm, mainly concentrated between October and April, while summers are generally dry. Mean temperatures range from 9.4°C to 26°C.

2.2 Species Occurrence Datasets

The dataset includes occurrence records from three species of the genus *Phoenicurus* and four species of the genus *Oenanthe*. The *Phoenicurus* genus belongs to the Muscicapidae family (Redstarts), and the study focused on *Phoenicurus moussieri*, *Phoenicurus ochruros*, and *Phoenicurus phoenicurus*.

Since true absence data are difficult to obtain, pseudo-absence data were generated following the recommendations of [21]. Such pseudo-absences inherently involve uncertainty; therefore, absence points should not be sampled in ecologically suitable areas or too close to presence records [20]. In this study, pseudo-absence locations were randomly selected within a 250 km radius around presence points, as described in [21] and supported by previous studies such as [27], and applied in works including [1,26]. Buffer zones were further implemented around presence records to reduce environmental similarity and spatial bias. The presence-only dataset for *Phoenicurus* species contains 10,993 valid records [12], obtained after removing four records with missing values from the initial 10,997

entries. The wheatear dataset includes 1,750 records [11], all retrieved from the GBIF global database. Data preprocessing involved removing duplicates, handling missing values, and eliminating outliers to prevent degradation of model performance [22]. The wheatear records were primarily sourced from four repositories, with the EOD – eBird Observation Dataset contributing 91.2% of the data (1,597 records from ebird.org) and the Western Palearctic migratory bird dataset providing 4.4% (78 records from https://macroecology.ku.dk/resources/african_migrants). The observations span from 1965 to 2005.

All species names were validated according to ITIS standards and are associated with permanent Taxonomic Serial Numbers (TSNs) within the GBIF database (https://www.gbif.org/), as shown in Table 1. Ecologically, *Phoenicurus moussieri* is typically found in mountainous and rocky habitats with sparse vegetation. *Phoenicurus ochruros* prefers rocky natural and urban environments for nesting, while *Phoenicurus phoenicurus* is commonly associated with deciduous and mixed forests, forest edges, and open woodland areas. Table 2 summarizes the occurrence records.

Table 1. Description of the four wheatear species in Morocco

Sample Image	Scientific Name	Observations	TSN
	Oenanthe Deserti	400	561647
	Oenanthe Leucopyga	473	561652
	Oenanthe Leucura	469	561653
	Oenanthe Oenanthe	408	179814

Table 2. Description of the three redstart birds in Morocco

Sample Image	Scientific Name	Common Name	Total observations
	Phoenicurus Moussieri	Moussier's Redstart	5223
	Phoenicurus Ochruros	Black Redstart	3364
	Phoenicurus Phoenicurus	Common Redstart	2406

2.3 Environmental Data

Species distributions are strongly shaped by geographic and climatic conditions. Given that the persistence of the seven studied bird species is closely linked to environmental factors, climatic variables were selected as the primary predictors

in the distribution models. These variables enable a detailed representation of environmental conditions and support more reliable habitat suitability predictions.

A total of 19 bioclimatic variables were obtained from the WorldClim global database [10]. These variables describe average climatic conditions for the period 1970–2000. A spatial resolution of 2.5 arc-minutes (approximately 5 km) was adopted, as it provides a suitable compromise between ecological relevance and computational efficiency at the regional scale. This resolution is also consistent with the spatial uncertainty commonly associated with species occurrence records, particularly those derived from observational sources such as GBIF. Using a finer resolution could introduce an artificial level of precision not supported by the occurrence data. In addition, elevation data from the SRTM digital elevation database were incorporated [23]. In total, 20 environmental predictors were used to model the distribution of the seven bird species.

For the four wheatear species, land cover information was also included. Data were derived from NASA's MODIS Land Cover Type Product (MCD12Q1 v6), specifically the LC_type4 (LC4) classification scheme. This dataset categorizes land cover into classes such as broadleaf evergreen forests, non-vegetated areas, and urban zones. Integrating land cover variables allowed the models to better account for habitat characteristics that may influence species distribution patterns.

2.4 Experimental Process

This section outlines the experimental protocol adopted to evaluate the proposed methodology.

To investigate the effectiveness of the FS techniques, we applied a combination of univariate, multivariate, and wrapper approaches. These methods generated multiple reduced feature subsets based on different selection criteria. The impact of each subset was then assessed using four classification algorithms. All resulting variants, including those obtained from FS techniques and the complete feature set, were systematically evaluated across the seven datasets. The evaluation procedure followed an eight-step pipeline, summarized below.

Step 1: For each dataset, feature subsets were generated using six univariate ranking methods with two selection thresholds (40% and 50%), two multivariate methods, and six wrapper approaches. This resulted in 20 variants per dataset, yielding a total of 140 dataset variants (7 datasets $\times$ 20 variants).

Step 2: Four classifiers: DT, LGBM, XGB, and RF were trained on each feature subset generated in Step 1, as well as on the original feature set. A 5-fold cross-validation strategy was employed to compute accuracy, kappa, and f1-score. Overall, 147 model variants were produced per classifier across all datasets ((6 univariate filters $\times$ 2 thresholds + 2 multivariate filters + 6 wrappers + 1 original set) $\times$ 7 datasets).

Step 3: The performance of filter and wrapper methods was compared using the SK test and the BC voting approach to determine statistically significant differences and overall rankings among methods.

3 Results

This section reports and analyzes the performance of the FS techniques in combination with the four classifiers across the seven datasets. To structure the comparison, we applied the SK test based on accuracy and the BC voting method, which integrates Accuracy, Kappa, and F1-score, to rank classifiers within the top SK cluster.

Accordingly, the evaluation focuses exclusively on these three metrics, selected to provide a comprehensive and balanced assessment of model performance in our study context. In particular, we compare and contrast the effectiveness of filter and wrapper techniques using Accuracy, Kappa, and F1-score (RQ1).

– **Evaluation of filters (univariate and multivariate) and wrappers (RQ1)**

In this subsection, we compared the performance of all previously described univariate filters (using 40% and 50% thresholds), multivariate filters, and wrapper methods. Each selected feature subset, as well as the full feature set, was used to train four classifiers. Classifier performance was evaluated using the SK test based on accuracy, and classifiers within the top SK cluster were ranked using the BC voting method.

Figures 2, 3, 4, 5, 6 and 7 and 8 in the Appendix show the SK test findings for univariate and multivariate filters, while Table 3 summarizes the proportion of wrapper and filter methods in the top SK cluster for each dataset.

Table 3. Proportion of wrapper and filter methods in the top SK cluster for each dataset

Dataset	Wrapper (%)	Filter (%)	Interpretation
P.moussieri	54.17 (13/24)	36.67 (22/60)	Wrapper methods perform better than filters
P.ochruros	33.33 (8/24)	36.67 (22/60)	Wrapper and filter methods perform similarly
P.phoenicurus	50 (12/24)	25 (15/60)	Wrapper methods are preferable
O.deserti	75 (18/24)	75 (45/60)	Wrapper and filter methods perform equally
O.leucopyga	75 (18/24)	68.33 (41/60)	Wrapper methods slightly outperform filters
O.leucura	75 (18/24)	33.33 (20/60)	Wrapper methods are preferable
O.oenanthe	58.33 (14/24)	31.67 (19/60)	Wrapper methods perform better

To evaluate how filter and wrapper methods perform overall, we counted how often each feature selection technique appeared in the best SK clusters and then applied the BC using the top 10 ranks across the seven datasets (Table 5). Filter methods can appear up to 56 times per dataset [(6 univariate × 2 thresholds + 2 multivariate) × 4 classifiers] [(6 univariate ×2 thresholds + 2 multivariate)×4 classifiers], and wrapper methods up to 24 times (6 wrappers × 4 classifiers) (6

wrappers × 4 classifiers), resulting in maximum totals of 392 and 168 occurrences, respectively, across all datasets. As shown in Table 4, Boruta obtained the highest overall ranking, followed by the remaining techniques in the order reported in the table. Thus, when the feature-set size is not included in the evaluation criteria, wrapper techniques emerge as the strongest performers.

Table 4. Occurrence rates and BC rankings of the FS methods appearing in the top clusters over the seven datasets

FS technique	Occurrence rates	BC Ranking
Boruta	19 times out of 28 (67.86%)	3rd, 5th, 8th Ranks: 1x 6th Rank: 2x
CON	18 times out of 28 (64.29%)	5th Rank: 4x 6th Rank: 1x
GA	17 times out of 28 (60.71%)	1st, 8th Ranks: 2x 4th, 7th, 9th Ranks: 1x
RFE	17 times out of 28 (60.71%)	2nd, 4th, 10th Ranks: 1x
SFS	16 times out of 28 (57.14%)	1st Rank: 4x 2nd Rank: 2x 6th, 9th Ranks: 1x
PI	16 times out of 28 (57.14%)	1st, 2nd, 5th, 7th Ranks: 1x 3rd, 4th Ranks: 2x
Shap	16 times out of 28 (57.14%)	4th, 5th, 9th, 10th Ranks: 1x 7th Rank: 2x
F	26 times out of 56 (46.43%)	2nd, 6th, 7th, 8th, 9th Ranks: 1x
M	25 times out of 56 (44.64%)	8th, 9th Ranks: 1x 10 Ranks: 2x
CFS	12 times out of 28 (42.86%)	3rd, 6th Rank: 1x
MR	22 times out of 56 (39.29%)	9th Rank: 1x
TT	21 times out of 56 (37.5%)	2nd, 3rd, 10th Ranks: 1x
A	21 times out of 56 (37.5%)	3rd, 7th, 9th, 10th Ranks: 1x
C	21 times out of 56 (37.5%)	8th Rank: 1x

Moreover, to rank the FS techniques, we applied the BC to identify the top 10 methods within the best SK cluster. The best classifierâĂŞfilter (or wrapper) combination is the one that attains the top BC ranking while selecting the smallest number of features.

Table 5 reports the top-ten ranked techniques and their corresponding feature counts; for example, RFORG(20) indicates that RFORG was trained with 20 features, while RFSFS(10) used 10 features. Since all techniques in the best SK cluster show equivalent accuracy, those highlighted in bold represent the most efficient options (highest Borda rank with fewer features).

Overall, these findings highlight the importance of feature selection in improving classifier performance. The full feature set ranked second or third only once and appeared in lower ranks (fourth, sixth, and seventh) when RF was trained with 20âĂŞ21 features, whereas both filters and wrappers achieved comparable performance with far fewer features. When considering feature dimensionality, SFS and Fisher Score showed a clear advantage, consistently delivering strong performance with reduced feature sets.

Table 5. BC top ten rankings of filter and wrapper approaches, including the corresponding feature set size in parentheses

P.mous	P.och	P.pho	O.deserti	O.leucopyga	O.leucura	O.oenanthe
RFSFS(10)	**RFSFS(10)**	**RFSFS(10)**	**RFPI(10)**	RFGA(15)	**RFSFS(10)**	RFGA(13)
RFPI(10)	RFF50(10)	RFRFE(17)	RFSFS(10)	RFTT40(8)	RFORG(21)	**RFSFS(10)**
RFORG(20)	RFCFS(9)	RFPI(10)	**XGBTT50(11)**	RFA50(11)	RFPI(10)	RFBoruta(21)
RFShap(10)	RFORG(20)	RFGA(14)	RFORG(21)	**XGBPI(10)**	RFRFE(21)	RFPI(10)
RFCON(13)	RFShap(10)	RFBoruta(19)	RFCON(16)	RFPI(10)	RFCON(13)	RFCON(18)
XGBSFS(10)	**RFF40(8)**	RFCON(17)	RFBoruta(20)	**RFCFS(6)**	RFBoruta(21)	RFORG(21)
RFA40(8)	RFPI(10)	RFORG(20)	RFShap(10)	RFShap(10)	RFGA(10)	**RFF40(8)**
RFGA(7)	RFM50(10)	XGBORG(20)	RFC50(11)	RFBoruta(20)	**RFF50(11)**	XGBGA(13)
RFM40(8)	RFMR50(10)	XGBGA(14)	RFA50(11)	RFSFS(10)	XGBShap(10)	RFF50(11)
RFM50(10)	RFM40(8)	**RFM40(8)**	RFTT50(11)	RFRFE(11)	RFShap(10)	RFA50(11)

In this subsection, we computed the median accuracy, Kappa, and F1-score across all datasets to determine the most effective FS techniques. For **RF**, the top accuracy was reached using RFSFS (91.79%); for **XGB**, XGBShap (91.54%) and XGBSFS (91.52%); for **LGBM**, LGBMSFS (91.28%); and for **DT**, DTORG and DTBoruta (89.43%) (Table 6). Figure 1 shows classifier performance grouped into 65 SK clusters: techniques sharing a color are statistically indistinguishable, while different colors indicate significant differences. The top clusters for RF contained RFSFS, RFORG/RFPI, and RFRFE, trained with 10, 21, 10, and 14 features, respectively, indicating that RF performs best with SFS. Overall classifier ranking is RF > XGB > LGBM > DT [8]. Considering feature count, XGBShap, LGBMSFS, and DTBoruta were the best combinations.

Without accounting for feature numbers, Boruta ranked highest, followed by Consistency-based subset selection, Genetic Algorithm, RFE with CV, SFS, Permutation Importance, Shap, Fisher Score, Mutual Information, Correlation-based subset selection, mRMR, t-test, ANOVA F-value, and Linear Correlation. When feature count is considered, RF combined with SFS is preferred, as it uses only 10 features and appears in the top SK cluster (see Fig. 1).

Table 6. Median results for the three metrics, computed for the filters over all seven datasets

Classifier	FS technique	Accuracy(%)	F1_score(%)	Kappa(%)
RF	M40	91.53	91.53	83.06
	M50	91.3	91.3	82.6
	C40	91.25	91.25	82.5
	C50	91.36	91.36	82.72
	A40	91.41	91.41	82.82
	A50	91.38	91.38	82.77
	F40	91.36	91.36	82.72
	F50	91.58	91.58	83.16
	TT40	90.9	90.9	81.82
	TT50	91.25	91.25	82.51
	MR40	91.16	91.16	82.33
	MR50	91.32	91.32	82.64
	ORG	91.75	91.75	83.51
	CON	91.63	91.63	83.26
	CFS	91.41	91.41	82.82
	RFE	91.7	91.7	83.41
	PI	91.75	91.75	83.5
	Boruta	91.63	91.63	83.26
	Shap	91.53	91.53	83.06
	SFS	**91.79**	**91.79**	**83.58**
XGB	M40	90.84	90.84	81.69
	M50	90.83	90.83	81.66
	C40	90.49	90.49	80.97
	C50	90.55	90.55	81.1
	A40	90.49	90.49	80.97
	A50	90.55	90.55	81.1
	F40	91.1	91.1	82.21
	F50	91.25	91.25	82.5
	TT40	90.74	90.74	81.48
	TT50	90.88	90.88	81.76
	MR40	90.4	90.4	80.8
	MR50	90.94	90.94	81.88
	CON	91.36	91.36	82.72
	CFS	90.9	90.9	81.81
	ORG	91.26	91.26	82.52
	Boruta	91.15	91.15	82.3
	SFS	**91.52**	**91.52**	**83.04**
	PI	91.33	91.33	82.67
	Shap	**91.54**	**91.54**	**83.09**
	GA	91.01	91.01	82.01
	RFE	91.15	91.15	82.3

(*continued*)

Table 6. (*continued*)

Classifier	FS technique	Accuracy(%)	F1_score(%)	Kappa(%)
LGBM	M40	90.45	90.45	80.9
	M50	90.71	90.71	81.42
	C40	89.93	89.93	79.86
	C50	90.2	90.2	80.41
	A40	89.93	89.93	79.86
	A50	90.2	90.2	80.41
	F40	90.31	90.31	80.64
	F50	90.48	90.48	80.97
	TT40	89.67	89.67	79.34
	TT50	90.41	90.41	80.83
	MR40	89.84	89.84	79.67
	MR50	90.3	90.3	80.6
	ORG	91.13	91.13	82.25
	CON	91.09	91.09	82.18
	CFS	90.6	90.6	81.22
	RFE	91.02	91.02	82.03
	PI	91.03	91.03	82.06
	Boruta	91.02	91.02	82.03
	Shap	91.13	91.13	82.25
	GA	91.01	91.01	82.03
	SFS	**91.28**	**91.28**	**82.56**
DT	M40	89.06	89.06	78.12
	M50	88.94	88.94	77.88
	C40	88.88	88.88	77.75
	C50	88.94	88.94	77.87
	A40	88.85	88.85	77.69
	A50	89.05	89.05	78.1
	F40	88.94	88.94	77.87
	F50	88.81	88.81	77.61
	TT40	87.93	87.93	75.86
	TT50	88.61	88.61	77.21
	MR40	88.69	88.69	77.38
	MR50	88.62	88.62	77.24
	ORG	**89.43**	**89.43**	**78.86**
	CON	88.84	88.84	77.68
	CFS	88.83	88.83	77.65
	RFE	89.21	89.21	78.42
	PI	89.16	89.16	78.32
	Boruta	**89.43**	**89.43**	**78.86**
	Shap	89.12	89.12	78.25
	GA	89.09	89.09	78.18
	SFS	89.25	89.25	78.49

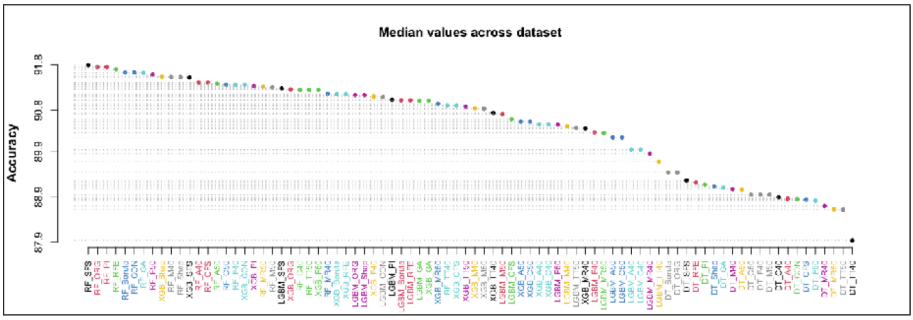


Fig. 1. SK plot of filter (univariate and multivariate) and wrapper techniques over the seven datasets.

