Deep Learning-Based Acoustic Event Detection in IoT Edge Devices

Daniel Quirumbay-Yagual(✉), Luis Benalcazar-Yagual, and José Miguel Sánchez-Aquino

Universidad Estatal Peninsula de Santa Elena, Santa Elena, Ecuador
{dquirumbay,luisbenalcazaryagual5499,jsanchez}@upse.edu.ec

Abstract. Acoustic event monitoring in high-variability scenes requires edge-aware intelligence that preserves privacy while operating under memory and compute constraints. We present an integrated IoT-to-edge workflow that combines ESP32-C6 acquisition, timestamped local storage, and multi-stage data curation to train acoustic classifiers from field recordings. We benchmark CNN, TCNN, GRU, LSTM, and attention-based CRNN models on 16 kHz mel-spectrograms, reaching up to 95.6% Macro-F1 and 95.3% accuracy. Results show that CRNN/LSTM maximize discrimination, whereas TCNN provides stronger probability reliability. This study is a laboratory validation setting (TRL 2–3: pre-deployment): no real-time inference has yet been executed on the ESP32-C6 platform. The study focuses on reproducible edge-aware offline experimental evaluation. These findings support informed pre-deployment model selection in real-world IoT scenarios, where reliability and resource efficiency must be jointly optimized.

Keywords: Internet of Things · Deep Learning · Edge AI · Acoustic Event Detection · Smart Environments

Nomenclature

Acronym	Definition
CNN	Convolutional Neural Network
CRNN	Convolutional Recurrent Neural Network
GRU	Gated Recurrent Unit
LSTM	Long Short-Term Memory
TCNN	Temporal Convolutional Neural Network
IoT	Internet of Things
ESP32	Espressif Systems 32-bit Microcontroller

1 Introduction

IoT acoustic sensing systems must characterize speech, impacts, alerts, and background noise under high scene variability, intermittent connectivity, and limited edge resources, ideally without continuous raw-audio offloading. Prior work

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026 Workshops, LNCS 16762, pp. 177–194, 2027.
https://doi.org/10.1007/978-3-032-30524-4_12

shows that recording conditions strongly affect generalization [25,30], while IoT and edge-computing studies emphasize local processing for latency and resilience [2,3,34,35]. Communication-centric MEC analyses further detail orchestration constraints in heterogeneous networks. In practice, these constraints define a multi-objective design space where accuracy, memory footprint, latency, and reliability must be co-optimized rather than tuned independently [5,24,36,39].

Unlike prior works that evaluate models in isolation on benchmark datasets, this paper contributes a reproducible, leakage-safe, calibration-aware evaluation framework grounded in real IoT acquisition. The main contribution is an integrated IoT-to-edge workflow that bridges real-world data acquisition with calibration-aware model selection under deployment constraints.

This work applies three techniques: (1) adaptive firmware-triggered acquisition and multi-stage curation with leakage-safe partitioning; (2) a unified comparative evaluation of CNN, TCNN, GRU, LSTM, and CRNN under discrimination, calibration, and complexity criteria; and (3) a pre-deployment selection protocol based on auditable run artifacts and trade-off analysis. This study links real-world acoustic acquisition with edge-aware model selection in an end-to-end pipeline. The contribution focuses on integration and evaluation rather than new architectures. Specifically, the evaluation corresponds to technology readiness level (TRL) 2–3 (laboratory validation / proof-of-concept): experiments are offline in a laboratory validation setting rather than on embedded fielded inference. We emphasize that "edge-aware" in this study refers to constraint-informed evaluation design, not validated embedded inference on the ESP32-C6.

1.1 Edge AI and IoT Architectural Perspective

The proposed system follows a layered Edge AI architecture where each layer solves a distinct function under resource constraints: (i) *sensing layer* for event-triggered acquisition and local persistence, (ii) *edge analytics layer* for lightweight filtering and feature-oriented preprocessing, and (iii) *modeling layer* for offline training and periodic model refresh. This decomposition is consistent with current edge-intelligence design recommendations [34,35] and broader convergence analyses between deep learning and edge systems [14].

In this framework, we view edge operation as orchestrating heterogeneous compute roles rather than running the largest model on the smallest device. The study therefore focuses on integration and evaluation: we compare model families under an edge-aware feasibility protocol using a real acquisition-to-training pipeline. Full on-device inference on the ESP32-C6 remains future work; the present results provide offline, laboratory validation.

We hypothesize that CRNNs yield better discrimination in constrained edge settings because they capture local transients (convolutions) and temporal context (recurrent layers) while keeping a modest footprint.

2 Related Work

Recent acoustic-event literature shows three stable findings relevant to this paper: (i) spectrogram-based deep models consistently outperform handcrafted features in multi-class sound recognition [28,31]; (ii) temporal modeling (CRNN

and recurrent variants) improves robustness in non-stationary scenes [1,6]; and (iii) edge deployment requires joint analysis of discrimination, calibration, and model size rather than accuracy alone [5,34–36]. Our study focuses on this integrated perspective with field-recorded data and leakage-safe evaluation.

From a methodological standpoint, two additional technical aspects remain underemphasized in many prior pipelines: domain-shift sensitivity and confidence quality. Cross-dataset evidence indicates that performance can degrade notably when recording conditions, microphones, or scene priors differ from training assumptions [9,25,30]. In parallel, calibration studies show that modern deep classifiers may produce overconfident probabilities even when top-1 accuracy is high, which directly impacts threshold-based alerting in IoT operation [13,26,29,40]. These findings motivate our use of leakage-safe splits and explicit calibration reporting as first-class evaluation criteria rather than secondary diagnostics.

