A Model Implementation to Detect Hate Speech Associated with Gender Violence by Natural Language Processing and Machine Learning

Genesis Calapaqui, David Guarderas, and Walter Fuertes(✉)

Department of Computer Science, Universidad de las Fuerzas Armadas ESPE, Sangolquí 171103, Ecuador
gicalapaqui@espe.edu.ec, adguarderas@espe.edu.ec, wmfuertes@espe.edu.ec

Abstract. The expression of hate speech against women on social media has become a worrying problem due to the popularity of these platforms and the reach they provide to such messages. This study addresses this problem from a technological perspective by developing an Artificial Intelligence model to detect and mitigate hate speech against women on social media, combining Natural Language Processing (NLP) and Machine Learning (ML) techniques. The process begins with an ETL process to collect and clean training data, which includes removing duplicates, null values, and special characters; transforming labels using a threshold of 0.5; and consolidating three datasets into a final corpus of 440,380 records with a balanced distribution between hate speech and non-hate speech categories. Next, both detection models are trained separately and then combined using weighting. The resulting classifier is integrated into a three-stage pipeline that incorporates Gemini Flash 1.5 for gender-specific filtering. The model is then implemented within a web prototype of a content wall, developed in Python using the Flask framework under the MVC architectural pattern and with a Firebase Firestore non-relational database to evaluate its performance in real-world environments. Finally, performance and quality tests of the system are conducted. The study demonstrates the relevance of the proposed solution, as the combination of BERT- and random forest-based models achieves 96% accuracy.

Keywords: Hate speech · Natural Language Processing · Machine Learning

1 Introduction

In recent years, hate speech has increased on social networks, seriously affecting women's lives and limiting their freedom of expression. Although much research has focused on the identification of explicit hate and violence, these problems also manifest themselves in more subtle ways, through stereotypes and normalized

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16768, pp. 202–219, 2027.
https://doi.org/10.1007/978-3-032-30488-9_13

attitudes that, although often unnoticed, have a profound and harmful impact on women and society as a whole [26].

Social networks have facilitated the rapid spread of hate speech, exacerbated by online anonymity and the increasing number of digital hate crimes. Given the huge volume of data, manual analysis is impractical and prone to bias [23]. As a result, several researchers have developed hate speech detection models for social networks using Natural Language Processing (NLP) and Machine Learning (ML) algorithms. For example, the FAST-RRN detection model achieved one of the highest accuracy rates at 98%. This is because it is a model within the field of ML, specifically Deep Learning, that uses neural networks to predict and classify hate speech. The second model to achieve high accuracy was BERT, with 97%. This is due to its transformer-based architecture and its pre-training as an NLP model [17].

The main objective of this study was to develop a model for detecting and preventing hate speech against women, combining NLP and ML techniques. To this end, we constructed a dataset of 440,380 labeled data points and trained two models: one based on NLP using BERT, and the other using ML with Random Forest. Subsequently, a three-stage pipeline was designed: first, BERT and Random Forest classify the text in parallel; second, their outputs are combined using probability-weighted fusion to produce a binary classification; and third, Gemini Flash 1.5 filters only the texts classified as hate speech to determine if the content is specifically directed at women. In addition, Google Translate was incorporated to extend the model's coverage to publications in English and Spanish, since the models were trained in English. Finally, a functional web prototype was developed as a message board that allows users to write comments; these are automatically analyzed by the system, and if hate speech is detected, the message is blocked before its publication.

The main contribution of this article is the design of a multi-stage moderation pipeline. This architecture comprises a hybrid model combining BERT and Random Forest, integrated with a contextual validation layer based on Gemini Flash 1.5, to discern whether hate speech is implicitly or explicitly directed at women. This addresses a limitation present in previous studies that focus on general toxicity. The system is validated by a web prototype that blocks content in real time. This demonstrates the viability of a scalable approach focused on preventing gender-based violence in the digital sphere.

The remainder of the article is organized as follows: Sect. 2 describes the related work and the theories that underpin this study. Section 3 presents the methodological design to detect and prevent hate speech. Section 4 presents the evaluation of the results, along with the quality and performance tests and their respective discussion. Finally, Sect. 5 presents the conclusions and future work.

2 Related Work

Previous studies, such as [8], aim to determine the best source of training data for detection models. X (formerly Twitter) is a social network whose content is

primarily text-based [10]. For this reason, several researchers extract the data necessary to train hate speech detection models from this social network using the X API [1]. Similarly, the platform's features allow developers to find relevant data, identifying hate-related content through hashtags, keywords, and search terms [19,20]. Other common sources for hate speech data include Facebook [4,15,16,24], the YouTube comments section [4,25,27,29], and others. Similarly, some researchers choose to use pre-made datasets available online [2,5,9,11,12, 21,32]. Even song lyrics can be a useful source of information, as demonstrated by [9], who collected data from songs containing offensive expressions for women.

Another aspect to consider when collecting data for training is the language of the data. English, being such a popular language globally, contains a wealth of data and datasets on various topics and biases, which contribute to the development of better artificial intelligence models, including hate speech detection models [10,16,19,20].

In recent years, several authors have focused on addressing the lack of data for other languages, which despite their large numbers of speakers worldwide, lack sufficient data for adequate detector training. Some of these languages include Arabic [4], Urdu [22], Spanish [25], and Hindi [19]. Some studies, such as [31], have gone a step further by training their models in multiple languages. In this case, models such as mBERT and XLM-R have been trained in more than a hundred languages.