4 Threats to Validity

This section summarizes the main validity concerns of the study.

Internal Validity: To reduce overfitting, 5-fold cross-validation was employed. Nevertheless, further hyperparameter tuning (e.g., pruning) could improve model generalization. Although the implementation was carefully verified, minor errors may still exist. The results could also vary if other classifiers (e.g., KNN, Logistic Regression, or Naïve Bayes) were used. In addition, computational cost was not assessed, even though univariate filters are generally less expensive than multivariate and ensemble approaches.

External Validity: The experiments were conducted on seven datasets from the *Phoenicurus* and *Oenanthe* groups. Consequently, findings may differ with other datasets, classifiers or ranking strategies. Replication using additional feature selection techniques and more diverse datasets would enhance generalizability.

Construct Validity: Model performance was evaluated using three widely adopted metrics: accuracy, F1-score, and Cohen's Kappa. Their results were aggregated using the SK test and the BC method with equal weights, ensuring a balanced and robust evaluation.

5 Conclusion

This study empirically compared 588 configurations of four classifiers (LGBM, RF, DT, and XGB) across seven bird species datasets. The comparison included: (1) six univariate filters with two selection thresholds (40% and 50%) plus the original feature set, (2) two multivariate techniques, and (3) six wrapper methods. Performance was assessed using accuracy, Cohen's Kappa, and F1-score, combined through the SK test and BC ranking. The main findings are summarized as follows:

(RQ1): SFS achieved the highest overall performance among feature selection methods. However, considering feature reduction efficiency, the Fisher Score combined with RF was the most economical approach, selecting an average of nine features.

Acknowledgments. This research was supported by the National Center for Scientific and Technical Research (CNRST) of Morocco under the "PhD-Associate Scholarship - PASS" program.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

Appendix

SK test results over the seven datasets for the filter and wrappers FS techniques

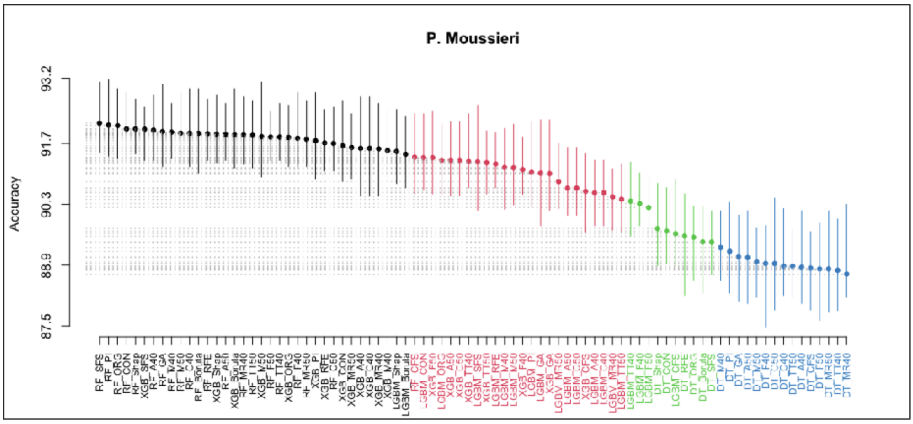


Fig. 2. SK test plot of the filter and wrapper methods across P.moussieri dataset.

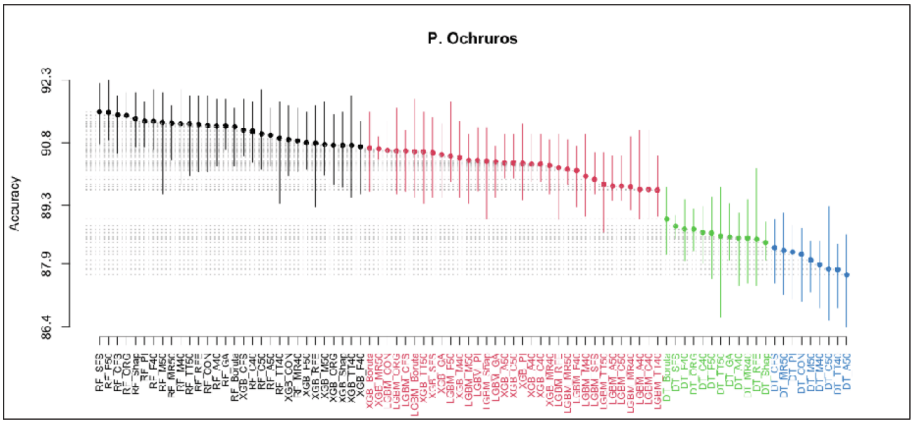


Fig. 3. SK test plot of the filter and wrapper methods across P.ochruros dataset.

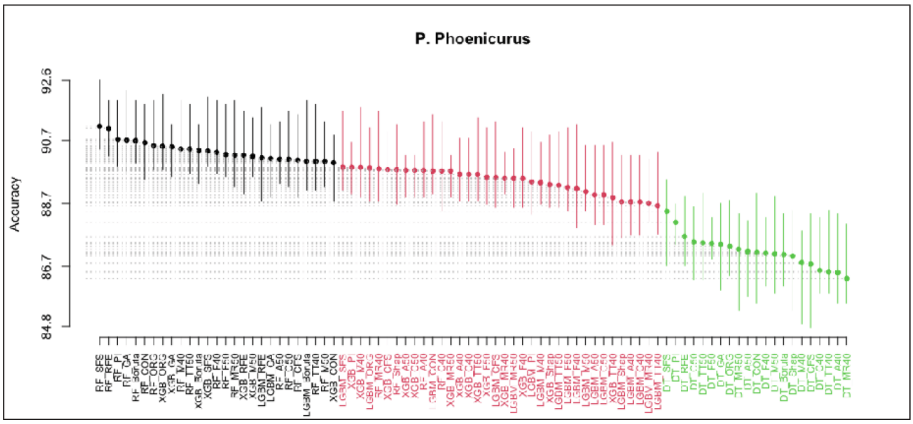


Fig. 4. SK test plot of the filter and wrapper methods across P.phoenicurus dataset.

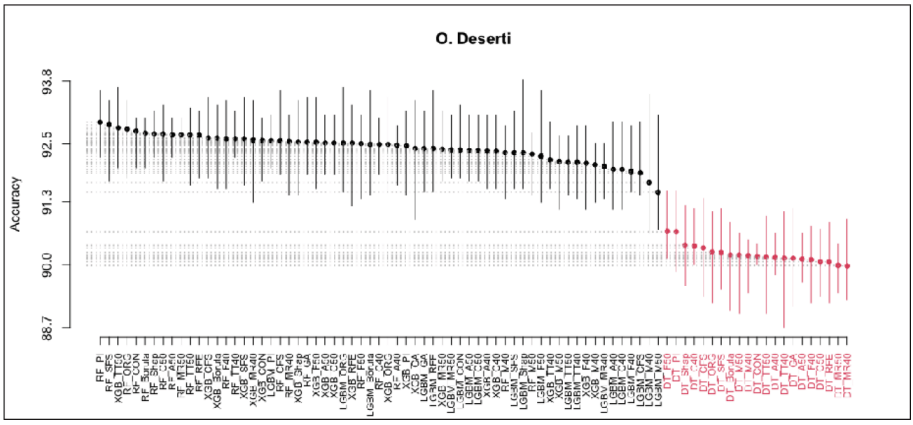


Fig. 5. SK test plot of the filter and wrapper methods across O.deserti dataset.

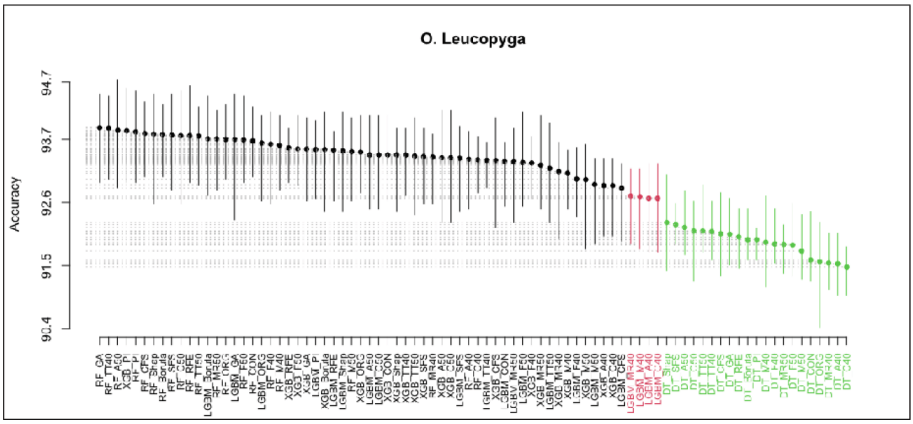


Fig. 6. SK test plot of the filter and wrapper methods across O.leucopyga dataset.

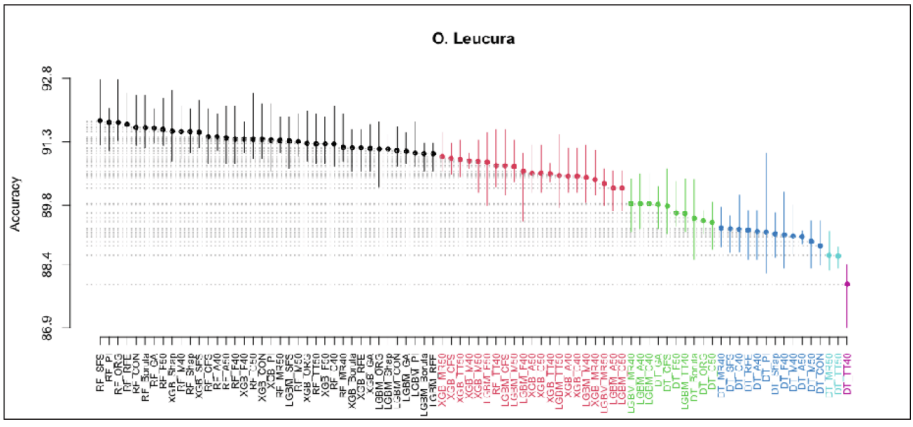


Fig. 7. SK test plot of the filter and wrapper methods across O.leucura dataset.

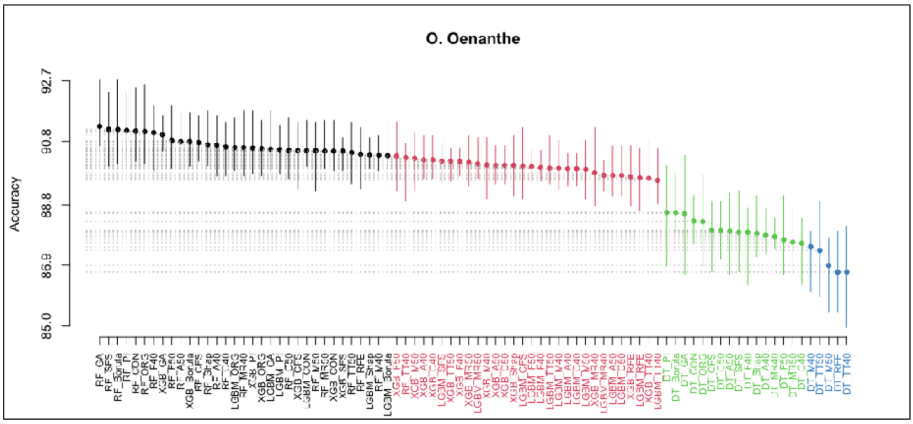


Fig. 8. SK test plot of the filter and wrapper methods across O.oenanthe dataset.

References

1. Barbet-Massin, M., Jiguet, F., Albert, C.H., Thuiller, W.: Selecting pseudo-absences for species distribution models: how, where and how many? Methods Ecol. Evol. **3**(2), 327–338 (2012)
2. Benhar, H., Idri, A., Hosni, M.: Impact of threshold values for filter-based univariate feature selection in heart disease classification. In: HEALTHINF, pp, pp. 391–398 (2020)
3. Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A.: Recent advances and emerging challenges of feature selection in the context of big data. Knowl.-Based Syst. **86**, 33–45 (2015). https://doi.org/10.1016/j.knosys.2015.05.014, https://www.sciencedirect.com/science/article/pii/S0950705115002002
4. Calinski, T., Corsten, L.: Clustering means in anova by simultaneous testing. Biometrics, 39–48 (1985)
5. Chen, C.W., Tsai, Y.H., Chang, F.R., Lin, W.C.: Ensemble feature selection in medical datasets: combining filter, wrapper, and embedded feature selection results. Expert. Syst. **37**(5), e12553 (2020)
6. CHERIF, Y., IDRI, A.: Evaluating the efficacy of wrappers in predicting species distribution. In: 2024 World Conference on Complex Systems (WCCS), pp. 1–6 (2024). https://doi.org/10.1109/WCCS62745.2024.10765519
7. Cherif, Y., Idri, A.: Single and ensemble based filters in environmental data. Expert. Syst. **42**(7), e70076 (2025)
8. Cherif., Y., Idri., A., El Alaoui., O.: Impact of thresholds of univariate filters for predicting species distribution. In: Proceedings of the 15th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - KDIR, pp. 86–97. INSTICC, SciTePress (2023). https://doi.org/10.5220/0012203000003598
9. Effrosynidis, D., Arampatzis, A.: An evaluation of feature selection methods for environmental data. Eco. Inform. **61**, 101224 (2021). https://doi.org/10.1016/j.ecoinf.2021.101224, https://www.sciencedirect.com/science/article/pii/S1574954121000157

10. Fick, S., Hijmans, E., R.: Worldclim 2: new 1-km spatial resolution climate surfaces for global land areas. Int. J. Climatol., 1–14 (2017)
11. GBIF, biodiversity, species occurrences (2023). https://doi.org/10.15468/dl.7uxqxk, https://www.gbif.org/occurrence/download/0239931-220831081235567
12. GBIF, biodiversity, species occurrences (2023). https://doi.org/10.15468/dl.vuzsmjhttps://www.gbif.org/occurrence/download/0003099-230530130749713
13. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.A.: Feature extraction: foundations and applications, vol. 207. Springer (2008). https://doi.org/10.1007/978-3-540-35488-8
14. Jaganathan, P., Kuppuchamy, R.: A threshold fuzzy entropy based feature selection for medical database classification. Comput. Biol. Med. **43**(12), 2222–2229 (2013)
15. Jovic, A., Brkić, K., Bogunovic, N.: A review of feature selection methods with applications. pp. 1200–1205 (May 2015). https://doi.org/10.1109/MIPRO.2015.7160458
16. Lahmar, C., Idri, A.: On the value of deep learning for diagnosing diabetic retinopathy. Heal. Technol., 1–17 (2022)
17. Li, J., et al.: Feature selection: a data perspective. ACM Comput. Surv. (CSUR) **50**(6), 1–45 (2017)
18. Lichman, M.: Uci machine learning repository, p. 2013 (2013)
19. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Trans. Knowl. Data Eng. **17**(4), 491–502 (2005)
20. Lobo, J.M., Jiménez-Valverde, A., Hortal, J.: The uncertain nature of absences and their importance in species distribution modelling. Ecography **33**(1), 103–114 (2010)
21. MacKenzie, D.I., Royle, J.A.: Designing occupancy studies: general advice and allocating survey effort. J. Appl. Ecol. **42**(6), 1105–1114 (2005)
22. Nyitrai, T., Virág, M.: The effects of handling outliers on the performance of bankruptcy prediction models. Socioecon. Plann. Sci. **67**, 34–42 (2019)
23. Orlandi, A.G., Carvalho, O.A., Guimarães, R.F., Bias, E.D.S., Corrêa, D.C., Gomes, R.A.T.: Vertical accuracy assessment of the processed srtm data for the brazilian territory. Boletim de Ciências Geodésicas **25** (2019)
24. Peter, T.J., Somasundaram, K.: Study and development of novel feature selection framework for heart disease prediction. Int. J. Sci. Res. Publ. **2**(10), 1–7 (2012)
25. Seijo-Pardo, B., Porto-Díaz, I., Bolón-Canedo, V., Alonso-Betanzos, A.: Ensemble feature selection: homogeneous and heterogeneous approaches. Knowl.-Based Syst. **118**, 124–139 (2017)
26. Senay, S.D., Worner, S.P., Ikeda, T.: Novel three-step pseudo-absence selection technique for improved species distribution modelling. PLoS ONE **8**(8), e71218 (2013)
27. VanDerWal, J., Shoo, L.P., Graham, C., Williams, S.E.: Selecting pseudo-absence data for presence-only distribution modeling: how far should you stray from what you know? Ecol. Model. **220**(4), 589–594 (2009)
28. Worsley, K.J.: Confidence regions and tests for a change-point in a sequence of exponential family random variables. Biometrika **73**(1), 91–104 (1986)
29. Zouhri, H., Idri, A.: A comparative assessment of wrappers and filters for detecting cyber intrusions. In: World Conference on Information Systems and Technologies, pp. 118–127. Springer (2024). https://doi.org/10.1007/978-3-031-60221-4_12
30. Zouhri, H., Idri, A., Ratnani, A.: Evaluating the impact of filter-based feature selection in intrusion detection systems. Int. J. Inf. Secur. **23**(2), 759–785 (2024)