Recent progress in audio representation learning has shifted toward large-scale pretraining and transferability. AudioSet-scale supervision and pretrained audio networks improve robustness under dataset shift [10,16,23], while transformer-based models report strong cross-benchmark performance in environmental sound analysis [7,11,12,21]. In self-supervised settings, CPC and wav2vec-family approaches support low-label adaptation in acoustic tasks [4,17,33,41]. Cross-domain architectural transfer has also influenced audio modeling practice: residual and channel-attention mechanisms improve feature reuse [15,20,22], mobile-oriented designs improve efficiency under constraints [8,18,19,32,37,38].

However, despite advances in large-scale pretraining, these approaches are rarely evaluated within fully integrated IoT-to-edge acquisition pipelines under real-world constraints. This motivates the methodological design presented in Sect. 3.

Table 1 summarizes representative prior studies in comparison with our approach. As observed, previous works focus primarily on controlled benchmark datasets such as ESC-50 and UrbanSound8K [30], whereas our study extends evaluation to real-world field-acquired data under IoT constraints. Additionally, prior works do not explicitly address edge-awareness and calibration reliability jointly, which are central to our methodology.

2.1 Model Families Considered in This Study

The model families compared in this paper cover complementary inductive biases: local spectral pattern extraction (CNN), recurrent temporal memory (GRU/LSTM), hybrid spectro-temporal modeling (CRNN), and temporal convolution dynamics (TCNN).

3 Methodology

This section describes IoT acquisition, dataset assembly and partitioning, feature extraction, model implementations, and evaluation protocol. We emphasize reproducibility and auditable experimental artifacts.

Table 1. Representative comparison between prior literature and this study.

Work	Dataset/Task	Model Family	Reported Result	Edge
Piczak (2015) [28]	ESC-50, environmental sound classification	CNN	64.5% accuracy (reported)	No
Salamon and Bello (2017) [31]	UrbanSound8K, augmented ESC setting	CNN + augmentation	73.7% accuracy on US8K (reported)	No
This work	Field AED with mixed real events	CNN, TCNN, GRU, LSTM, CRNN	95.6% Macro-F1 (best model) with calibration analysis	Partial

3.1 IoT Edge Acquisition and Adaptive Signal Processing

Each sensing node uses ESP32-C6 DevKitC v1.4, INMP441 digital microphone, NEO-6M GPS synchronization, and SPI microSD logging. Audio is captured at 32 kHz and stored as 16-bit PCM WAV. Figure 1 shows the wiring architecture used in field collection.

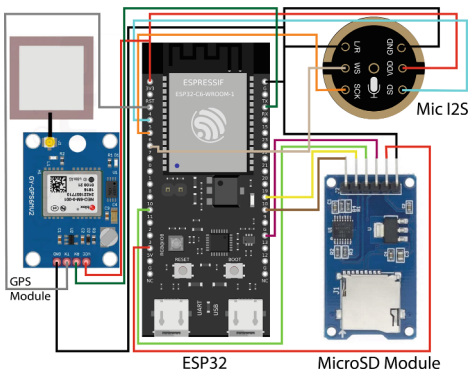

Fig. 1. Hardware architecture of the ESP32-C6 acoustic sensing node integrating I^2S microphone, SPI storage, and GPS synchronization. *Key takeaway: Autonomous edge acquisition with local persistence enables offline feature extraction and model training without raw-audio offloading.*

Triggering is based on adaptive short-term energy:

$$\tilde{x}[n] = \frac{x[n]}{2^{15}}, \quad \mathrm{RMS}_k = \sqrt{\frac{1}{N}\sum_{n=1}^{N} \tilde{x}_k[n]^2}, \quad L_k = 20\log_{10}(\mathrm{RMS}_k + \varepsilon) \tag{1}$$

$$B_k = \alpha B_{k-1} + (1-\alpha)L_k, \quad \Delta_k = L_k - B_k \tag{2}$$

with $\alpha = 0.92$, threshold $\theta = 3.5$ dB, pre-buffer $T_p = 2$ s and recording window $T_r = 6$ s.

Equation (1) computes RMS energy and converts it to a logarithmic level L_k for robust amplitude tracking. Equation (2) constructs an exponential moving

baseline B_k, so $\Delta_k = L_k - B_k$ measures deviation from background noise. The trigger fires on positive deviations to detect transient events while suppressing steady noise. This simple, streaming-friendly method avoids spectral transforms and reduces unnecessary transmission and storage in constrained IoT nodes.

Algorithm 1. Adaptive Edge Acoustic Triggering

```
Acquire audio block and compute L_k, B_k, and Δ_k
if Δ_k > θ and cooldown is satisfied then
    Append pre-buffer + current segment to WAV file
    Save timestamp and event metadata in event_log.csv
end if
```

3.2 Dataset Assembly and Partitioning

The curated corpus contains 6,400 clips across seven classes: car_horn, children, dog, ball, siren, ambient, and people. Source material combines proprietary field recordings with selected UrbanSound8K samples [30]. We applied balancing and targeted augmentation and used the high-frequency ratio R_{hf} to preserve child-vocalization spectral content. Specifically, targeted augmentation was applied to the real field recordings collected in educational environments to improve representation for classes that are frequent in school settings; augmentation procedures follow common practices in environmental-sound augmentation [31]. We note that several classes required additional augmented examples due to scarcity of natural samples, and that most proprietary recordings originate from a single institutional context, which constrains geographic and demographic generalization and should be considered when interpreting results [9,25].

Inclusion criteria: (i) event-labelled clips with unambiguous annotations; (ii) intact audio files without clipping or corruption; (iii) segments that pass the VAD filter (see Sect. 5.4) and with a minimum event duration after VAD of 0.5 s; and (iv) retention of spectral content indicated by R_{hf}. Exclusion criteria: corrupted files, clips with ambiguous or conflicting labels, recordings dominated by silence or non-informative noise, and event types judged irrelevant to the target educational context. These curation rules follow common dataset-assembly best practices [25,30] and are intended to reduce label noise and dataset leakage. We selected classes that are commonly present in educational environments due to their high prevalence and occurrence frequency; additional samples from UrbanSound8K were added when appropriate, while classes outside the scope of the educational/use-case context were discarded to keep the taxonomy task-relevant.

Table 2 provides descriptive statistics of the curated dataset, including class distribution, median duration, and signal intensity. These values highlight relevant variability across classes, particularly in duration and RMS levels, which motivates robust spectro-temporal feature extraction and class-aware training (Fig. 2).

Table 2. Descriptive statistics by class for the curated final dataset.

Class	Files	Median Duration [s]	Median RMS [dBFS]
car_horn	800	2.00	−23.3
childrens	1000	4.00	−31.5
bark_dog	1000	4.00	−25.5
kick_ball	1000	1.00	−26.0
siren	1000	4.00	−27.2
background_ambient	1000	4.00	−25.4
people	600	2.00	−36.2

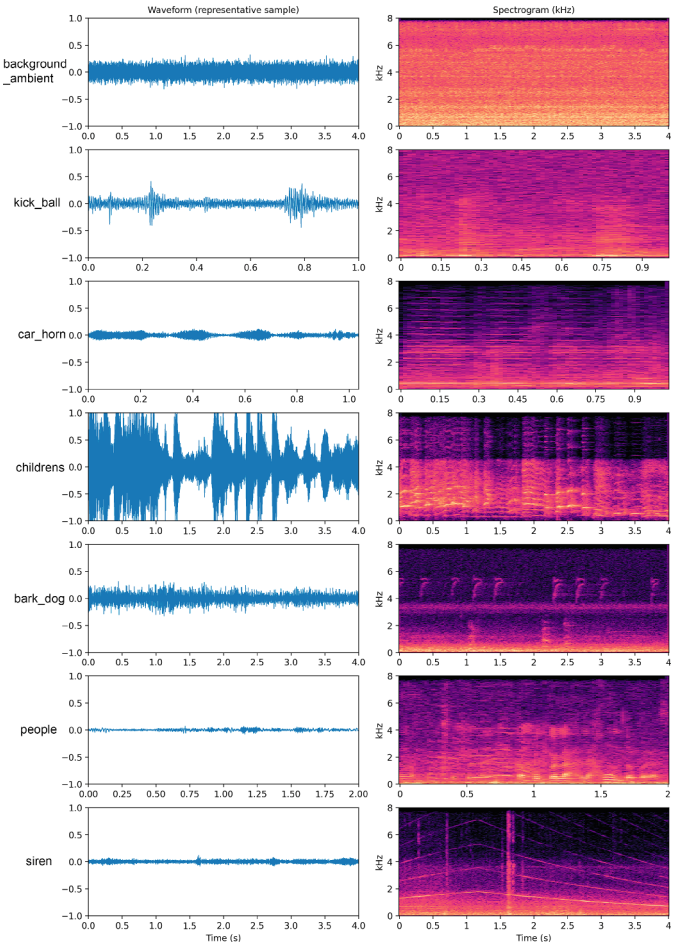

Fig. 2. Representative waveform and spectrogram examples for each target class in the curated dataset. *Key takeaway: Distinct spectro-temporal signatures across classes (e.g., ball transients vs. siren sweeps) motivate hybrid temporal modeling with attention mechanisms.*

The panels show qualitative differences used by downstream classifiers: ambient scenes present low-amplitude, broadband energy; *ball* events appear as brief, high-amplitude transients concentrated at higher frequencies; *car_horn* exhibits sustained tonal energy at lower frequencies; *children* vocalizations display longer-duration harmonic structure with variable pitch and energy; *dog* events present mid-frequency impulsive signatures; *people* clips are lower-energy speech-like segments; and *siren* shows strong narrowband sweeps. These distinctions motivate spectro-temporal representations and inform why hybrid models (CRNN, LSTM) can improve discrimination. If the figure prints in grayscale, note the relative time–frequency energy patterns (vertical intensity) and transient duration (horizontal extent) as the main visual cues.

Leakage-Safe Group Split in Training Workflow. To prevent contamination, splits are generated by group keys (`source_clip_id`, `device_id`, `session`) instead of individual clips. Groups are partitioned into train/validation/test = 70/15/15 with fixed seed, and augmentation is applied only to the training subset.