Provision of Financial Security of the State and Ways of Threat Prevention

Nadiia Davydenko[1], George Abuselidze[2(✉)], Olena Gudz[1], and Svitlana Onyshko[1]

[1] State Tax University, Universytetska, 31, Irpin 08205, Ukraine
[2] Batumi Shota Rustaveli State University, 35/32, Ninoshvili/Rustaveli Street, Batumi 6010, Georgia
george.abuselidze@gmail.com

Abstract. The article aims to define the essence of the financial security of the state, substantiate current threats to the financial security and study the real level of financial security. It is established that the financial security of the state is determined by the following factors: protection of financial interests at all levels of financial relations; sufficient level of financial autonomy; stability and steadiness of the state's financial system under the influence of external and internal destabilizing factors that pose a threat to national interests in the financial sphere; the ability of the state's financial system to ensure efficient functioning of the national economic system and sustainable economic growth. To prevent threats to the financial security of the state under martial law, it is advisable to minimize the risks associated with the implementation of politically motivated decisions in the field of state financial policy; the risks of inflationary processes associated with the real loss of the national currency value; the risks associated with an increase in public debt and a reduction in available sources of financing; financial risks arising from the need to purchase weapons; the risks of inefficient and irrational insurance, banking and tax systems of the state; the risks associated with compensation payments related to military operations.

Keywords: financial security · threats · risks · public debt · debt burden · budget security · prevention

1 Introduction

The turn of the second and third decades of the twenty-first century revealed that an objective threat to sustainable development is not only the cyclical nature of the economy, but also the intensification of the already common economic crises by non-economic shocks that pose not so much financial risks as danger to human life and health and their property (wars, pandemics, natural and man-made disasters). These and similar global shocks have been called «black swans» in professional circles because of the impossibility to predict their occurrence and, consequently, the difficulty to assess the consequences for economic development at the stage of the relevant events' unfolding. The uncertainty

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 644–659, 2026.
https://doi.org/10.1007/978-3-032-30491-9_40

of the direct impact and consequences of existential shocks in the national economy in general and the financial sector in particular complicates crisis response for both public authorities and ordinary business entities. Although traditional methods of financial security management are still characterized by signs of effectiveness, the challenges of non-economic shocks call into question the strategic effectiveness of traditional strategies. Therefore, the development of ways to ensure the financial security of the state should not only protect the country's economy from the consequences of existential crises but also contribute to a rational post-war recovery.

Ensuring the financial security of the state is a key factor for maintaining Ukraine's sovereignty, creating conditions for its progressive development and improving the living standards of the population [1]. In the global context, each state faces the challenge of ensuring its own financial security. Current trends of globalization and new security threats are leading to increased attention of the international community to the issues of forming and strengthening the financial security of countries.

The research on this issue is of particular relevance nowadays, since national security is a fundamental prerequisite for the full functioning and progress of individuals, society, and the state as a whole. In the current realities, the financial security sphere is characterized by a multiplicity of threats of various scales, many of which are global in nature, including armed conflicts and military confrontations.

The formation of an effective mechanism for ensuring the financial security of the state requires a thorough academic analysis of its economic nature, the development and improvement of assessment methodologies, as well as the rethinking of theoretical concepts to ensure the stability and resilience of the state financial system. Despite a considerable amount of research in this area, no consensus has been reached on a precise definition of the "financial security of the state" concept and a clear identification of the risks associated with it.

Moreover, in our opinion, there is a lack of academic research on assessing the financial security of the state under martial law. This gap in research requires special attention, given the relevance of this issue in the realities of martial law in the country.

The article [2] define the financial security of the state as the ability of government agencies to respond promptly and appropriately to internal and external threats that may negatively affect financial stability, both in peacetime and in emergencies.

According to Varnalii and Tomashevskyi, the financial security of the state is one of the main components of economic security, since the state of public finances is closely related to the general state of the national economy. They emphasize that the economic well-being of the state largely depends on the level of financial security of individual citizens, society, and the nation as a whole [3].

Ensuring the financial security of the state is one of the key tasks in formulating strategic goals for the development of the financial system, as noted by Shkolnyk [4]. The impact of public debt on the financial security of Ukraine was studied in the works [5–7] and an assessment of Ukraine's debt security is presented in the study [8, 9].

Korobtsova emphasizes that in the context of martial law declared by the Decree of the President of Ukraine of February 24, 2022, the issues of financial security of the state are of particular relevance. This is due to the fact that the state's defense as well as the functioning of the army and defense authorities depend on sufficient and stable financial

resources, which can be both domestic and attracted through international assistance. The availability of financial resources, their accumulation and stable use to achieve goals that contribute to the victory over the aggressor, the elimination of threats to the state integrity and the protection of national interests and security of citizens are components of the financial security of the state that require legal regulation [10].

Thus, it can be argued that in the difficult geopolitical situation that emerged in early 2022, Ukraine needs to strengthen its national security. An important component of this process is financial security, as it ensures the maintenance of the financial potential and sovereignty of the state, as well as creates conditions for improving its position in the international arena. The relevance of a detailed analysis of the strengthening process of the state's financial security is due to the presence of negative internal and external factors that serve as criteria for ensuring the proportional and balanced functioning of the state.

The main threats to the financial security of the state are the factors that are most sensitive to the destructive impact of the war, such as a decrease in labor potential, as well as physical or financial shortages of critical resources and logistics. There is no official data on the number of deaths in the war. Obviously, most of these losses are among people of working age. At the same time, there are huge demographic losses due to a decline in the birth rate, high migration, including women of reproductive age, and an aging population. Currently, Ukraine has all the elements that could lead to a demographic catastrophe and a 24–33% decline in population, depending on the duration of the war, namely additional deaths due to the war, increased emigration (refugees), lower birth rates, and the likely loss of territories with populations living there. This will lead to a significant shortage of labor and, as a result, a decrease in labor potential. In addition, there are important issues related to the ability to quickly recover and the possibility of promptly addressing the imbalances that have accumulated during the conflict. The uncertainty arising from a full-scale military invasion of Ukraine by Russia also poses a serious threat. This includes the impossibility of conducting economic activity in large areas, risks and uncertainty in the functioning of logistics routes, as well as different potential for diversification of production and economic processes due to the geographical conditions of warfare [10, 11].

We agree with the definition offered by Vlasiuk regarding the concept of threats to the financial security of Ukraine. According to his interpretation, these threats are a set of existing and potential factors and phenomena that have a destabilizing impact on the financial sector of the state. The manifestations of these threats can be diverse and include: a significant reduction in gross domestic product, disruption of the mechanisms of formation and distribution of centralized state funds, destabilization of the banking sector, devaluation of the national currency, and the possibility of default on government debt obligations. In addition, one of the significant consequences of these threats is the growing tendency to shadow the economy [12].

2 Methodology

To analyze the state of financial security and its components, as well as the dynamics and structure of public and publicly guaranteed debt, the method of comparing absolute and relative indicators was used, which were calculated based on empirical data provided by

the State Statistics Service of Ukraine, the National Bank of Ukraine and the Ministry of Finance of Ukraine. In calculating the level of financial security of Ukraine, official indicators and thresholds of indicators were used, as provided for in the Methodological Recommendation for Calculating the Level of Economic Security of Ukraine, approved by the Order of the Ministry of Development and Trade of Ukraine No. 1277 dated 29.10.2013 [13]. In the process of determining the possibilities of ensuring the financial security of the state and preventing potential threats, the methods of historical and logical analysis, as well as the method of expert evaluation, were applied.

Objectives. The main aim of the article is to define the essence of the financial security of the state, to substantiate the current threats to the financial security of Ukraine under martial law, and to study the real level of financial security in Ukraine.

3 Results and Discussion

The economic realities of the present indicate that further development, dynamics and main directions of socio-economic progress of domestic enterprises largely depend on the current state of economic security of the state, in which financial security occupies a key position in the overall system.

Analyzing the different points of view of scholars, it can be argued that the main challenges to financial security are threats to the smooth operation of all its components. At the same time, it should be taken into account that all financial security issues become particularly relevant during financial crises, both at the level of individual states and in the context of global changes, when there is a "shrinkage" of financial markets and increased competition for limited resources, which are available only to those countries whose solvency is beyond doubt.

An effective method of minimizing the impact of external threats is to form mutually beneficial partnerships with states that are sources of such threats. In other situations, in order to achieve the optimal level of financial security of the state, it is advisable to create reserve funds and purposefully apply fiscal, budgetary, monetary and other instruments of state regulation that are consistent with market conditions.

Financial security can be characterized as a complex multilevel system consisting of a number of subsystems, each of which has its own unique structure and conceptual content (Fig. 1).

Today, the condition of Ukraine's financial system is being shaped by a number of destructive internal challenges, the main ones include a decline in real gross domestic product, rising inflation, persistently high budget deficit and public debt, and worsening imbalances in the fiscal and monetary spheres.

The amount of public debt, as well as its allocation and repayment, has a direct or indirect impact on most economic processes in the state. This applies to economic growth, income distribution, budget deficit, money supply and changes in aggregate supply and demand, consumption and savings, current account balance, international solvency, and the state's credit rating. Thus, public debt acts not only as a tool for attracting financial resources to meet public needs but also as an important element of financial policy, the mismanagement of which can lead to serious complications or even a financial crisis. Here is an analysis of the level and dynamics of the state's guaranteed public debt (Fig. 2).

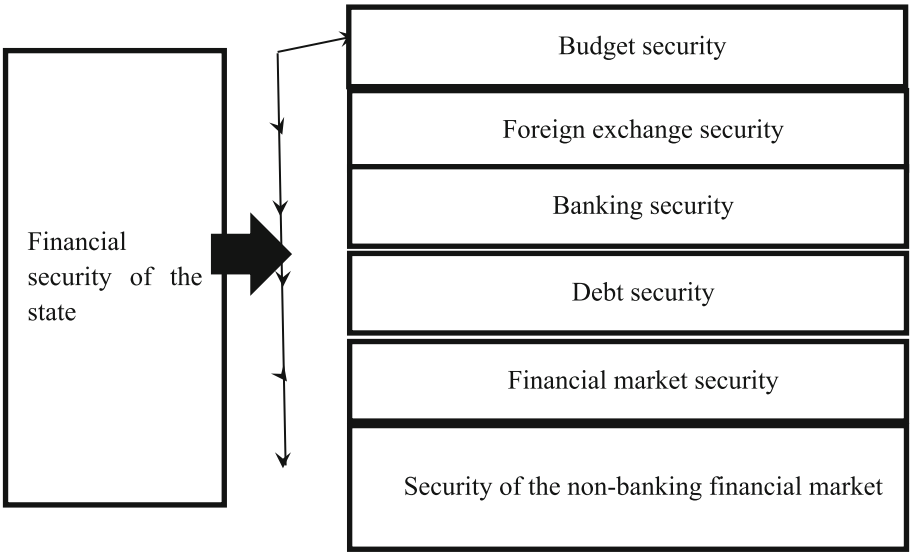


Fig. 1. Components of financial security of the state* *compiled by the authors on [1, 3, 4, 14]

An analysis of the data presented in Fig. 2 allows us to conclude that the total volume of public debt obligations in dollar terms increased relatively moderately in 2012–2022. There is an increase from 64.5 billion USD in 2012 to 111.3 billion USD in 2022, which is an increase of 73%. At the same time, the dynamics of external and internal debt show different growth rates. External debt in dollar terms increased by 85% during the said period (from 38.7 billion USD to 71.4 billion USD), while internal debt increased by 55% (from 25.8 billion USD to 39.9 billion USD).

It is worth noting that the absolute value of public debt is not a representative indicator of the state's real economic situation. This is due to the fact that debt growth can occur in parallel with GDP growth, and its value is also affected by inflationary processes. For a more objective assessment of the debt burden, a relative indicator is used - the ratio of public debt to GDP. According to the second part of Article 18 of the Budget Code of Ukraine, the limit value of this indicator is set: the total amount of public and publicly guaranteed debt at the end of the budget period should not exceed 60% of the annual nominal GDP of the state. The dynamics of Ukraine's GDP in 2012–2022 are shown in Table 1, and the dynamics of the relative debt burden in Ukraine are shown in Fig. 3.

An analysis of the data presented in Table 1 shows that the dynamics of Ukraine's nominal GDP in 2012–2022 were characterized by a predominantly upward trend in national currency, with the exception of 2022. The most significant growth was recorded in 2021, when the increase was 30.2% in hryvnia terms and 28.4% in dollar terms.

The decline in GDP in 2022 is mainly due to Russia's large-scale military aggression against Ukraine. However, it is worth noting that this is not the only factor that affected the economic downturn. Other factors include the growth of external debt, a significant shadow economy, and the lack of investment attractiveness of Ukrainian enterprises.

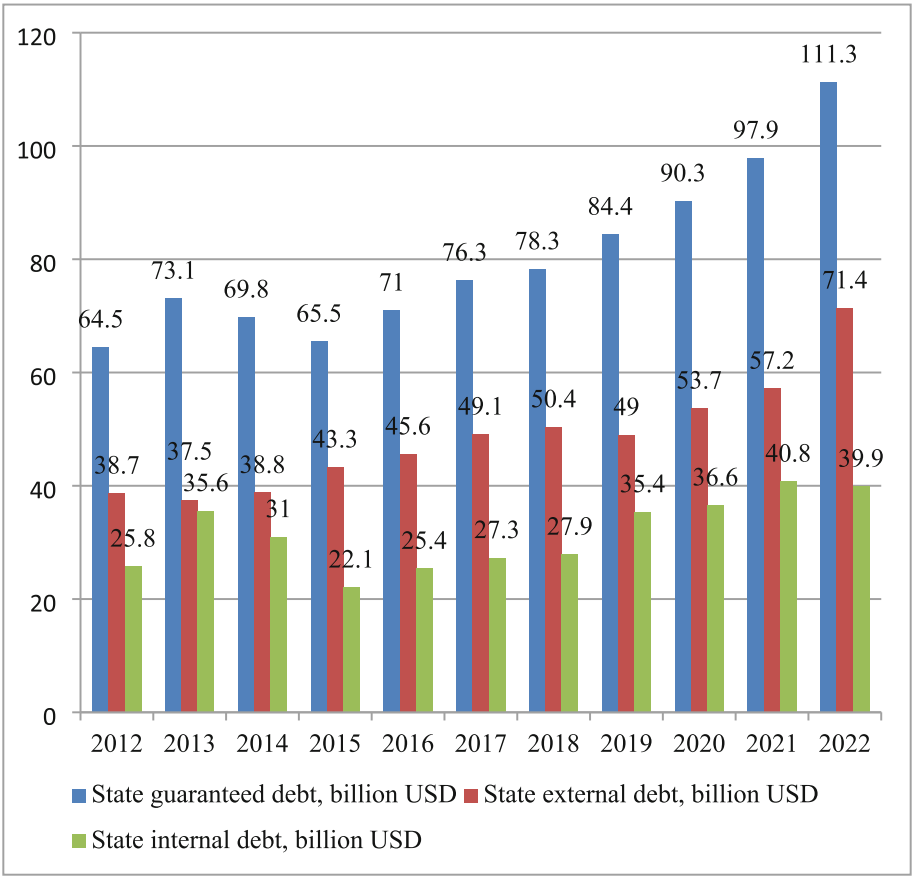


Fig. 2. Dynamics of the guaranteed public debt of Ukraine in 2012–2022 Source: compiled by the authors on [15, 16]

Despite the overall positive trend in the national currency, the dollar equivalent of nominal GDP experienced a sharp decline in 2015. This was caused by the rapid devaluation of the hryvnia: while in 2014 the exchange rate was 11.89 hryvnia to the dollar, in 2015 it almost doubled to 21.84 hryvnia to the dollar. This currency volatility had a significant impact on the dollar value of Ukraine's GDP.