3.3 Deep Learning Stack

Each model (CNN, TCNN, GRU, LSTM, CRNN) is trained with the same protocol: 128-bin mel features, Adam optimizer (1×10^{-3}), class weighting, early stopping, and held-out test evaluation. Reported outputs include discrimination metrics (Macro-F1, Weighted-F1, Macro-Precision, Macro-Recall), ranking metrics (ROC-AUC, PR-AUC), and calibration error (ECE), together with model size indicators.

For each model family we follow a short, repeatable pipeline: load balanced manifest, map class indices, extract unified multi-channel features (mel, cepstral, temporal derivatives), split groups into train/validation/test as defined in Sect. 3.2, convert to batched tensors, and train with Adam (lr 1×10^{-3}), categorical cross-entropy, class weighting, and early stopping. After training, the notebook evaluates the held-out test set, records discrimination and calibration metrics, exports artifacts, and saves the final checkpoint.

Algorithm 2. Leakage-Safe Group Train/Validation/Test Split (Implemented Protocol)

1: **Input:** Manifest D with $(filepath, classID, source_clip_id, device_id, session)$
2: **Input:** Random seed $s = 42$
3: **Output:** Disjoint sets $(D_{train}, D_{val}, D_{test})$
4: Build group key $g = (source_clip_id, device_id, session)$
5: Aggregate one representative class label per group for stratification
6: Split groups into train/temp (70/30) with fixed seed s
7: Split temp groups into val/test (50/50) with fixed seed s
8: Assign each sample to the split of its group key g
9: Verify: no group appears in more than one subset

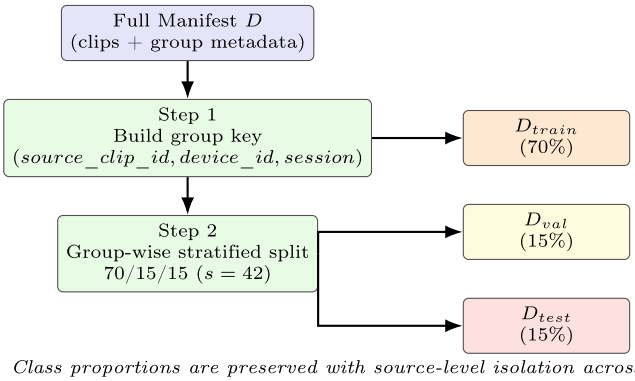

Fig. 3. Illustrative scheme of the implemented leakage-safe split policy: group-wise class-stratified partitioning (70/15/15). *Key takeaway: Group-level partitioning with source isolation prevents data contamination and ensures reproducible validation metrics under realistic deployment constraints.*

Under this protocol, class balance is preserved across subsets and experiments remain reproducible through fixed random seed while grouping by source context is enforced, which mitigates leakage risk from highly related clips when interpreting validation and test metrics (Fig. 3).

To address leakage-related concerns explicitly, all experimental results reported in Table 5 and Figs. 4, 5, 6 and 7 were obtained exclusively using the proposed split protocol (Sect. 3.2). No clip-level random splits were used.

Model Construction by Notebook (Layer Definition). Before presenting the common training flow, each notebook defines a specific network topology while preserving a shared design principle: a compact front-end for robust spectro-temporal representation and a softmax output layer for multi-class inference.

- **CNN baseline:** uses an EfficientNetB0 backbone without classification head (global average pooling output), followed by a dense projection block and final softmax layer.
- **TCNN:** stacks temporal convolutional blocks with convolution, batch normalization, ReLU activation, max-pooling, and dropout; features are aggregated with global average pooling and mapped through dense layers to softmax outputs.
- **GRU-attention:** applies a convolutional front-end (three Conv-BN-ReLU-pooling stages), reshapes features into temporal sequences, processes them with bidirectional GRU units, and applies an attention layer before dense classification.
- **LSTM-attention:** follows the same convolutional front-end and sequence reshaping strategy, replacing GRU with bidirectional LSTM recurrent units and attention-guided pooling before the softmax classifier.

– **CRNN-attention:** combines convolutional feature extraction, bidirectional recurrent encoding, and an attention-based aggregation head, ending in dense and softmax layers for final class probabilities.

Notebook-Based Training Workflow. During training we applied on-the-fly time–frequency augmentations inspired by SpecAugment to increase variability and robustness in model optimization [27].

Algorithm 3. Unified Training Pipeline per Model Notebook

Input: Balanced audio manifest and model family $m \in$ {CNN, TCNN, GRU, LSTM, CRNN}
Output: Trained model and exported metrics artifacts
Load balanced metadata and remap class indices
Extract fixed-size acoustic feature tensors
Build stratified train/validation/test splits
Instantiate architecture corresponding to m
Compile with Adam optimizer and categorical loss
Train with class weighting and early stopping
Evaluate on test split and compute summary metrics
Export metrics for table generation
Save final model weights and experiment artifacts

4 Experimental Results

This section reports empirical results from the offline leakage-safe protocol across five deep learning architectures. We present ablation analysis of preprocessing and augmentation contributions, domain generalization tests to characterize dataset-shift sensitivity, and unified comparative evaluation highlighting the discrimination-calibration-efficiency trade-off space. All experiments use the same train-validation-test partitioning with independent acquisition sets to exclude data leakage.

4.1 Ablation Study

To assess the contribution of each preprocessing and augmentation stage described in Sect. 3, an ablation study was performed using the best-performing architecture (CRNN). Table 3 shows that removing Voice Activity Detection (VAD) significantly reduces performance, highlighting the importance of excluding silent segments for transient sound events. Likewise, data augmentation and the proposed high-frequency ratio (R_{hf}) make a measurable contribution to preserving high Macro-F1 scores.

These ablation results indicate that pipeline design choices are not cosmetic preprocessing steps but primary determinants of downstream discrimination performance.

Table 3. Impact of pipeline components on classification performance: ablation study.

Configuration (CRNN)	Macro-F1
Full Pipeline (Baseline)	0.952
- w/o Augmentation	0.914
- w/o VAD Filtering	0.887
- w/o R_{hf} Modulation	0.931

4.2 Domain Generalization

A critical requirement for edge-aware evaluation in audio classification is robustness against environmental shifts. Table 4 provides explicit domain generalization results, demonstrating how a model trained exclusively on proprietary data degrades when exposed to the UrbanSound8K domain [30]. Conversely, the mixed training regime recovers performance significantly.

Table 4. Domain generalization performance exhibiting model robustness across source datasets.

Training Source	Evaluation Target	Macro-F1
Mixed (UrbanSound + Proprietary)	Balanced Test Set	0.952
Proprietary Only	UrbanSound8K Only	0.814
Mixed	UrbanSound8K Only	0.925
Mixed	Proprietary Only	0.961

4.3 Model Comparison

Models are evaluated on a held-out test set produced by the defined split protocol in Sect. 3.2, representing all seven classes without source-group contamination across subsets. To avoid metric inflation and improve methodological clarity, we restrict reporting to standard multi-class metrics: Macro-F1, Weighted-F1, Macro-Precision, Macro-Recall, classwise ROC/PR behavior, and calibration diagnostics (ECE with calibration-curve analysis). While full on-device deployment is left for future work, models are instead compared through verifiable complexity indicators: parameter count, serialized model size (MB), and profiling artifacts (CPU latency/memory).

This constitutes a broad comparative evaluation across five model families under a unified protocol.

To strengthen reproducibility, we implement: (i) an explicit split manifest keyed by `source_clip_id`, `device_id`, and `session`; (ii) domain-generalization test manifests with UrbanSound-only and proprietary-only partitions when

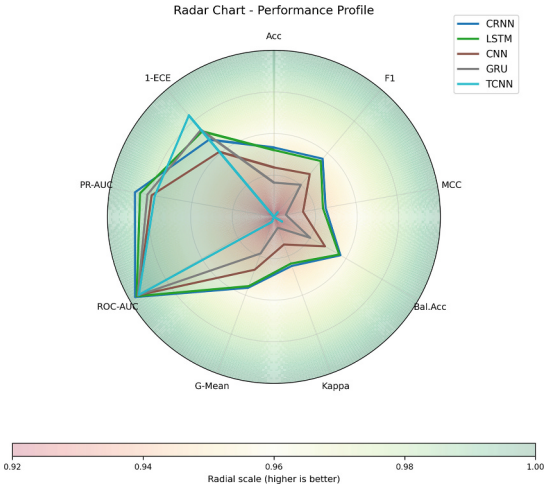

Fig. 4. Radar performance profile for the five models. Calibration is expressed as $1 - \text{ECE}$ (higher is better), and efficiency is represented by FP32 model size (MB). *Key takeaway: CRNN/LSTM dominate discrimination axes; TCNN excels in calibration efficiency.*

origin metadata is available; (iii) a basic edge-validation protocol for CPU latency/memory profiling; and (iv) an ablation pipeline (without-VAD, without-R_{hf}, without-augmentation) driven by run artifacts.

Table 5 presents the consolidated quantitative comparison across the five architectures under the same leakage-safe evaluation protocol.

Table 5. Comparison of standard multi-class metrics across all evaluated models.

Model	Acc	Macro-F1	Weighted-F1	Macro-P	Macro-R	ROC-AUC(m)	PR-AUC(m)	ECE
CRNN	0.9531	0.9563	0.9532	0.9559	0.9569	0.9971	0.9879	0.03150
LSTM	0.9521	0.9550	0.9521	0.9542	0.9562	0.9962	0.9854	0.02672
CNN	0.9437	0.9468	0.9434	0.9458	0.9483	0.9953	0.9799	0.04241
GRU	0.9365	0.9401	0.9364	0.9413	0.9402	0.9955	0.9818	0.02552
TCNN	0.9187	0.9228	0.9181	0.9232	0.9245	0.9950	0.9782	0.01885

Interpretation of Table 5. Macro-F1 and Weighted-F1 quantify discrimination; ROC-AUC(m) and PR-AUC(m) capture ranking quality; ECE reflects probability reliability (lower is better). Two regimes emerge: CRNN/LSTM for peak discrimination, TCNN for calibration-oriented operation.

To complement Table 5 with a multi-metric perspective, Fig. 4 visualizes the models across discrimination, calibration, and efficiency axes.