The study of debt burden trends presented in Fig. 3 shows a significant deterioration in the state's debt security since 2014. Subsequent periods were characterized by a progressive weakening of the debt position, reaching its critical point in 2016 for the entire period under analysis from 2012 to 2022. The key factors that led to the instability of Ukraine's debt position are the systematic budget deficit and the irrational use of attracted resources. A statistical review of the debt burden evolution shows that 2022 was marked by a sharp increase in public debt. This has led to a number of negative economic consequences, including increased inflation, destabilization of the payment

Table 1. The dynamics of the nominal GDP of Ukraine in 2012–2022

Year	Nominal GDP					
	Million UAH	Growth, million UAH	Growth, %	Million USD	Growth, million USD	Growth, %
2012	1408889	+92289	+7.0	175781	+12622	+7.7
2013	1454931	+46042	+3.3	183310	+7529	+4.3
2014	1566728	+111797	+7.7	131805	−51505	−28.1
2015	1979458	+412730	+26.3	90615	−41190	−31.3
2016	2383182	+403724	+20.4	93270	+2655	+2.9
2017	2982920	+599738	+25.2	112154	+18884	+20.2
2018	3558706	+575786	+19.3	130832	+18678	+16.7
2019	3974564	+415858	+11.7	153781	+22949	+17.5
2020	4194102	+219538	+5.5	155582	+1801	+1.2
2021	5459574	+1265472	+30.2	199770	+44188	+28.4
2022	5191028	−268546	−4.9	160500	−39270	−19.7

Source: compiled by the authors on [16]

balance, reduction of production capacity, decrease in private investment, and depletion of the state's gold and foreign exchange reserves.

The following is an assessment of the state's financial security components (Table 2).

After analyzing the key components of the state's financial security, it can be concluded that its condition generally deteriorated in the period from 2012 to 2022. In particular, the level of budgetary security, as an important component of financial security, during 2012–2022 demonstrated a critical state in 2014–2016. In the period from 2017 to 2021, an unsatisfactory level was observed; however, this had a slight but positive impact on the overall state of the national financial system. In 2022, the budget security indicator decreased to 0.34, so it was not possible to maintain the positive trends, even though they were insignificant.

In our opinion, currency security is one of the key elements of the integrated assessment for Ukraine, as the country is import-dependent. The study for 2012–2022 did not reveal any clear trends. In 2012–2014, there was a sharp decline and the indicator was within critical limits. From 2015–2021, there was a gradual increase in the value of the indicator and a shift to the danger zone. In 2022, there was a slowdown in the growth rate and an approach to the critical level, which may mean that this indicator did not become the basis for stabilizing the level of financial security.

An analysis of trends in domestic debt security reveals the most problematic aspects. In particular, in 2015, there was a critically low level of this indicator, which amounted to only 0.024. The period from 2012 to 2015 was characterized by a steady decline in debt security. However, from 2016 to 2018, there was a gradual increase in the value of this indicator.

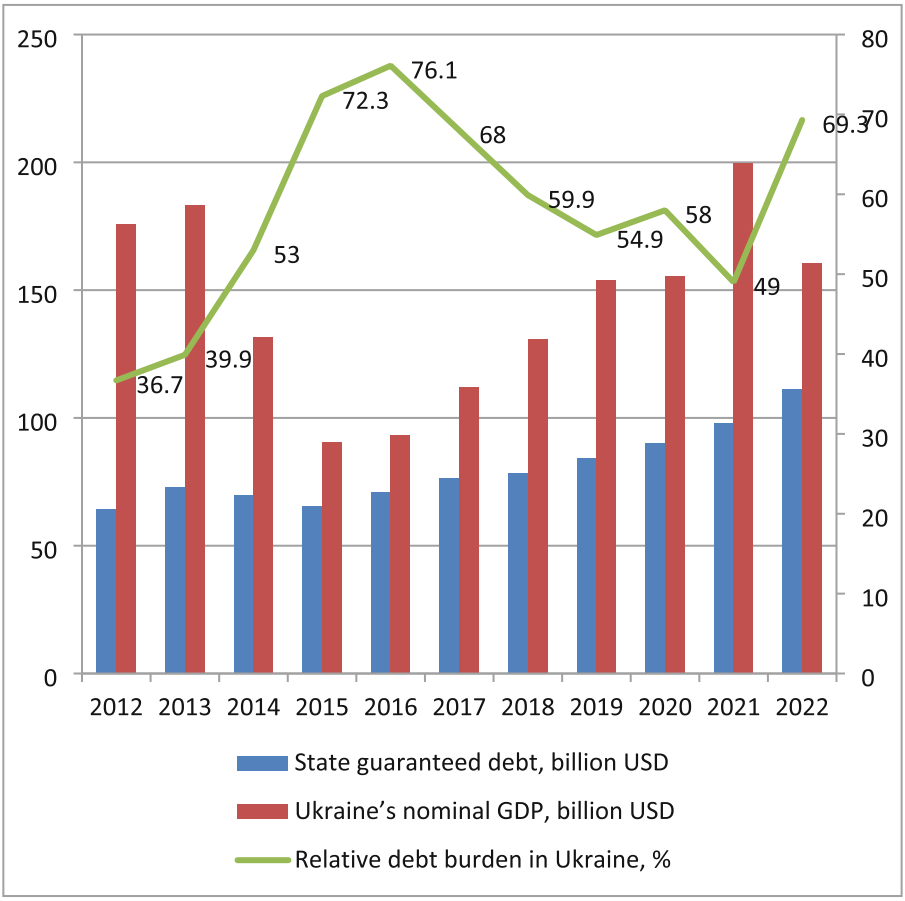


Fig. 3. Dynamics of debt burden in Ukraine in 2012–2022 Source: compiled by the authors on [15–18]

Of particular note is the period of 2019–2022, which was marked by an unprecedented increase in the debt security indicator, while it remained within the danger zone. This happened despite an increase in borrowings to finance current expenditures due to a decline in revenues. Moreover, there was an increase in the debt-to-GDP ratio, which was caused by the devaluation of the national currency. In other words, these trends indicate a difficult situation in Ukraine's debt security and point to the need to take measures to improve it.

In 2012–2013 and 2019–2020, the dynamics of the banking security level only slightly exceeded the integral financial security indicator, reflecting the negative impact of the banking system on the overall assessment. During 2015–2022, there was an upward trend in the value of this indicator, which showed an increase in the level of danger in the banking system: from an unsatisfactory level in 2015 (0.19) to a critical level in 2022 (0.43). This was caused by a 20% increase in the share of overdue loans in 2022 compared to 2015, a 30% decrease in the volume of bank loans and deposits in foreign

Table 2. Assessment of Ukraine's financial security components for 2012–2022*

Financial security components	Years										
	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
Banking	0.61	0.53	0.29	0.19	0.23	0.35	0.38	0.45	0.45	0.40	0.43
Currency	0.60	0.44	0.26	0.36	0.38	0.40	0.42	0.41	0.46	0.49	0.47
Budgetary	0.30	0.55	0.54	0.46	0.45	0.62	0.62	0.64	0.66	0.67	0.34
Debt	0.27	0.24	0.036	0.024	0.15	0.20	0.17	0.42	0.39	0.39	0.37
Non-banking financial market	0.51	0.49	0.46	0.34	0.32	0.30	0.31	0.35	0.34	0.34	0.44
Monetary and credit security	0.40	0.59	0.70	0.46	0.55	0.54	0.42	0.37	0.38	0.39	0.37

acompiled by the authors based on data from [15–18]

currency, a 9% increase in the share of foreign capital in the authorized capital of banks, and a 12% increase in the share of the five largest banks in the total assets of the banking system.

Trends in the level of monetary and credit security were low and unstable throughout the period under review. Short-term growth was observed only in 2013–2014 and 2016–2017, which makes it possible to classify this indicator as problematic in the structure of the state's financial security.

During 2012–2022, the level of security of the non-banking financial market demonstrated stable dynamics, which were proportional to changes in the integral financial security indicator. This study suggests that the non-banking sphere of the financial market has limited potential to counteract economic factors that reduce financial security. At the same time, it should be taken into account that the security of the non-banking financial market has the least weight in the integral indicator.

Before assessing the level of financial security of the state, it is advisable to analyze the level of the shadow economy, since it significantly affects the financial security indicator. The growing shadowing of Ukraine's economy and the emergence of new schemes of economic activity outside the official sector require intensification of measures to counteract this destructive phenomenon. These problems were especially intensified during Russia's full-scale war against Ukraine, which gave them a qualitatively new meaning, and the existing de-shadowing measures proved to be ineffective and inefficient (Fig. 4).

When analyzing the dynamics of Ukraine's shadow economy over the period 2012–2022 (Fig. 4), it should be noted that there were no clearly expressed stable trends in the change of the integral indicator of the economy's shadowing. The study shows that the growth of shadow processes in the national economy correlates with periods of intensification of socio-economic and socio-political crises in the country. In the context of this analysis, several key crisis stages can be identified: 2013–2015: This period was characterized by significant political transformations, including the events of the Revolution of Dignity, which led to a change in the power structure. Moreover, the annexation of the Autonomous Republic of Crimea and the beginning of Russia's hybrid aggression against

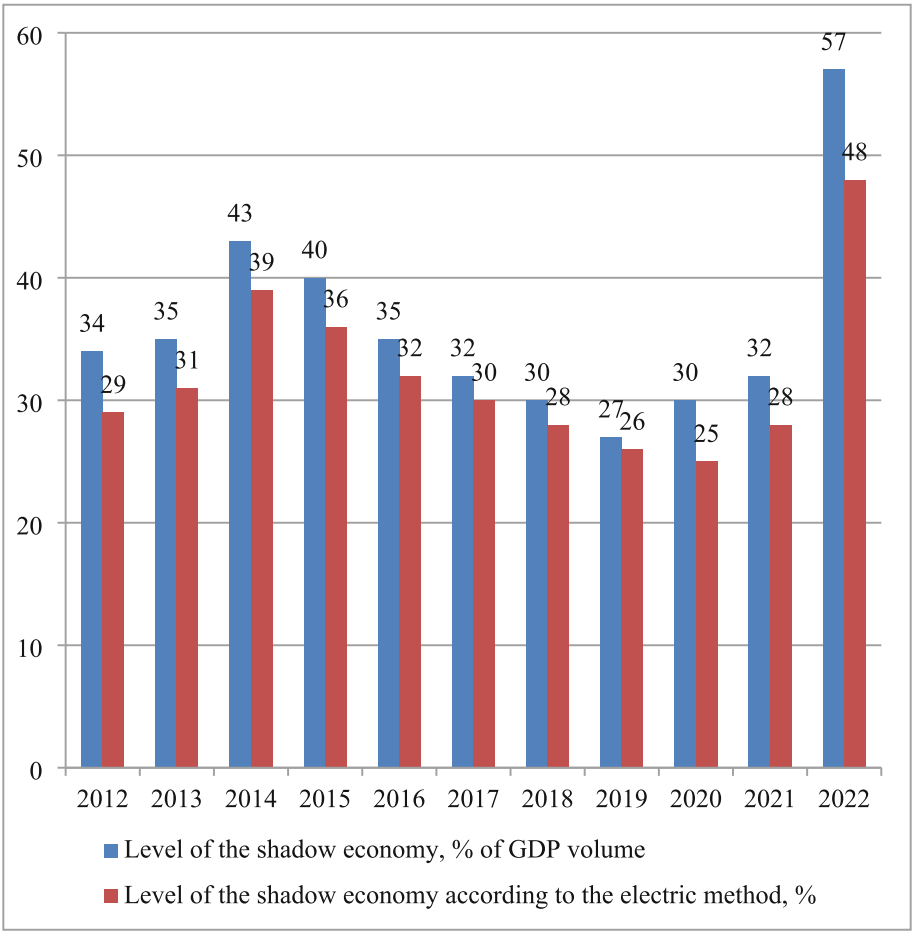


Fig. 4. Dynamics of the integral indicator of the shadow economy in Ukraine, %* *compiled by the authors on [15, 16].

Ukraine significantly affected the economic situation; 2020–2021: this was the period of the global COVID-19 pandemic, which led to the introduction of strict quarantine measures that significantly affected the activities of business entities and their financial and economic operations; 2022: this year was marked by the beginning of Russia's full-scale military invasion of Ukraine. The consequences were massive destruction of infrastructure, loss of human resources, and restricted access of the Ukrainian government to large areas of the country, which led to significant changes in the economic structure and functioning. In our opinion, these selected periods had a significant impact on the level of economic shadowing, demonstrating the interconnection between socio-political events and economic processes in the country.

The calculation of the level of shadow economy using the electric method is based on comparing the growth of domestic electricity consumption with the growth of GDP,

assuming that the increase in domestic electricity consumption should be proportional to the increase in real GDP. If there is an excess of growth in domestic electricity consumption over GDP growth, it is assumed that electricity is used for production in the shadow economy. In our opinion, this assumption is correct, provided that the technological level of production of goods, works and services remains unchanged.

When investing in projects aimed at improving energy efficiency and implementing energy-saving measures, there is an increase in the gap between the index of changes in domestic electricity consumption and the index of changes in GDP, which is due to scientific and technological progress. However, this approach has a drawback, as it does not take into account the growth of electricity consumption by the population due to the purchase of various household appliances [19–21].

According to the calculation methodology, the dynamics of the shadow economy growth according to this method is explained by the fact that the decline in domestic electricity consumption (excluding utility needs) is less than the decline in GDP, or the growth in domestic electricity consumption is accompanied by a decline in GDP.

According to this method, the dynamics of the shadow economy growth remained, driven by rising demand for energy resources in the context of increased economic activity in 2021, growing national investment demand, and favorable conditions in foreign markets.

Analyzing the state and trends in the level of financial security of Ukraine for the period 2012–2022 (Fig. 5), it can be noted that it decreased during periods of shadow economy growth in the state, which indicates their direct dependence and interconnection. In particular, in 2014–2015, the level of financial security decreased to 38% and 36%, respectively. Later, as it can be seen, the level of financial security of the state decreased from 42% to 38% in 2020, and the critically low level reached 29.6% in 2022.

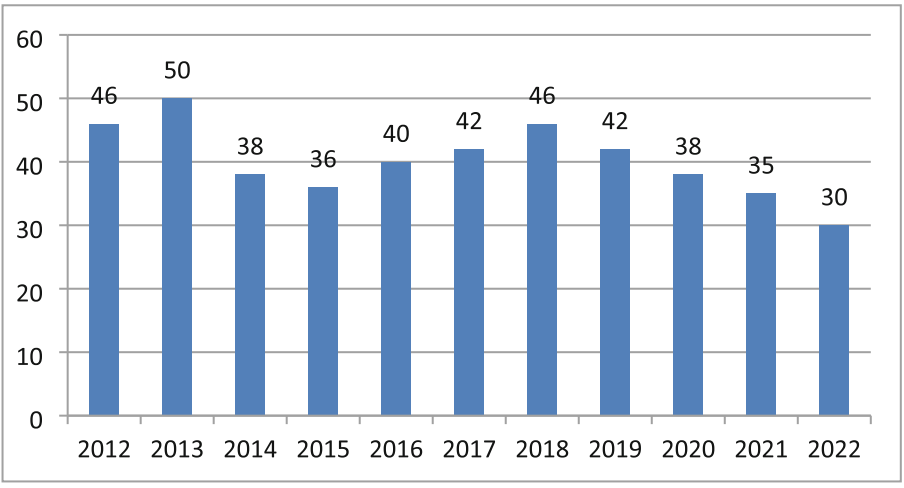


Fig. 5. Dynamics of changes in the level of the integral indicator of financial security of Ukraine in 2012–2022, % * compiled by the authors on [15–18]

In general, we can note a deterioration in the state of national financial security during 2012–2022, which is manifested in a decrease in the value of the integral financial security indicator from 0.46 in 2012 to 0.30 in 2022.

We are convinced that in the context of the current challenges faced by Ukraine, budgetary security holds an important place in financial security and plays a key role in ensuring the financial stability and solvency of public finances, allowing public authorities to effectively perform their functions. In accordance with these tasks, a set of budget security indicators is formed to assess the level of budget revenues, the balance of revenues and expenditures, the ratio of budget revenues to GDP growth rates, the degree of financial decentralization in the state, and the differentiation of regions by the level of budget expenditures [22–28]. The results of calculating the budget security indicators are shown in Fig. 6.

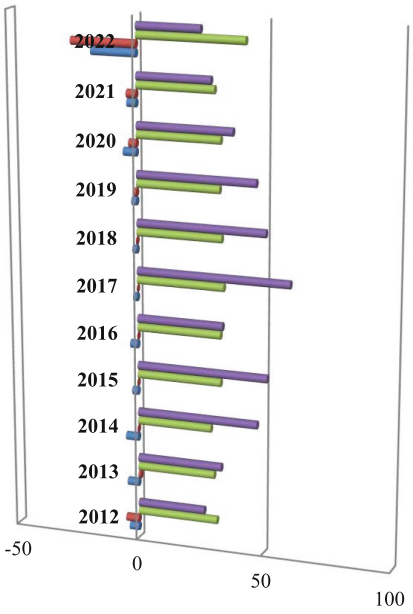
The indicator of the state budget deficit/surplus to GDP ratio allows assessing existing macroeconomic and budgetary challenges and risks in general. The analysis of this indicator for 2012–2022 shows that it is outside the optimal range, which reflects the negative impact of the government's fiscal policy on the current state of the financial system, balance of payments, and domestic demand.