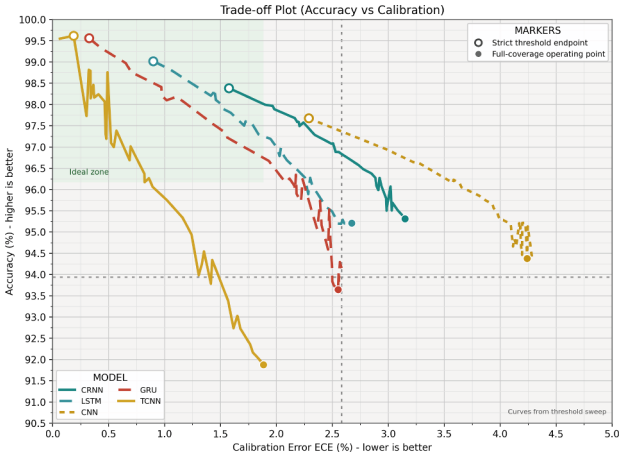

Fig. 5. Trade-off between Macro-F1 and ECE for the five models. Marker size is proportional to parameter count; distinct marker shapes and line styles were chosen for black-and-white readability. *Key takeaway: CRNN maximizes discrimination (95.6% Macro-F1); TCNN maximizes calibration reliability (ECE 0.01885).*

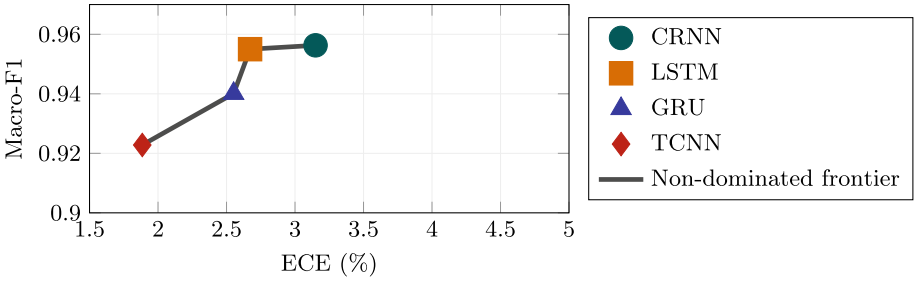

Fig. 6. Trade-off frontier between Macro-F1 and ECE. Each marker represents one model, and the line joins the non-dominated solutions in order of decreasing Macro-F1. *Key takeaway: Only CRNN, LSTM, GRU, and TCNN are non-dominated candidates; selection depends on application risk tolerance parameter α.*

Interpretation of Fig. 4. Each spoke represents one metric family; larger radius indicates stronger performance. CRNN and LSTM dominate discrimination axes, while TCNN excels in calibration efficiency.

To make the discrimination-calibration compromise explicit in two dimensions, Fig. 5 maps Macro-F1 against ECE with marker size proportional to model scale and highlights the trade-off frontier among the evaluated models.

Interpretation of Fig. 5. The axes expose the accuracy–calibration frontier directly. Lower calibration error trades off against peak discrimination; operating points should lie on the non-dominated frontier. For deployment, practitioners

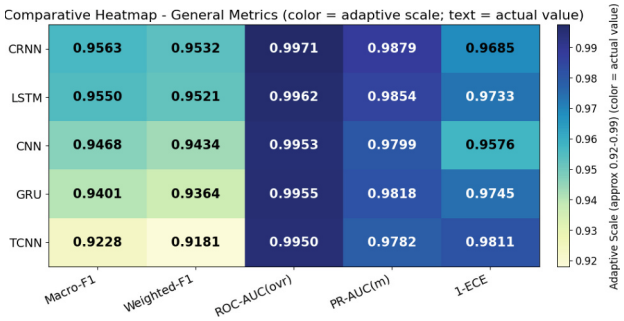

Fig. 7. Comparative heatmap across Macro-F1, Weighted-F1, ROC-AUC (macro), PR-AUC (macro), and 1 − ECE. Colors indicate min-max normalized scores, and annotations show raw values. *Key takeaway: CRNN and LSTM dominate discrimination metrics, while TCNN exhibits superior calibration—no single architecture is optimal across all objectives.*

select a utility function $U(\alpha) = \text{MacroF1} - \alpha \times \text{ECE}$ where $\alpha \in [0, 1]$ encodes risk preference: $\alpha = 0$ prioritizes discrimination, $\alpha = 1$ prioritizes calibration.

Interpretation of Fig. 7. Rows represent the five models and columns represent standard metrics. Color intensity encodes normalized quality on [0,1], while annotations preserve raw values for traceability. This joint view confirms that ranking behavior is stable across plots and tables, and that no single architecture simultaneously dominates discrimination, calibration, and size.

5 Discussion

5.1 Accuracy-Efficiency Trade-Off and Performance Analysis

The experimental results show a clear multi-objective trade-off between classification performance, calibration reliability, and computational consumption. CRNN's superior performance (95.6% Macro-F1) is attributed to its hybrid inductive bias, while LSTM offers similar discrimination with lower complexity, and TCNN provides a compressed profile with more reliable confidence (ECE 0.01885). GRU and CNN models obtained lower discrimination in our experiments and are therefore less competitive on this dataset. This is not a single-model dominance result: higher Macro-F1 typically demands more memory and may worsen calibration, while lower ECE often trades off peak discrimination.

5.2 Computational Complexity Under Edge Constraints

Computational complexity is a critical factor for edge deployment. CRNN requires the largest parameter count and memory, while TCNN and CNN offer substantially smaller footprints. Convolutional models incur cost proportional to kernel size, channel depth, and feature-map dimensions; recurrent/attention

models scale with sequence length but capture temporal context more effectively. CRNN/LSTM occupy the high-discrimination region with higher resource demands, while TCNN offers an operating point with lower resource consumption and superior calibration. Since this manuscript does not claim final embedded deployment, complexity claims remain bounded to auditable artifacts; hardware-specific throughput conclusions are deferred to future full deployment experiments.

5.3 Generalization, Domain-Shift, and VAD Limitations

The multi-stage manifesting process improves traceability and reproducibility. While the 6,400-clip dataset is simple compared to large-scale reference benchmarks, it reflects controlled, traceable acquisition under real IoT constraints, prioritizing experimental validity over scale. Domain-shift risks include: (i) *geographic specificity*—recordings collected in a single Ecuadorian institution may not represent diverse acoustic conditions, which may limit external validity in unseen environments; (ii) *voice-cohort homogeneity*—children vocalizations lack broader demographic variability; (iii) *event scarcity*—ball and siren events required additional augmentation of the real recordings, limiting naturally recorded samples. VAD filtering ($\tau_v \in [0.15, 0.55]$) improves consistency but can discard events in crowded scenes, reducing multi-speaker context representation.

5.4 Calibration and Confidence Implications

Although models achieved strong discrimination, the calibration analysis reveals that predictive confidence requires caution. CRNN's recurrent and attention-driven structure tends to generate sharper softmax distributions, leading to overconfident predictions, particularly for low-frequency classes like siren. By contrast, TCNN exhibits lower ECE (0.01885), suggesting more reliable confidence estimates suitable for downstream decision logic (threshold-based alarms, abstention mechanisms, two-stage verification pipelines) at the edge.

5.5 Comparison with Prior Work

Previous studies in acoustic event detection, such as those reported by [30,31], commonly achieve accuracy levels between 90% and 95% on the UrbanSound8K dataset under controlled experimental conditions. In our case, the best-performing CRNN model reached a Macro-F1 score of 95.6% with an overall accuracy of 95.3% on a domain-specific seven-class dataset.

While these results are competitive, direct comparison should be approached carefully due to differences in class taxonomy, dataset composition, and validation protocols. In particular, our dataset was specifically curated for the target application context, which differs from general-purpose urban sound benchmarks.

6 Conclusion and Future Work

6.1 Summary and Practical Recommendations

We present a pragmatic IoT-to-edge evaluation workflow with three outcomes: (1) a reproducible acquisition and curation pipeline with source-wise partitioning and VAD verification; (2) a leakage-safe evaluation protocol reporting discrimination, calibration, and complexity metrics; and (3) an application-oriented selection procedure based on the Pareto trade-off frontier. This positions the framework as a practical bridge between controlled model evaluation and real-world edge deployment constraints.

Key findings: CRNN produces the highest discrimination (95.6% Macro-F1) at moderate footprint; TCNN yields the lowest ECE and more reliable probabilistic outputs. To pick a deployment model, practitioners select a trade-off parameter α in $U(\alpha) = \text{MacroF1} - \alpha \times \text{ECE}$ and maximize $U(\alpha)$ over the non-dominated set (Fig. 6). Prefer discrimination-oriented models when false negatives are costlier; prefer calibration-oriented models when false alarms or human-verification costs dominate.

6.2 Limitations and Future Directions

Generalization and Transfer Learning: Future work will validate model robustness across geographically and architecturally diverse recording sites. Techniques such as domain adversarial training, source-free domain adaptation, or federated learning could mitigate geographic specificity while preserving privacy. Exploring transfer from large-scale multilingual speech datasets (if publicly available) is also warranted.

Multi-label and Overlapping Events: The current 7-class single-label formulation does not capture scenarios with simultaneous sirens and crowd noise. Extending the pipeline to multi-label or hierarchical classification (e.g., presence of 'siren' + presence of 'crowd') would improve ecological validity in real-world settings.

Feature Extraction Optimization: Optimizing mel-spectrogram extraction and intermediate representations could reduce network and storage footprint by a factor of $\sim$10 in future data-collection pipelines.

Concluding Remarks. This work demonstrates that reproducible acoustic event modeling can be achieved with low-cost IoT hardware when acquisition, curation traceability, and multi-objective model evaluation are designed as a single system.

Data Availibility Statement. Code and documentation will be released on Mendeley Data. The **IoT-AED** dataset and manifests are available at https://doi.org/10.17632/dg5k9pwv4d.1.

Conflict of Interest. The authors declare no conflict of interest.