The optimal value of the share of the deficit of budgetary and extra-budgetary funds of the general government sector in GDP is considered to be zero, while exceeding 3% indicates a critical state of the indicator. During 2012–2022, there was a fluctuation in the value of this indicator between the optimal and dangerous levels. This dynamic is caused by the provision of budget support to extra-budgetary funds, in particular, the allocation of a significant part of the budget to cover the Pension Fund's deficit. These factors also affect the process of attracting loans from the International Monetary Fund.

The analysis of the level of GDP redistribution through the consolidated budget is an important aspect of assessing budget security, as it reflects the amount of financial resources redistributed through the public financial system and indicates the degree of centralization of the state's financial system. In the period from 2012 to 2017, there was an increase in this indicator (from 31.6% in 2012 to 34.09% in 2017), which shows an approach to a dangerous level. This was caused by numerous problems in the fiscal system as well as in customs and tax administration. However, in 2018–2021, the value of the indicator decreased (from 33.28% in 2018 to 30.45% in 2021), which indicates an improvement in budgeting tools and the process of budget allocation. In 2022, the indicator reached a critical level of 42.31%. In our opinion, this is due to a decline in GDP by UAH 268.5 billion due to the political and economic situation in the country. These results highlight the discouraging nature of the indicator, which, in turn, negatively affects the overall level of national budget security.

4 Conclusions

The results of the conducted analysis of the current state and dynamics of development of the financial security of the state show that financial security in all the studied sectors is at dangerous and critical levels. The financial system is largely dependent on external factors and is unable to effectively protect national interests and the stability of the state's financial sector. Although in 2022 the level of financial security reached its highest value



	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
Ratio of total payments on public debt servicing and repayment to state budget revenues, %	26.4	33.24	47.26	51.1	33.62	59.75	50.28	46.5	37.5	29.05	25.07
The level of GDP redistribution through the consolidated budget, %	31.6	30.44	29.11	32.95	32.85	34.1	33.28	32.45	32.82	30.45	42.31
Deficit/surplus of budgetary and extra-budgetary funds of the general government sector, % of GDP	-5.2	0.9	-0.23	-0.11	0.13	0.03	-0.21	-0.96	-3.02	-3.81	-25.59
State budget deficit to GDP ratio, %.	-3.8	-4.46	-4.99	-2.29	-2.95	-1.6	-1.7	-1.97	-5.19	-3.6	-17.63

Fig. 6. Dynamics of budget security indicators of Ukraine for 2012–2022* *compiled by the authors on [15–18]

for the period from 2012 to 2022, there are potential risks and threats that do not allow us to speak of the stability of Ukraine's financial security in the long or even medium term.

The socio-political crisis of 2013–2014 in Ukraine triggered a deep economic crisis that negatively affected the banking sector. Internal challenges and threats, including the military conflict in the eastern part of the state, had a negative impact on economic development, leading to a decline in the confidence of financial services consumers in banks. Economic activity also declined during the coronavirus pandemic in 2020.

An important factor that determines the risks and threats, problems in the field of budgetary and then financial security is the imbalance and high level of the state budget

deficit, which causes the growth of the state budget deficit of Ukraine as compared to GDP. The period of 2012–2022 can be characterized by a low level of domestic budgetary security, which is within dangerous limits (from 29.6 in 2012 to 34.3 in 2022), as a significant number of indicators exceed the respective permissible values, and the level of budget revenues is insufficient to cover even the most important state expenditures, which leads to the accumulation of the state's debts and a decrease in the level of budgetary and financial security of the state as a whole, making it a hostage to external creditors.

Our analysis of financial security for 2012–2022 allowed us to identify key shortcomings and problems in ensuring the financial security of the state, such as: an increase in the state budget deficit; increased debt dependence on external creditors; and increased inflationary factors. In our opinion, the main threats to national financial security include:

- a critically high level of burden on the budget system at the national, regional, and local levels;
- rising inflation;
- depreciation of the national currency;
- rapid growth of public debt and a decrease in the number of financing sources;
- inefficiency and irrationality of the insurance, banking and tax systems

Thus, this threatening state of national financial security indicates the existence of key problems in the functioning of the financial sector and requires the formation of a system of effective measures to counteract the destabilizing impact of risks and threats.

References

1. Romaniv, V.V., Doroshenko, N.O.: Financial security of Ukraine. Ekonomika, Upravlinnia ta Administruvannia. **4**(90), 230–235 (2019)
2. Abuselidze, G., Zoidze, G.: Sustainable economic development, international security and technology transfer in the age of artificial intelligence. Springer Proceedings in Business and Economics, 101-113 (2025). https://doi.org/10.1007/978-3-031-88052-0_7
3. Varnalii, Z., Tomashevskyi, T.: Financial Security of Ukraine in Hybrid War. Bull. Taras Shevchenko National Univ. Kyiv, Economics (2019) .https://doi.org/10.17721/1728-2667.2019/204-3/1
4. Shkolnyk, I., Kozmenko, S., Polach, J., Wolanin, E.: State financial security: comprehensive analysis of its impact factors. J. Int. Stud. **13**(2), 291–309 (2020). https://doi.org/10.14254/2071-8330.2020/13-2/20
5. Brezhnieva-Yermolenko, O.V., Kharytonenko, A.V.: Assessment of the impact of public debt on the state's financial security. Ekonomichnyi Visnyk Dniprovskoho Derzhavnoho Tekhnichnoho Universytetu. **2**(3), 46–53 (2021). https://doi.org/10.31319/2709-2879.2021iss2(3).254827pp46-53
6. Moroz, S.V.: Zaborhovanist Ukrainy yak zahroza natsionalnii bezpetsi [Ukraine's debt as a threat to national security]. Prychornomorski Ekonomichni Studii. **24**, 65–70 (2017). [in Ukrainian]
7. Davydenko, N., Abuselidze, G., Titenko, Z., Tsiukalo, L., Sydorenko, O.: The impact of state debt on the financial security of a country. Stud. Syst., Decision Control. **659**, 549–561 (2026). https://doi.org/10.1007/978-3-032-15820-8_46

8. Fominykh, M., Parfeniuk, A.: (2021). Assessment of Ukraine's debt security and ways to strengthen it. Naukovi zapysky Natsionalnoho universytetu «Ostrozka akademiia». Seriia «Ekonomika», 21(49), 109–113 (2021). https://doi.org/10.25264/2311-5149-2021-21(49)-109-113 [in Ukrainian].
9. Davydenko, N., Mykhaylichenko, M., Titenko, Z., Tsiukalo, L.: External debt management in the system of financial security of the state. WSEAS Trans. Bus. Econ. **20**, 144–153 (2023)
10. Korobtsova, D.V.: Legal support for the financial security of the state under martial law. Elektronne naukove vydannia «Analitychno-porivnialne pravoznavstvo». **2**, 141–146 (2022) https://journal-app.uzhnu.edu.ua/article/view/261779/258188
11. Topical challenges and threats to Ukraine's economic security under martial law. Natsionalnyi instytut stratehichnykh doslidzhen. https://niss.gov.ua/publikatsiyi/analitychni-dopovidi/aktualni-vyklyky-ta-zahrozyekonomichniy-bezpetsi-ukrayiny-v [in Ukrainian].
12. Vlasiuk, O.S.: Topical Issues of Ukraine's Financial Security in the Context of Post-Crisis Transformation. Kyiv, NISD (2014)
13. On approval of the Methodological Recommendations for Calculating the Level of Economic Security of Ukraine No. 1277 dated 29.10.2013 (Verkhovna Rada Ukrainy, 2014). https://zakon.rada.gov.ua/rada/show/v1277731-13#Text
14. Koval, Ya., Yurkiv, N.: The state policy in the field of providing financial security as a component of national security: the experience of the European Union countries. Public Admin. Law Rev., **2**, 10–18 (2020). https://doi.org/10.36690/2674-5216-2020-2-10
15. Debt policy: Ministry of Finance of Ukraine. https://mof.gov.ua/uk/borgova-politika
16. Statistical information. State Statistics Service of Ukraine. https://ukrstat.gov.ua
17. Statistics. National Bank of Ukraine. https://bank.gov.ua/ua/statistic
18. State budget: Ministry of Finance of Ukraine. https://mof.gov.ua/uk/state-budget.
19. On amendments to the guidelines for calculating the level of the shadow economy. Ministerstvo ekonomiky Ukrainy. https://www.me.gov.ua/Documents/Detail?lang=uk-UA&id=dce6fa7f-5a76-4112-8740-a8e24533a24f&title=NakazMinekonomikiVid20-01-2021-104-proVnesenniaZminDoMetodichnikhRekomendatsiiSchodoRozrakhunkuRivniaTinovoiEkonomiki
20. Khavanov, A.V.: Criteria and indicators of economic security in the context of public debt and their impact on the Ukrainian economy. Biuleten Mizhnarodnoho Nobelivskoho ekonomichnoho forumu. **1**(5), 374–382 (2012)
21. Davydenko, N., Abuselidze, G., Hordei, O., Patsai, B., Novytska, O.: The sustainability of financing for the demining process in Ukraine's territory. E3S Web Conf. **694**, 04011 (2026). https://doi.org/10.1051/e3sconf/202669404011
22. Stoiko, O. Ya., Shubenko, I. A.: assessment of the state of Ukraine's public debt and directions for its optimization. Problemy ekonomiky, **1**(47), 123–133 (2021). https://doi.org/10.32983/2222-0712-2021-1-123-133
23. Davydenko, N., Wasilewska, N., Boiko, S., Wasilewski, M.: Development of rural areas in Ukraine in the context of decentralization: an empirical study. Sustainability. **14**(11), 6730 (2022). https://doi.org/10.3390/su14116730
24. Abuselidze, G.: The intergovernmental relations and their regulation in the context of decentralization of fiscal policy. E3S Web. Conf. **280**, 02010 (2021). https://doi.org/10.1051/e3sconf/202128002010
25. Abuselidze, G.: Local budget revenue formation reality, obstacles and prospects in Georgia: an empirical examination and creative approach. Lect. Notes Comput. Sci. **13957**, 197–209 (2023). https://doi.org/10.1007/978-3-031-36808-0_13
26. Mohapi, T.J., Sambo, V.: Fiscal decentralisation in Lesotho: a closer look at intergovernmental fiscal transfers. African J. Governance Develop. **14**(1), 27–47 (2025)

27. Ivanyuta, P., Kartashov, E., Datsii, N., Kovalskyi, M., Abuselidze, G., Aleinikova, O.: Right ownership as the particularization of territorial formations in the conditions of decentralization reform of Ukraine. Lect. Notes Comput. Sci. **13376**, 126–143 (2022). https://doi.org/10.1007/978-3-031-10450-3_10
28. Oluyko, V., Samsin, I., Stohova, O., Filipenko, T., Filipenko, A.: Fiscal decentralization practices in developing countries. Cuestiones Políticas. **40**(73), 128–150 (2022)

Bayesian Models to Predict Above Ground-Biomass

Ricardo Coelho[1(✉)], Isabel Natário[1,2], and Silvia Fraile[3]

[1] NOVA MATH – Center for Mathematics and Applications, Nova University of Lisbon, Caparica, Portugal
rpe.coelho@campus.fct.unl.pt
[2] Department of Mathematics, Nova School of Science and Technology, Nova University of Lisbon, Caparica, Portugal
icn@fct.unl.pt
[3] GEOSAT SATELITES SLU, Boecillo, Valladolid, Spain
silvia.fraile@geosat.space

Abstract. Accurate estimation of Above-Ground Biomass (AGB) plays a key role in monitoring forest carbon stocks and supporting climate mitigation policies. This study compares four Bayesian models for AGB prediction, a Geostatistical model (BGM), a Bayesian Additive Regression Trees (BART) model, the recently proposed BARTSIMP model, and a novel Hybrid BART (HBART) framework inspired by Random Forest–Ordinary Kriging. The HBART model combines the flexibility of BART in capturing non-linear relationships with a spatial model for residual dependence. These models are applied to predict AGB in forest areas of the *Páramos-Valles* region (Spain). To this end, remote sensing data from the GEOSAT-2 satellite are integrated with field inventory data from the Spanish National Forest Inventory, including satellite variables such as reflectance bands, vegetation indices, and texture. In addition, topographic covariates such as altitude, slope, and aspect are also considered. Predictive performance is assessed using spatial block cross-validation and a set of accuracy and uncertainty metrics. Results show that the BGM achieves the best point prediction accuracy, while HBART provides the most reliable uncertainty quantification, with substantially higher coverage probabilities.

Keywords: Above-Ground Biomass · Bayesian machine learning · Geostatistical model · Remote sensing data

1 Introduction

Monitoring carbon sequestration plays a crucial role in climate change mitigation and ecosystem management. Above-Ground Biomass (AGB) is an indicator that well describes the amount of carbon present in forests and is widely used [17]. AGB can be estimated using direct or indirect methods [18]. Direct methods involve the destruction of trees while indirect methods do not. Among indirect methods, the allometric equations are quite used. These equations use field

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 660–678, 2026.
https://doi.org/10.1007/978-3-032-30491-9_41

data about characteristics of trees (for example, diameter at breast height, tree height, among others) to estimate AGB. Although this method provides reliable estimates at the plot level, it is typically labor-intensive, costly, and spatially limited, thus making it difficult to apply over large areas. Recent advances in remote sensing technologies have provided a faster and more cost-effective alternative. Satellite imagery enables the extraction of variables, including spectral bands, vegetation indices, and texture measures, which are commonly used as covariates in AGB estimation. Integrating field measurements with variables derived from remote sensing data enables the use of statistical models and Machine Learning (ML) models to produce estimates of AGB.

Several ML models such as Random Forest (RF), Support Vector Machines (SVM), and Gradient Boosting, have been applied to AGB prediction using remote sensing data [10,17]. Although these models can capture complex non-linear relationships between AGB and covariates, many traditional ML approaches do not explicitly account for predictive uncertainty, which is particularly important in environmental applications. Bayesian ML models provide a useful alternative, as they incorporate uncertainty both in the model parameters and in the predictions. One of the most widely used Bayesian ML models is the Bayesian Additive Regression Trees (BART) model, proposed by Chipman et al. (2010) [2]. This model represents the response variable as the sum of multiple regression trees, allowing flexible non-linear relationships and interactions between covariates.

In recent years the BART model has been applied to spatial data [16], although without accounting for spatial autocorrelation. For example, Yankey et al. (2024) [20] applied the BART model to disaggregate census population data from district-level totals to high-resolution grid cells. Zhang et al. (2020) [22] applied the BART model to environmental data, for predicting the concentration of the air pollutant $PM_{2.5}$. This can lead to underestimated prediction uncertainty and reduced predictive performance when modelling spatial data [17]. Also, AGB in forested areas generally exhibits spatial heterogeneity and autocorrelation, and in some cases, non-stationarity. Moreover, remote sensing variables are themselves spatially and temporally correlated [6,19].

To overcome this limitation, Jiang et al. (2025) [7] developed the Bayesian Additive Regression Trees for Spatial INLA Modelling and Prediction (BART-SIMP) model. This model combines the BART model with spatial modelling of spatial autocorrelation. Geostatistical modelling provides an alternative approach by treating AGB as a continuous spatial process observed with error at discrete locations. These models are advantageous because they naturally account for spatial dependence, integrate covariates, and quantify uncertainty, which is essential for informed decision-making [5]. Unlike most ML methods, geostatistical models explicitly model spatial autocorrelation, providing interpretable uncertainty estimates.

Recognizing the strengths of both statistical and ML approaches, hybrid methods have been proposed, as the Random Forest Ordinary Kriging (RFOK), in which the deterministic component is modelled using the Random Forest (RF)

model, while Ordinary Kriging interpolates the residuals obtained from the RF model [21]. Inspired by this, we propose in this study a hybrid BART model to improve predictive performance and capture spatial patterns in AGB.

The main purpose of this study is to evaluate and compare different modelling strategies for predicting AGB using remote sensing data. Specifically, the goal is to assess the predictive performance of four approaches: a Bayesian Geostatistical Model, the BART Model, the hybrid BARTSIMP Model, and a new hybrid BART framework. This study applies the BART and BARTSIMP models to AGB estimation for the first time. The study area corresponds to the Forest Area of *Páramos-Valles*, located in the province of *Palencia*, within the autonomous community of *Castilla y León* in northwestern Spain.

The paper is organized as follows: Sect. 1 provides an introduction to the work; Sect. 2 describes the datasets; Sect. 3 presents the models considered, together with the evaluation metrics used for their assessment; Sect. 4 presents the results; and Sect. 5 discusses the main findings and concludes.

2 Data

In this study three datasets are used to estimate AGB(t/ha): field data from Fourth Spanish Forest Inventory (*Cuarto Inventorio Forestal Nacional* - IFN4); Forest Map of Spain (*Mapa Forestal de España* - MFE); and remote sensing data from GEOSAT-2 satellite, operated by the company GEOSAT.

2.1 Ground Data

The study area is depicted in Fig. 1.

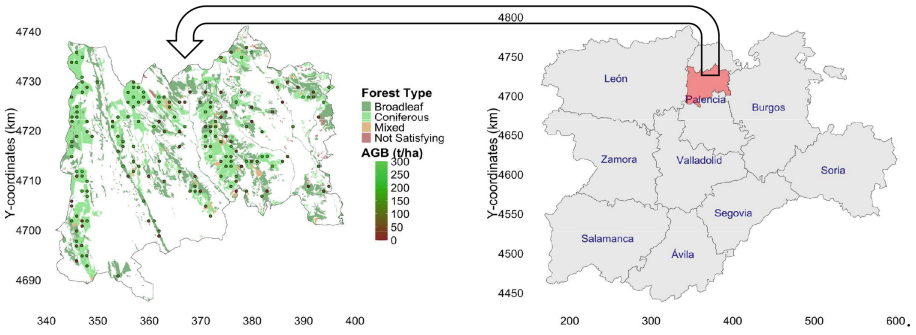


Fig. 1. Area of Interest: The left panel illustrates the forested areas along with the *estadillos*, the right panel depicts the provinces within *Castilla y León*.

Data from the IFN4 were used to calculate the response variable, AGB(t/ha). These field data were acquired and processed between 2008 and 2018, corresponding to one of the IFN 10-year cycles [3]. The inventory collects information

about trees within permanent sampling plots, the *estadillos*. Each *estadillo* is a circular plot of radius 25 m, georeferenced by its center coordinates.

Each *estadillo* is divided into four concentric circles of different radius [3]. For each tree, the IFN records information about its characteristics. Only trees within specific diameter at breast height classes are measured in each circle. AGB for individual trees is calculated using allometric equations, with an expansion factor used to extrapolate AGB to the *estadillo* level (t/ha). Ground data may contain uncertainty associated with measurement errors, which propagates into AGB estimates through allometric equations. The total AGB of an *estadillo* is obtained by adding the weighted contributions of all trees within the *estadillo*. Only *estadillos* within Forest Areas, as defined by the MFE (i) tree cover fraction exceeding 20%, ii) mature tree height greater than 3 m, and iii) a minimum area of 1 ha), were included. All others were excluded.

In the *Parámos-Valles* region, a total of 180 *estadillos* belong to the Forest Areas, of which 105 belong to Coniferous forests, 59 belong Broadleaf forests and the remainder belong to Mixed forests.

2.2 Remote Sensing Data

Remote sensing data used in this study were obtained from the GEOSAT-2 satellite, with imagery acquired between May and August 2020. The satellite collects information in five spectral bands: one panchromatic band and four multispectral bands (Red, Blue, Green, and Near Infrared (NIR)).
The multispectral bands have a spatial resolution of 3 m. Several covariates were derived from the multispectral bands, including the reflectance bands (NIR, Red, Green, and Blue), vegetation indices, such as the Normalized Difference Vegetation Index (NDVI), Enhanced Vegetation Index (EVI), Green Chlorophyll Index (GCI), and the Normalized Difference Water Index (NDWI) and texture metrics, specifically homogeneity. These variables were the covariates considered in this study.

In addition, topographic covariates as Elevation, Slope and Aspect were included. These variables were obtained from the Digital Terrain Model (DTM) produced by the *Centro Nacional de Información Geográfica* (CNIG) of Spain [13], developed as part of the *Plan Nacional de Ortofotografía Aérea* (PNOA) and based on LiDAR data from the First LiDAR Coverage.

2.3 Integrate Ground Data with Remote Sensing Data

To integrate the datasets from different sources, it was necessary to perform a spatial alignment between the remote sensing data and the field data. This process involves orthorectify the images and align them with the study coordinate reference system (EPSG:25830) using ground control points and a 5 m Digital Elevation Model (DEM) to compensate for terrain related distortions. For each *estadillo*, only pixels entirely within the boundaries were selected, and any partially overlapping pixels were excluded. The remote sensing metrics for each *estadillo* were then considered as the median values of the selected pixels.

3 Methods

3.1 Bayesian Geostatistical Model (BGM)

Geostatistical models are designed to capture spatial dependence in data observed at known locations [9]. Let $\mathcal{D} \subset \mathbb{R}^2$ denote a fixed and continuous spatial domain [5]. Spatial phenomena observed over $\mathcal{D}$ can be described as realizations of a stochastic process $\{U(\mathbf{s}) : \mathbf{s} \in \mathcal{D}\}$. Geostatistical (point-referenced) data consist in observations collected at a finite set of locations $\{\mathbf{s}_1, \ldots, \mathbf{s}_n\} \subset \mathcal{D}$, yielding measurements $\{y(\mathbf{s}_i)\}_{i=1}^n$, which can be viewed as a partial realization of the process $U(\mathbf{s})$. This process is often modelled as a Gaussian Field (GF), where the assumptions of second-order stationarity and isotropy are assumed.

The spatial dependence can be analysed using a semivariogram, describing how variability between observations changes with distance, providing guidance for selecting appropriate covariance functions in spatial modelling [4]. In addition, Monte Carlo envelopes can be used to assess the statistical significance of spatial autocorrelation by comparing the empirical semivariogram with simulations under spatial independence hypothesis.

In geostatistical models, the linear predictor combines a deterministic component, which captures large-scale variation through covariates, with a spatially structured random effect accounting for residual spatial dependence [5]. The spatial random effect $u(\mathbf{s})$ is assumed to be a realization of a latent stochastic process $U(\mathbf{s})$. This process is commonly modelled as a GF with zero mean and covariance function $C(u(\mathbf{s}_i), u(\mathbf{s}_j))$. Under the assumptions of second-order stationarity and isotropy, the covariance structure depends only on the Euclidean distance between locations, that is, $\text{Cov}\big(u(\mathbf{s}_i), u(\mathbf{s}_j)\big) = C\big(\|\mathbf{s}_i - \mathbf{s}_j\|\big)$, where $\|\mathbf{s}_i - \mathbf{s}_j\|$ denotes the Euclidean distance between $\mathbf{s}_i$ and $\mathbf{s}_j$.

Under this framework, the Generalized Geostatistical Model is defined hierarchically:

$$\begin{aligned} y(\mathbf{s}_i) &\sim \text{Exponential-Family}\big(\mu(\mathbf{s}_i), \sigma_\varepsilon^2\big) \\ h\big(\mu(\mathbf{s}_i)\big) &= \eta(\mathbf{s}_i) \\ \eta(\mathbf{s}_i) &= \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + u(\mathbf{s}_i), \;\; i = 1, \ldots, n, \end{aligned} \tag{1}$$

where $y(\mathbf{s}_i)$ is the observed response at location $\mathbf{s}_i$, $\mathbb{E}[y(\mathbf{s}_i)] = \mu(\mathbf{s}_i)$ is its mean linked to the linear predictor $\eta(\mathbf{s}_i)$ through a link function $h(\cdot)$, β_0 the intercept, β_j the regression coefficients for covariates x_j, $u(\mathbf{s}_i)$ the spatial random effect, and σ_ε^2 the variance of response variable, if required.

Taking a Bayesian approach for inference, Rue et al. (2009) [15] developed the deterministic methodology known as Integrated Nested Laplace Approximation (INLA), which efficiently approximates the posterior distributions of model parameters and hyperparameters in hierarchical Bayesian models. INLA is implemented in the `R` software through `R-INLA` package.

The Stochastic Partial Differential Equations (SPDE) approach can be used to model the spatial random effects [9]. In this approach, the exact stationary

solution of a given SPDE corresponds to a GF with a Matérn covariance structure:

$$\text{Cov}\left[u(\mathbf{s}_i), u(\mathbf{s}_j)\right] = \frac{\sigma^2}{\Gamma(\lambda)2^{\lambda-1}}\big(\kappa||\boldsymbol{s_i} - \boldsymbol{s_j}||\big)^{\lambda} K_{\lambda}\big(\kappa||\boldsymbol{s_i} - \boldsymbol{s_j}||\big), \tag{2}$$

where, σ^2 denotes the marginal variance of the spatial process, K_λ is the modified Bessel function of the second kind of order λ, which controls the smoothness of the process ($\lambda > 0$), and $\kappa > 0$ is a scaling parameter related to the spatial range r. The term $||\boldsymbol{s_i} - \boldsymbol{s_j}||$ represents the Euclidean distance between locations $\boldsymbol{s_i}$ and $\boldsymbol{s_j}$.

Using the Finite Element Method, the GF is approximated by a discretely indexed Gaussian Markov Random Field (GMRF), through a basis function representation in $\mathcal{D}$. This approach, implemented in the `R-INLA` package, enables efficient estimation of spatial dependence in the data. A key step in this approach is the construction of a triangulated mesh that discretizes the spatial domain and defines the nodes for the approximation. The mesh design should follow the recommendations of Moraga et al. (2021) [12] and Bakka (2017) [1].

Within the Bayesian approach, prior distributions of the parameters are added to the hierarchical formulation. In this study, the default non-informative priors provided by `R-INLA` for precision parameters and fixed effects were considered. For the hyperparameters of the Matérn covariance function, Penalized Complexity (PC) priors were used to control model flexibility and avoid overfitting [9], where $P(r < 15500) = 0.3$ and $P(\sigma > 48) = 0.1$, having these values been selected from the semivariogram of the linear model residuals and the spatial distances between observation points.

3.2 Bayesian Additive Regression Trees (BART)

The Bayesian Additive Regression Trees (BART) model is a Bayesian model that uses non-parametric regression trees to approximate the function $f(x) = \mathbb{E}[y|x]$. Here $f(x)$ is represented as a sum of m regression trees, each contributing a small portion to the overall prediction. Developed by Chipman et al. (2010) [2], this model has been applied to problems where the response variable is continuous, categorical, or time-to-event [8]. The BART model can be represented as:

$$\mathbf{y} \approx \sum_{j=1}^{m} g(x; T_j, M_j) + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \text{Normal}(0_n, \sigma_{\varepsilon}^2 I_n), \tag{3}$$

where each function $g(x; T_j, M_j)$ represents the contribution of the j^{th} tree, $j = 1, \ldots, m$, and $\boldsymbol{\varepsilon}$ is the error term. Each tree is defined by its structure T_j and a set of terminal node parameters $M_j = \{\mu_{j1}, \ldots, \mu_{jb_j}\}$ assigned to its leaves [2]. The value b_j is the number of terminal nodes for the j^{th} tree.

The BART model has three main stochastic components: the residual error with variance σ_{ε}^2, the structures of the regression trees $T_1, \ldots, T_m$, and the values at the leaf nodes $M_1, \ldots, M_m$ [2]. A prior is placed over the ensemble by assuming

independence between trees, between leaf nodes conditional on their tree, and between the residual variance and the trees:

$$p((T_1, M_1), \ldots, (T_m, M_m), \sigma) = \prod_{j=1}^{m} p(T_j)\, p(M_j \mid T_j)\, p(\sigma). \tag{4}$$

The tree structure prior $p(T_j)$ favors shallow trees through a depth-dependent splitting probability of the form $\alpha(1+d)^{-\beta}$, where $\alpha \in]0,1[$, $\beta \in [0,\infty[$, and node at depth $d \in \{0,1,2\}$, which regularizes the model and helps prevent overfitting [2,8]. At each internal node, splitting variables are selected uniformly, and cutpoints are chosen uniformly from the set of available values. Conditional on the tree structure, terminal node parameters are assumed independent and normally distributed, $\mu_{jl} \sim N(\mu_\mu/m, \sigma_\mu^2)$, $l = 1, \ldots, b_j$, shrinking individual tree contributions toward zero. The residual variance is assigned an inverse-gamma prior, $\sigma_\varepsilon^2 \sim \text{Inv-Gamma}\left(\frac{\nu}{2}, \frac{\nu\lambda^2}{2}\right)$, typically chosen to be weakly informative. The ν denotes the degrees of freedom and the scale parameter λ is calibrated from the data so that, by default, there is a prior probability of 0.90 that the BART model improves upon the root mean square error (RMSE) of a linear regression model.

Model estimation is performed using Markov Chain Monte Carlo (MCMC), specifically a backfitting Gibbs sampler that iteratively updates each tree conditional on the others [2,8]. This procedure generates posterior samples for all model parameters as well as for predictions. Posterior predictions are obtained by averaging over the MCMC samples, while uncertainty is quantified using credible intervals, typically constructed from the 2.5th and 97.5th percentiles of the posterior distribution.

The parameter m in the BART model can be treated as an unknown parameter with a prior distribution, following a fully Bayesian approach. This implementation has very high computational costs. Thus, an alternative and more computationally efficient strategy is to select the value of m using cross-validation methods. However, this strategy often result in very large number of trees, which results in very high computational costs. Consequently we have chosen to follow the advice to reduce this number to a more moderate value [2].

In this study, the BART model was implemented using the `bartMachine` package in `R`. The optimal number of trees, $m = 50$, was selected via 10-fold cross-validation by minimizing RMSE. The default values of the hyperparameters α and β are 0.95 and 2, respectively. In fitting this model, a total of $N = 4000$ MCMC iterations were performed, of which the first $B = 2000$ were discarded as burn-in, while the remaining $N - B = 2000$ were used for posterior inference. To interpret the fitted model, Partial Dependence Plots (PDP) were used. These plots represent the marginal effect of a covariate on the predicted response, averaging over the observed values of the remaining covariates [8]. PDP provide a useful tool for visualizing non-linear relationships and identifying regions where covariates have a stronger influence on the response variable.

3.3 Hybrid Bayesian Additive Regression Trees (HBART)

Using the BART model to analyse geostatistical data has the drawback of not explicitly accounting for the spatial autocorrelation present in the observations. To address this limitation, we propose a Hybrid Bayesian Additive Regression Trees (HBART) model, that is inspired in the traditional Hybrid ML models used in geostatistics, such as the Random Forest-Ordinary Kriging (RFOK) model [21].

Similarly to the RFOK hybrid framework, the proposed Hybrid BART model is estimated in two stages. In the first stage, a BART model is fitted to the response variable using the available covariates. This step captures potentially complex non-linear relationships between the covariates and the response variable. However, because BART does not explicitly model spatial dependence, the residuals obtained from this first stage may still exhibit spatial autocorrelation. Let $\varepsilon(\mathbf{s}_i) = y(\mathbf{s}_i) - \hat{y}_{BART}(\mathbf{s}_i)$ denote the residual of the BART model at location $\mathbf{s}_i$. In the second stage, these residuals are modelled using a BGM just with an intercept term and spatial random effects. The final prediction at location $\mathbf{s}_0$ is obtained by combining the BART prediction with the predicted spatial residual component:

$$\hat{\mathbf{y}}(\mathbf{s}_0) = \hat{\mathbf{y}}_{\text{BART}}(\mathbf{s}_0) + \hat{\varepsilon}_{\text{BGM}}(\mathbf{s}_0), \tag{5}$$

where $\hat{\mathbf{y}}_{\text{BART}}(\mathbf{s}_0)$ denotes the posterior predictive mean obtained from the BART model and $\hat{\varepsilon}_{\text{BGM}}(\mathbf{s}_0)$ represents the posterior predictive mean of the BART residual estimated using the BGM model.

Since the BART is Bayesian model, it provides posterior samples from the predictive distribution. These samples allow the construction of credible intervals that quantify the uncertainty associated with the prediction. In this method the prediction intervals are obtained by combining the posterior predictive uncertainty from the BART model with the uncertainty associated with the spatial prediction of the residuals. In practice, posterior samples from the BART predictive distribution are added with posterior samples from the spatial residual model, yielding a posterior predictive distribution for $y(\mathbf{s}_0)$. Credible intervals are then be derived from the corresponding quantiles. For the spatial residual model, the same PC-priors as in the BGM model were used.

3.4 BART for Spatial INLA Modelling and Prediction (BARTSIMP)

The BART model presented in Subsect. 3.2 offers a flexible way to capture complex non-linear relationships and interactions between covariates. However, it does not explicitly account for spatial dependence, which can play an important role when analysing spatially referenced data. To overcome this limitation, Jiang et al. (2025) [7] developed the BART for Spatial INLA Modelling and Prediction (BARTSIMP) model. In contrast to the HBART two stage approach, BARTSIMP defines a fully joint Bayesian hierarchical model in which covariate effects and spatial dependence are estimated simultaneously.

BARTSIMP combines the sum-of-trees structure of BART for modelling covariate effects with a latent spatial GF that captures residual spatial dependence. Let y_i denote the response variable at spatial location i, $\mathbf{x}_i$ the vector of covariates, and u_i the latent spatial effect. Let $\mathbf{T} = (T_1, \ldots, T_m)$ denote the collection of tree structures and $\boldsymbol{M} = (M_1, \ldots, M_m)$ the corresponding terminal node parameters of the BART model. The model can be expressed as

$$y_i \mid u_i, \mathbf{x}_i, \mathbf{T}, \boldsymbol{M}, \sigma_\varepsilon^2 \sim N\left(\sum_{l=1}^{m} g(\mathbf{x}_i; T_l, M_l) + u_i,\ \sigma_\varepsilon^2\right), \tag{6}$$

where $\sum_{l=1}^{m} g(\mathbf{x}_i; T_l, M_l)$ is the BART sum-of-trees contribution for the covariates, u_i is the spatial random effect, and σ_ε^2 is the residual variance.