References

1. Adavanne, S., Politis, A., Virtanen, T.: Sound event detection using spatial features and convolutional recurrent neural network. In: Proceedings of ICASSP, pp. 771–775. IEEE (2017)
2. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M.: Internet of Things: a survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tutor. **17**(4), 2347–2376 (2015)
3. Atzori, L., Iera, A., Morabito, G.: The internet of things: a survey. Comput. Netw. **54**(15), 2787–2805 (2010)
4. Baevski, A., Zhou, Y., Mohamed, A., Auli, M.: wav2vec 2.0: a framework for self-supervised learning of speech representations. Adv. Neural Inf. Process. Syst. (2020)
5. Banbury, C., Zhou, C., Fedorov, I.: MLPerf Tiny benchmark (2021). arXiv:2106.07597
6. Cakir, E., Parascandolo, G., Heittola, T., Huttunen, H., Virtanen, T.: Convolutional recurrent neural networks for polyphonic sound event detection. IEEE/ACM Trans. Audio Speech Lang. Process. **25**(6), 1291–1303 (2017)
7. Chen, S., Wu, Y., Wang, C.: BEATs: audio pre-training with acoustic tokenizers. In: Proceedings of ICML, pp. 5178–5193. PMLR (2023)
8. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: Proceedings of CVPR, pp. 1800–1807. IEEE (2017)
9. Fonseca, E., Favory, X., Pons, J., Font, F., Serra, X.: FSD50K: an open dataset of human-labeled sound events. IEEE/ACM Trans. Audio Speech Lang. Process. **30**, 829–852 (2022)
10. Gemmeke, J.F., Ellis, D.P.W., Freedman, D.: Audio set: an ontology and human-labeled dataset for audio events. In: Proceedings of ICASSP, pp. 776–780. IEEE (2017)
11. Gong, Y., Chung, Y.-A., Glass, J.: AST: audio spectrogram transformer. In: Proceedings of Interspeech, pp. 571–575 (2021)
12. Gong, Y., Chung, Y.-A., Karlinsky, L.: AudioMAE: masked autoencoders for self-supervised audio representation learning. Adv. Neural Inf. Process. Syst. (2022)
13. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Proceedings of ICML, pp. 1321–1330. PMLR (2017)
14. Han, Y., Wang, X., Leung, V.C.M., Niyato, D., Yan, X., Chen, X.: Convergence of edge computing and deep learning: a comprehensive survey. IEEE Commun. Surv. Tutor. **22**(2), 869–904 (2020)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of CVPR, pp. 770–778. IEEE (2016)
16. Hershey, S., Chaudhuri, S., Ellis, D.P.W.: CNN architectures for large-scale audio classification. In: Proceedings of ICASSP, pp. 131–135. IEEE (2017)
17. Hsu, W.-N., Bolte, B., Tsai, Y.-H.H., et al.: HuBERT: self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM Trans. Audio Speech Lang. Process. **29**, 3451–3460 (2021)
18. Howard, A.G., Zhu, M., Chen, B.: MobileNets: efficient convolutional neural networks for mobile vision applications (2017). arXiv:1704.04861

19. Howard, A., Sandler, M., Chu, G.: Searching for MobileNetV3. In: Proceedings of ICCV, pp. 1314–1324. IEEE (2019)
20. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of CVPR, pp. 4700–4708. IEEE (2017)
21. Huang, P.-Y., Chuang, S.-P., Lee, Y.-H.: SSAST: self-supervised audio spectrogram transformer. In: Proceedings of AAAI, pp. 2131–2139 (2022)
22. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of CVPR, pp. 7132–7141. IEEE (2018)
23. Kong, Q., Cao, Y., Iqbal, T., Wang, Y., Wang, W., Plumbley, M.D.: PANNs: large-scale pretrained audio neural networks for audio pattern recognition. IEEE/ACM Trans. Audio Speech Lang. Process. **28**, 2880–2894 (2020)
24. Lane, N.D., Bhattacharya, S., Georgiev, P., Forlivesi, C., Kawsar, F.: DeepX: a software accelerator for low-power deep learning inference on mobile devices. In: Proceedings of MobiSys, pp. 117–131. ACM (2016)
25. Mesaros, A., Heittola, T., Virtanen, T.: TUT database for acoustic scene classification and sound event detection. In: Proceedings of EUSIPCO, pp. 1128–1132. IEEE (2016)
26. Niculescu-Mizil, A., Caruana, R.: Predicting good probabilities with supervised learning. In: Proceedings of ICML, pp. 625–632. ACM (2005)
27. Park, D.S., et al.: SpecAugment: a simple data augmentation method for automatic speech recognition. In: Proceedings of Interspeech (2019)
28. Piczak, K.J.: Environmental sound classification with convolutional neural networks. In: Proceedings of MLSP, pp. 1–6. IEEE (2015)
29. Platt, J.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Smola, A., Bartlett, P., Scholkopf, B., Schuurmans, D. (eds.) Advances in Large Margin Classifiers, pp. 61–74. MIT Press (1999)
30. Salamon, J., Jacoby, C., Bello, J.P.: A dataset and taxonomy for urban sound research. In: Proceedings of ACM Multimedia, pp. 1041–1044. ACM (2014)
31. Salamon, J., Bello, J.P.: Deep convolutional neural networks and data augmentation for environmental sound classification. IEEE Sign. Process. Lett. **24**(3), 279–283 (2017)
32. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C.: MobileNetV2: inverted residuals and linear bottlenecks. In: Proceedings of CVPR, pp. 4510–4520. IEEE (2018)
33. Schneider, S., Baevski, A., Collobert, R., Auli, M.: wav2vec: unsupervised pre-training for speech recognition. In: Proceedings of Interspeech, pp. 3465–3469 (2019)
34. Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L.: Edge computing: vision and challenges. IEEE Internet Things J. **3**(5), 637–646 (2016)
35. Satyanarayanan, M.: The emergence of edge computing. Computer **50**(1), 30–39 (2017)
36. 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)
37. Tan, M., Le, Q.: EfficientNet: rethinking model scaling for convolutional neural networks. In: Proceedings of ICML, pp. 6105–6114. PMLR (2019)
38. Tan, M., Le, Q.: EfficientNetV2: smaller models and faster training. In: Proceedings of ICML, pp. 10096–10106. PMLR (2021)

39. Warden, P.: Speech commands: a dataset for limited-vocabulary speech recognition, (2018). arXiv:1804.03209
40. Zadrozny, B., Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In: Proceedings of KDD, pp. 694–699. ACM (2002)
41. van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv:1807.03748 (2018)