The BARTSIMP model assumes a prior independence between the tree parameters $(\mathbf{T}, \boldsymbol{M})$, the residual variance σ_ε^2, and the hyperparameters $\boldsymbol{\psi}$ of the spatial random effects modelled as a GF with the parameters of a Matérn covariance structure (marginal variance, σ^2, and spatial range, r, introduced in Subsect. 3.1):

$$p(\mathbf{T}, \boldsymbol{M}, \sigma_\varepsilon^2, \boldsymbol{\psi}) = \left[\prod_{j=1}^{m} p(T_j)\, p(M_j \mid T_j)\right] p(\sigma_\varepsilon^2)\, p(\boldsymbol{\psi})\,,$$

where the priors for $(\mathbf{T}, \boldsymbol{M}, \sigma_\varepsilon^2)$ follow the standard specification of Chipman et al. (2010) [2], as described in Subsect. 3.2. In particular, tree structures are regularized to act as weak learners, terminal node values are normally distributed, and the residual variance has a scaled inverse-gamma prior [7].

Penalized complexity (PC) priors are assigned to the spatial hyperparameters $\boldsymbol{\psi}$ of the spatial random effects u_i to provide weakly informative, interpretable priors that control model complexity while allowing sufficient flexibility [7].

Computationally, direct MCMC sampling of the high-dimensional spatial random field is prohibitive. To address this, BARTSIMP uses an INLA-within-MCMC strategy, in which the spatial random effects are not sampled explicitly, but instead integrated out through an INLA-based approximation of the marginal likelihood [7]. This approximation is then used within a Metropolis-within-Gibbs MCMC algorithm to update the remaining model parameters, including the BART tree structures and their associated parameters. By avoiding the need to directly sample the spatial field, the approach becomes computationally more efficient while still enabling coherent Bayesian inference for the joint model.

In this work, the BARTSIMP model was fitted using the `BARTSIMP` package [7] in the `R` software. The number of trees considered was the same as in the BART model. To interpret the fitted model, PDP were also used. Default priors were adopted for model fitting. For the hyperparameters of the spatial random effects, the same PC-priors as in the BGM model were used. A total of $N = 4000$ MCMC iterations were performed, of which the first $B = 2000$ were discarded as burn-in, while the remaining $N - B = 2000$ iterations were retained for posterior inference.

3.5 Evaluation of the Models

Model performance was evaluated using spatial cross-validation to account for spatial autocorrelation, since spatial data violate the independence assumption of traditional ML validation methods, such as K-fold cross-validation. Ignoring spatial dependence can lead to overly optimistic accuracy estimates and an increased risk of over-fitting [17].

The spatial cross-validation method considered in this study was K-fold spatial block cross-validation [14]. This method divides the dataset into K distinct, spatially separated folds, ensuring that training and testing data are not spatially autocorrelated. The folds were created by applying the K-means clustering algorithm [11] to the coordinates of the response variable. In this study, we set $K = 5$.

Model predictive performance was quantified using several metrics. The Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) were used to measure overall prediction accuracy by quantifying the magnitude of prediction errors. The Nash–Sutcliffe Efficiency (NSE) was used to evaluate the proportion of the data variability explained by the model in terms of Mean Square Error. The 95% Coverage Probability ($CP_{95\%}$) metric was used to assess the predictive uncertainty of the model. This metric gives the proportion of observed values that fall within the 95% prediction interval.

Figure 2 shows the spatial distribution of training and testing locations for each fold generated using the 5-fold spatial cross-validation approach.

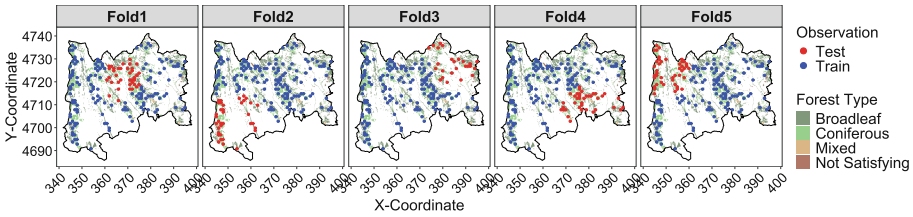


Fig. 2. Spatial distribution of training and testing locations using 5-fold spatial cross-validation.

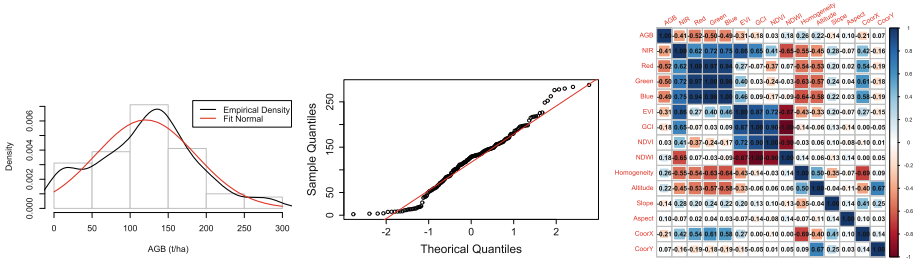
4 Results

In this Section, the results are presented. First, an explanatory analysis of the dataset used in this study is conducted, Subsect. 4.1. Subsections 4.2 and 4.3 present the predictive results obtained from the models described in Sect. 3 and the effects of each covariate in the considered models, respectively.

4.1 Explanatory Analysis

This Subsection begins with an assessment of the response variable distribution (AGB), an evaluation of multicollinearity among the covariates introduced in Sect. 2 and an analysis of residual spatial autocorrelation.

Figure 3 **a)** depicts the histogram along with the empirical distribution of AGB. Figure 3 **b)** shows a Normal Quantile–Quantile (Q–Q) plot, to evaluate the normality assumption. In general, the empirical quantiles closely follow the theoretical quantiles of the Normal distribution, suggesting that the normal approximation is reasonably appropriate, although small departures are noticeable in the distribution tails. Figure 3 **c)** presents the Spearman Correlation matrix between the response variable and the covariates, where high correlations can be observed among the reflectance bands as well as among the vegetation indices.



a) Histogram. **b)** Q-Q Plot. **c)** Spearman Correlation Matrix.

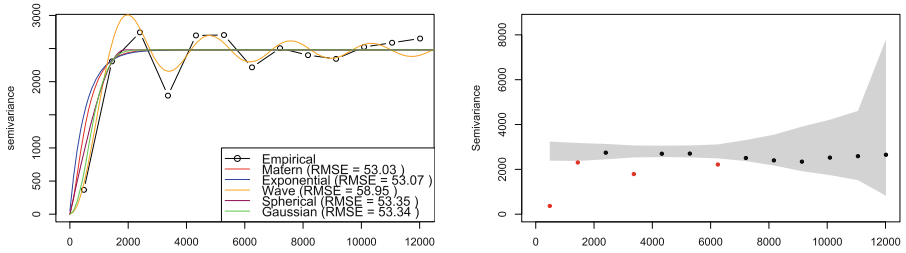
Fig. 3. Exploratory analysis: Histogram of AGB distribution (left), Q–Q plot for normality assessment (center), and Spearman correlation matrix (right).

To evaluate the normality of the response variable, a Kolmogorov-Smirnov test was done (p-value of 0.13). Thus, at a 10% significant level, it is reasonable to assume that the AGB follows a Normal distribution.

Due to the presence of high pairwise correlations among several variables, particularly the reflectance bands and vegetation indices, multicollinearity may be present, which can affect the performance of regression models. To address this issue, the Variance Inflation Factor (VIF) method [3], was applied. In this study, VIF was calculated iteratively using the `R` package `car`, with the covariate exhibiting the highest VIF removed at each step until all remaining covariates had acceptable VIF (below 10). The covariates retained were: the Blue reflectance band; NDVI and GCI vegetation indices; Homogeneity; Forest Type; and the topographic variables Altitude, Slope, and Aspect. These variables were included in all subsequent models.

To evaluate spatial autocorrelation in dataset, the empirical residual semivariogram from a linear regression model was analysed, along with its 95% Monte Carlo envelopes. Figure 4 **a)**, shows the empirical residual semivariogram

together with several theoretical models fitted to the semivariogram, including the Matérn, Exponential, Wave, Gaussian, and Spherical models. Figure 4 **b)** presents the empirical semivariogram together with its 95% Monte Carlo envelopes. From here, it can be concluded that residual spatial autocorrelation exists, since several points of the empirical semivariogram lie outside the 95% Monte Carlo envelopes.



a) Semivariogram. **b)** Envelopes.

Fig. 4. Empirical residual semivariogram and its 95% Monte Carlo envelopes: on the left, the semivariogram with fitted theoretical models (Matérn, Exponential, Wave, Gaussian, and Spherical); on the right, the empirical semivariogram with its 95% Monte Carlo envelopes.

In this study, the Matérn covariance function was chosen to model the spatial dependence structure because it provided the lowest RMSE when fitted to the empirical semivariogram, as shown in Fig. 4 **a)**, with RMSE values obtained through leave-one-out cross-validation procedure.

4.2 Predictive Results

To evaluate and compare the predictive performance of the all models considered, a 5-fold spatial block cross-validation was used. Table 1 summarizes the predictive results obtained for each of the models considered, reporting the RMSE, MAE, and NSE metrics, as well as the 95% prediction interval coverage metric ($CP_{95\%}$).

Analysing the results in Table 1, it can be observed that the BGM model achieves the lowest RMSE (23.09 t/ha) and MAE (8.45 t/ha), indicating superior performance in terms of average prediction error. Moreover, BGM also attains the highest NSE value (0.88), outperforming all other models according to this metric. Among the models based on BART, HBART shows the best performance (NSE = 0.79), followed closely by BART and BARTSIMP (both with NSE = 0.78). Regarding $CP_{95\%}$, values close to 95% indicate well-calibrated prediction intervals. HBART achieves the highest coverage (92.20%), suggesting better uncertainty representation. BART (89.03%) and BGM (88.89%) show similar performance, while BARTSIMP (76.11%) exhibits a substantially lower value, indicating a tendency to underestimate predictive uncertainty.

Table 1. Predictive results of all 4 models: BGM: Bayesian Geostatistical Model; BART: Bayesian Additive Regression Trees; HBART: Hybrid BART; and BARTSIMP: BART for Spatial INLA Modelling and Prediction.

	Model			
Metric	BGM	BART	HBART	BARTSIMP
RMSE	23.09 t/ha	58.19 t/ha	58.25 t/ha	58.06 t/ha
$CP_{95\%}$	88.89%	89.03%	92.20%	76.11%
MAE	8.45 t/ha	43.69 t/ha	43.72 t/ha	43.96 t/ha
NSE	0.88	0.78	0.79	0.78

Predictive Maps. Figure 5 shows the predictive maps for each of the four models considered in this study.

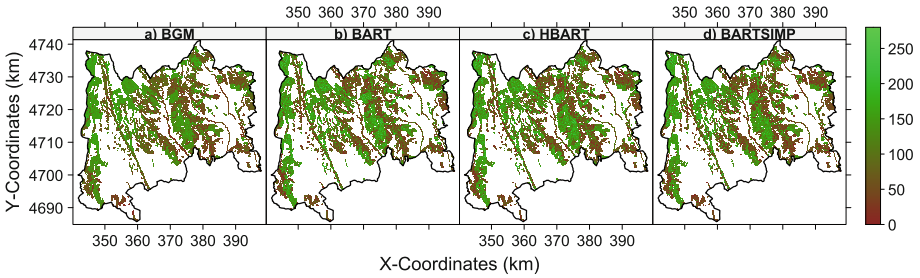


Fig. 5. Prediction maps of AGB(t/ha) in *Páramos-Valles*, considering the 4 models: BGM: Bayesian Geostatistical Model; BART: Bayesian Additive Regression Tree; HBART: Hybrid BART; and BARTSIMP: BART for Spatial INLA Modelling and Prediction.

Analysing the predictive maps AGB in *Páramos-Valles* (Fig. 5), it can be seen that all models show similar spatial patterns, with higher AGB values mainly in the north-eastern and central areas of the study region.

Figure 6 presents the mean GF estimated by the BGM and BARTSIMP models (Fig. 6 **a)** and Fig. 6 **c)**, respectively), and Fig. 6 **b)** shows the residuals of the BART model interpolated using the BGM model.

Analysing Fig. 6, it can be observed that all models exhibit clear and coherent spatial structures, indicating the presence of a well-defined spatial effect across the study region. The main differences between the maps are related to the scale and smoothness of the estimated effects rather than to their spatial configuration. The BGM model shows more pronounced spatial contrasts, with higher magnitude variations across space. In contrast, the HBART and BARTSIMP models are very similar in terms of the identified spatial pattern, differing mainly in scale, with both producing more attenuated and smoother representations of the spatial effect compared to BGM.

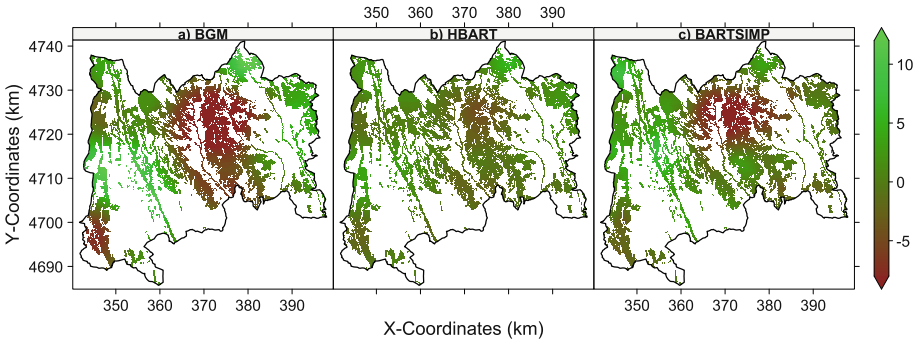


Fig. 6. Maps of spatial effect, considering the 3 models: BGM: Bayesian Geostatistical Model; BART: Bayesian Additive Regression Tree; and BARTSIMP: BART for Spatial INLA Modelling and Prediction.

4.3 Interpretation

BGM: Table 2 presents the posterior mean, standard deviation, and the corresponding Highest Posterior Density (HPD) intervals for the parameters and hyperparameters of the Bayesian Geostatistical Model. From Table 2, one observes that only covariates Aspect, GCI, and the two categories of the Forest Type variable (Broadleaf and Mixed) are statistically significant at the 5% level, as their HPD intervals do not include zero.

Table 2. Summary of the estimates of the parameters and hyperparameters of the BGM.

			HDP					HDP	
Variable	Mean	SD	Low	High	Variable	Mean	SD	Low	High
Blue	-0.00	0.04	-0.08	0.08	GCI	23.96	10.31	3.70	44.11
NDVI	15.00	30.47	-44.72	74.63	Homogeneity	4.33	25.98	-46.63	55.18
Altitude	0.07	0.05	-0.03	0.16	Slope	0.20	0.75	–1.28	1.67
Aspect	0.11	0.04	0.03	0.19	Broadleaf	–83.20	12.13	–106.97	–59.41
Mixed	-27.74	13.60	–54.40	–1.08					
Parameters of the Covariance Matrix									
r	16454.13		7949.40	25904.23	σ^2	305.28		37.81	782.33

Considering the four statistically significant covariates, the results indicate that Aspect and GCI are positively associated with AGB, whereas the Broadleaf and Mixed forest types show lower AGB values compared with Coniferous forests. Specifically, a one-unit increase in Aspect is associated with an average increase of 0.11 t/ha in AGB, while a one-unit increase in GCI corresponds to an average

increase of 23.96 t/ha, holding the remaining covariates constant. In contrast, Mixed forests present, on average, 27.74 t/ha lower AGB than Coniferous forests, whereas Broadleaf forests show an average decrease of 83.20 t/ha. Furthermore, the spatial component of the model revealed significant spatial dependence in AGB, with spatial correlation persisting up to approximately 16.5 km. The estimated marginal variance parameter also indicates considerable residual spatial variability not explained by the covariates included in the model.

BART Model: Figure 7 presents the Partial Dependence Plots (PDP) for the continuous covariates obtained from the BART model. These plots allow the evaluation of the marginal effect of each covariate on the predicted AGB values. Covariate NDVI shows a linear positive association with AGB, whereas covariates Blue and Slope exhibit a negative linear association with AGB (mild for the latter). In contrast, covariates CGI, Homogeneity, Altitude and Aspect display non-linear patterns, suggesting that their effects on AGB vary across different ranges of the covariates.

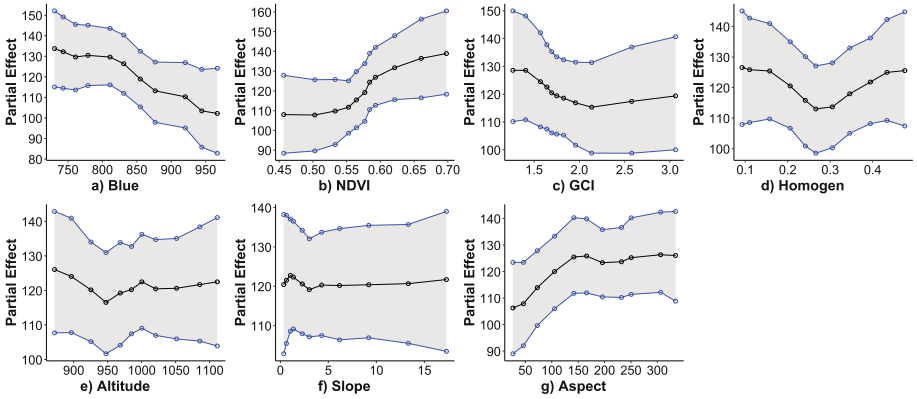


Fig. 7. Partial Dependence Plot (BART Model).

BARTSIMP Model: Figure 8 presents the PDP for the continuous covariates obtained from the BARTSIMP model. By inspecting the figure, it can be seen that the PDP are identical to those obtained from the BART model (Fig. 7), with the same interpretation.

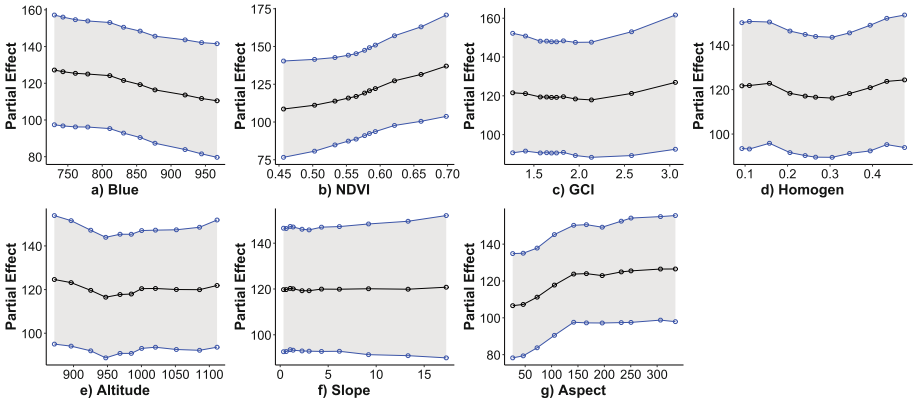


Fig. 8. Partial Dependence Plot (BARTSIMP Model).

5 Discussion and Conclusions

The findings of this study demonstrate discrepancies between the performances of the Bayesian ML models and the geostatistical model, particularly regarding predictive accuracy and uncertainty quantification. The BGM model demonstrated the best overall performance in terms of point prediction accuracy. In addition, BGM achieved the highest NSE value, indicating a superior ability to explain the variability of AGB compared to all other models considered.

In contrast, the models based on BART (BART, HBART, and BARTSIMP) produced noticeably higher RMSE and MAE values, indicating lower accuracy in point predictions. Among them, HBART demonstrated slightly superior performance, followed closely by BART and BARTSIMP. Even so, none of the models based on BART outperformed the BGM in any of the predictive accuracy metrics.

With respect to uncertainty quantification, the results were more nuanced. The HBART model demonstrated the highest coverage probability, approaching the nominal level and indicating well-calibrated prediction intervals. The BGM and BART models also provided reasonably adequate coverage, albeit still below the nominal level, suggesting a mild underestimation of uncertainty. In contrast, the BARTSIMP model demonstrated a markedly lower coverage, clearly indicating poorly calibrated predictive intervals and a systematic underestimation of uncertainty. The superior performance of the HBART model with respect to this metric may be related to its hybrid formulation, which combines the flexibility of BART with an explicit treatment of spatial structure in the residuals. This dual representation of uncertainty appears to improve interval calibration. However, this gain was accompanied by a decline in predictive accuracy, highlighting a trade-off between accurate point predictions and reliable uncertainty quantification.

Overall, these findings are partially aligned with those reported by Jiang et al. (2025) [7]. While the superior performance of the geostatistical model in terms of RMSE and MAE is consistent with their results, the advantage of the models based on BART in uncertainty quantification is not fully supported in this study. In the present study, only HBART demonstrated improved coverage, whereas BART and BARTSIMP failed to provide well-calibrated uncertainty estimates. These discrepancies may be attributed to dataset-specific characteristics, particularly the relatively small sample size and the extent to which the available covariates already explain spatial variability. In such circumstances, the advantages offered by highly flexible ML models may be restricted, while explicitly spatial approaches maintain greater robustness and efficiency.

Analysis of the predictive maps showed that all models exhibited analogous spatial patterns of AGB, with higher values concentrated in the northeastern and central regions of the study area. However, the analysis of spatial effects revealed some differences across models. The BGM model exhibited more pronounced spatial contrasts, whereas the HBART and BARTSIMP models were very similar in terms of the underlying spatial pattern, differing mainly in scale, with both producing smoother and more attenuated spatial effects. This finding indicates that the spatial component is more pronounced in the BGM model, while the models based on BART yield a more regularised representation. The interpolation of BART residuals using the BGM model further indicates that the BART model does not explicitly account for spatial dependence, thereby reinforcing the relevance of the hybrid approach to adequately represent this structure.

In terms of computational time, the models based on BART were substantially more demanding than the BGM model. The BGM model completed computations in approximately 15 min, whereas BART and HBART required around 30 min, and BARTSIMP approximately two days, which may represent a practical limitation in large datasets or when computational efficiency is a key concern.

The interpretation of covariate effects differs between the approaches considered. The BGM model identified only a few significant covariates (Aspect, GCI, and Forest Type), whereas the models based on BART exhibited more complex, non-linear relationships. Despite these, both approaches consistently identified Aspect and GCI as the most relevant variables, with a positive effect on AGB.

The results show that no single model dominates across all evaluation criteria. The BGM model demonstrated the highest predictive accuracy, while the HBART model provided the most reliable uncertainty quantification. The findings of this study indicate that hybrid approaches hold considerable promise, as they integrate the strengths of ML and geostatistical modelling. Therefore, the choice of model should be guided by the primary objective of the analysis, whether prioritising point prediction accuracy or the quality of uncertainty estimates.

Acknowledgments. This work is funded by national funds through the FCT – Fundação para a Ciência e a Tecnologia, I.P., under the scope of the projects UID/00297/2025 (https://doi.org/10.54499/UID/00297/2025) and

UID/PRR/00297/2025 (https://doi.org/10.54499/UID/PRR/00297/2025) (Center for Mathematics and Applications). National funding by FCT - Fundação para a Ciência e Tecnologia, I.P., through the individual research grant 2023.01166.BDANA (https://doi.org/10.54499/2023.01166.BDANA) of Ricardo Coelho.

References

1. Bakka, H.: What is needed for a good mesh? (2017). https://haakonbakkagit.github.io/btopic104.html#31_What_is_needed_for_a_good_mesh, Accessed 7 January 2026
2. Chipman, H.A., George, E.I., McCulloch, R.E.: BART: Bayesian Additive Regression Trees. Annals Appli. Stat. **4**(1), 266–298 (2010). https://doi.org/10.1214/09-AOAS285
3. Coelho, R., Natário, I., Fraile, S.: Bayesian geostatistical models for predicting above-ground biomass from remote sensing data. Remote Sensing Appli. Soc. Environ. **41**, 101855 (2026). https://doi.org/10.1016/j.rsase.2025.101855
4. Diggle, P.J., Giorgi, E.: Model-based geostatistics for global public health: methods and applications. Chapman & Hall/CRC (2019)
5. Diggle, P., Ribeiro, P.: Model-based Geostatistics. Springer Series in Statistics. Springer (Mar 2007)
6. Du, H., et al.: Spatial heterogeneity and carbon contribution of aboveground biomass of moso bamboo by using geostatistical theory. Plant Ecol. **207**, 131–139 (2010). https://doi.org/10.1007/s11258-009-9659-3
7. Jiang, A.Z., Wakefield, J.: BARTSIMP: flexible spatial covariate modeling and prediction using Bayesian Additive Regression Trees. Spatial Spatio-Temporal Epidemiol. **55**, 100757 (2025). https://doi.org/10.1016/j.sste.2025.100757
8. Kapelner, A., Bleich, J.: bartMachine: machine learning with bayesian additive regression Trees. J. Stat. Softw. **70**(4), 1–40 (2016). https://doi.org/10.18637/jss.v070.i04
9. Krainski, E., Gómez-Rubio, V., et al.: Advanced Spatial Modeling with Stochastic Partial Differential Equations Using R and INLA. Chapman and Hall/CRC (Dec 2018). https://doi.org/10.1201/9780429031892
10. Li, Y., Li, M., Li, C., Liu, Z.: Forest aboveground biomass estimation using Landsat 8 and Sentinel-1A data with machine learning algorithms. Sci. Rep. **10**(1), 9952 (2020). https://doi.org/10.1038/s41598-020-67024-3
11. Likas, A., Vlassis, N., J. Verbeek, J.: The global k-means clustering algorithm. Pattern Recog. **36**(2), 451–461 (2003). https://doi.org/10.1016/S0031-3203(02)00060-2, biometrics
12. Moraga, P., Dean, C., Inoue, J., Morawiecki, P., Noureen, S.R., Wang, F.: Bayesian spatial modelling of geostatistical data using INLA and SPDE methods: a case study predicting malaria risk in Mozambique. Spatial Spatio-Temporal Epidemiol. **39**, 100440 (2021). https://doi.org/10.1016/j.sste.2021.100440
13. National Geographic Institute (IGN): QSM - CNIG: National center for geographic information (2025). https://www.ign.es/web/qsm-cnig Accessed 12 December 2025
14. Roberts, D.R., et al.: Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. Ecography **40**(8), 913–929 (2017). https://doi.org/10.1111/ecog.02881

15. Rue, H., Martino, S., Chopin, N.: Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. J. Royal Stat. Soc. Ser. B (Stat. Methodol.) **71**(2), 319–392 (2009). https://doi.org/10.1111/j.1467-9868.2008.00700.x
16. Tan, Y.V., Roy, J.: Bayesian additive regression trees and the General BART model. Stat. Med. **38**(25), 5048–5069 (2019)
17. Turton, A.E., Augustin, N.H., Mitchard, E.T.A.: Improving estimates and change detection of forest above-ground biomass using statistical methods. Remote Sensing **14**(19) (2022). https://doi.org/10.3390/rs14194911
18. Vashum, K.T., Jayakumar, S.: Methods to estimate above-ground biomass and carbon stock in natural forests-a review. J. Ecosyst. Ecography **2**(4), 1–7 (2012)
19. Wang, J., et al.: Remote sensing estimation of bamboo forest aboveground biomass based on geographically weighted regression. Remote Sensing **13**(15) (2021). https://doi.org/10.3390/rs13152962
20. Yankey, O., et al.: Disaggregating census data for population mapping using a Bayesian Additive Regression Tree model. Appl. Geogr. **172**, 103416 (2024). https://doi.org/10.1016/j.apgeog.2024.103416
21. Zhang, L., et al.: Mapping topsoil total nitrogen using random forest and modified regression kriging in agricultural areas of central China. Plants **12**(7) (2023). https://doi.org/10.3390/plants12071464
22. Zhang, T., Geng, G., Liu, Y., Chang, H.H.: Application of bayesian additive regression trees for estimating daily concentrations of PM2.5 components. Atmosphere **11**(11) (2020). https://doi.org/10.3390/atmos11111233

Author Index

A
Abuselidze, George 644
Aguiar, Camile Maria Cunha 544
Alihodžić, Adis 130
Almeida, Edson L. 613
Almeida, Mathews Edwirds Gomes 561
Almeida, Rute 340
Alves, Maria João 179
Araújo, Thiago 357

B
Bairrada, Cristela 179
Bello-Orgaz, Gema 492
Bernatskyi, Andrii 393
Bez, Jean Luca 357
Bieliakov, Robert 393
Bietolini, Francesco 236
Boeres, Maria Claudia Silva 269
Boldt, Jessica 478
Bonesso, Gustavo Adolpho 67
Brandão, Diego 613
Brun, André Luiz 321
Bunyak, Filiz 478

C
Casaca, Wallace 510
Ceddia, Marcos Bacis 613
Cherif, Yousra 628
Coelho, Regina Célia 67
Coelho, Ricardo 660

D
da Silva Franco, Roberto Yuri 544
da Silva Maciel, Luiz Maurílio 529, 561
da Silva Maciel, Luiz Maurilio 580
da Silva Thiago, Vinícius 287
da Silva, Gabriel Rezende 561
Davydenko, Nadiia 644
de Abreu, Alexandre Augusto Alberto Moreira 161
de Araújo Bhering Bittencourt, Pedro 529, 561
de Carvalho, Bruno Campos 561
de Castro Martins Ferreira Nogueira, João Vítor 561
de Godoy, Carlos Marcelo Gurjão 67
de Moraes, Renato Elias Nunes 269
de Oliveira, Sanderson L. Gonzaga 161
de Sousa, Thiago Augusto Soares 544
de Souza Dias, Gabriela Moutinho 287
de Souza Marins, Thais 561
Dias, Luís G. 204
Djamaa, Badis 221
Dmitriev, Maxim 599
dos Santos, Joel André Ferreira 613
Duarte, Caio Seixas 580
Duarte, Francisco J. 340
Duarte, Julio Cesar 287
Durão, Jonas Lucas 204

F
Ferreira, Ana Carolina 613
Ferreira, Angela 613
Ferretti, Eliana Chaves 67
Fesenko, Oleksii 393
Fong, Alvis 110
Fraile, Silvia 660
Frederico, Nathalia 461
Freytag, Gabriel 357

G
Gamazo-Real, Jose-Carlos 445
Gervasi, Osvaldo 236
Gowda, Likhith 428
Gudz, Olena 644
Guerra, Luan B. 510

H
Hasanspahić, Damir 130
Hasić, Damir 130
Ho, Phat Hoang 413

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 679–681, 2026.
https://doi.org/10.1007/978-3-032-30491-9

I
Ibersiene, Lyes 221
Ibraheem, Abdulrahman 18
Idri, Ali 628

J
Jigalur, Pranav 428

K
Khalid, Omer 445
Kiani, Narsis 18
Kim, Young Seo 599
Kipli, Kuryati 375
Koltovskov, Dmytro 393
Koppal, Praveen 428
Künas, Cristiano 303, 357

L
Laos, Kazim 492
Le, Minh Tu 3
Le, Trung-Dung 251
Lit, Asrani 375
Louzada, Nilo Marchiori 269
Lytaev, Mikhail S. 51, 144

M
Machado, Rodrigo 303
Maia, Brian Luís Coimbra 561
Marinello, Marco 192
Martins, Daniel 179
McIntosh, Ian Alexander Zahner 613
Meiguins, Bianchi Serique 544
Menendez, Hector D. 492
Merabet, Ahmed Yadine 221
Miranda, Rafael 461
Mishra, Rishav 110
Monteiro, Walbert Cunha 544
Morais, Jefferson 461
Morales, Bruno 303
Mota, Vinícius F. S. 269

N
Natário, Isabel 660
Navaux, Philippe O. A. 303
Navaux, Philippe 357
Nguyen, An Hoang 413
Nguyen, Van Duan 3
Nguyen, Van Nghi 3

O
Onyshko, Svitlana 644

P
Padilha, Mariana 35
Peng, Song 96
Pereira, Cátia 340
Perri, Damiano 236
Petry, Tailí 35
Protasov, Maxim 599

Q
Qiang, Wang 96
Quiles, Marcos G. 204

R
R Pai, Sindhu 428
Radzivilov, Hryhorii 393
Rezende, Daniel Muller 561
Ribas, Lucas C. 510
Ribeiro, Rui C. 340
Rigon, Pedro 303
Rodrigues, Ivo A. 340
Rodrigues, Luiz Antonio 321
Roibás-Millán, Elena 445
Rong, Yang 96
Rufino, José 35

S
S Rai, Kripa 428
Sabitov, Denis 599
Salles, Charles Giovane 321
Salles, Ronaldo Moreira 287
Santana, Lucas Silva 561
Sapawi, Rohana 375
Schepke, Claudio 35
Senouci, Mustapha Reda 221
Sharma, Aninda Kumar 79
Shih, HongYi 478
Shrestha, Bade 110
Silva, Leopoldo O. 340
Silva, Saulo 461
Soares, Jorge 613
Suhaili, Shamsiah 375

T
Tegner, Jesper 18
Tran, Hanh Thi 3
Trang, Kien 413

V
Verniz, Debora 478
Vieira, Marcelo Bernardes 529, 561, 580
Villaça, Rodolfo S. 269
Villela, Saulo Moraes 529, 561, 580
Vu, Minh Thang 3
Vuong, Bao Quoc 413

W
Wan Masra, Sharifah Masniah 375

Z
Zahali, Oussama 221
Zerati, Ana Beatriz S. 510