According to the research community, BERT is one of the most widely used models to detect hate speech, as its transformer structure allows it to capture contextual and semantic relationships in English and other languages [27]. Studies such as [13] highlight its popularity for classifying the emotional tone within hate speech. Multiple models have also been built using traditional machine learning techniques, including Support Vector Machines (SVMs), Naive Bayes, and Random Forests. Although these models have limitations in their accuracy with complex texts, they are characterized by their efficiency in handling large amounts of data [1,2,11]. Zero-Shot Learning (ZSL) machine learning models, such as LLAMA2-13B, are also relevant in this field, as they, like BERT, utilize transformers. However, this model is distinguished by its ability to perform classification tasks in contexts that have not been previously trained, making it useful for multilingual environments [28].

Similarly, some authors consider the use of deep neural networks to detect sequential patterns and high-level features in text [10]. Examples of this include CNNs for extracting local features and LSTMs [15,20] for capturing long-term dependencies.

Other researchers, such as [30], consider the potential of large language models (LLMs), including ChatGPT and Bard. For example, they can be used to analyze sentiment, detect hate speech, and identify disinformation and propaganda campaigns. These models are continually improving and are a valuable contribution to the study and understanding of social phenomena, such as expressions of hate. Some researchers are using a combination of architectures to improve

detection performance and adapt to hate speech, which in many cases can be complex [6,25].

Research shows that combining NLP algorithms with other artificial intelligence techniques improves results in classification and pattern-detection tasks. The work presented in [7], for example, combines NLP with Deep Learning (DL) to analyze website content and detect phishing attacks. Essentially, they use NLP to pre-process the input text by removing irrelevant content that generates noise and reduces analysis performance. They then evaluate different DL algorithms to determine whether a word is phishing-related, using pattern recognition and weight adjustments as text from web pages is exposed. This provides greater accuracy when performing detection tasks.

Across these lines, [3] applied text mining combined with machine learning techniques to identify cyberattack patterns and trends in cybersecurity research, demonstrating that the integration of NLP-based methods with other artificial intelligence approaches improves the analysis and detection of harmful behaviors in digital environments. Similarly, [18] implemented a hybrid architecture that combines a variant of BERT, Nor-BERT, with an LSTM to detect hate speech in bilingual environments (Norwegian and English).

The FAST-RRN detection model is one of the most effective, with an accuracy of 98% [17], along with BERT, which has an accuracy of 97%. This is because their architectures enable them to understand and detect linguistic patterns in data more deeply. They also apply learning techniques such as fine-tuning to leverage previously learned linguistic representations and adapt their knowledge to specific tasks. It should be noted that their efficiency is maximized due to the large volume of data developers use to train these models [20].

Unfortunately, the reviewed models have limitations that lead to bias in their hate-speech detection. This is the case with [14]. This study focused on the use of BERT to detect hate speech in German. However, while it showed optimal performance with offensive language that uses specific terms, its performance dropped considerably when detecting hate speech expressed with terms and expressions currently used by young people.

3 Research Design

3.1 Proposed Architecture

In this study, we implemented a system based on the client-server architecture pattern. We divided the detection and classification model developed into two separate modules: The server, on the one hand, contains and runs the model, along with the various plugins it uses to enhance its performance. Meanwhile, the client comprises a prototype web application that provides users with a platform for posting content to a wall.

As shown in Fig. 1, the system implements a three-stage pipeline. First, BERT and Random Forest classify the text in parallel using their respective representations. Second, their probabilities are combined using weighted fusion (60/40), producing a binary classification. Third, only texts classified as hate

speech are sent to Gemini Flash 1.5, which determines whether the content is specifically targeted at women. This design separates general detection from gender-specific filtering.

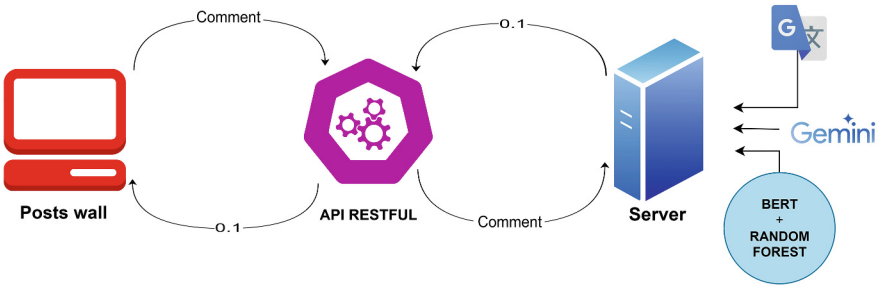

Fig. 1. System Architecture Diagram for the prototype.

3.2 Classification Models and Techniques Used

This section describes the models used for hate speech detection. We include BERT, a natural language processing model; Random Forest, a decision tree-based algorithm; and Gemini Flash, a complementary tool that improves classification efficiency and accuracy.

Dataset Construction: The data set was constructed using an Extraction, transformation and Loading (ETL) process. In the extraction phase, data was collected from Kaggle (http://www.kaggle.com/datasets, a repository of datasets on various topics for researchers, developers, and learners. Since there was no specific data set on hate speech against women in English, three more general datasets were selected, containing 24,783, 1,048,570, and 1,804,874 records, respectively, with tags such as *hate_speech*, *offensive_language*, *insult*, *sexual_explicit*, and *obscene*.

In the cleaning and transformation phase, duplicates, null values, and special characters such as usernames were removed from text extracted from social networks. Additionally, the original labels were transformed into Boolean values (i.e., 1 for hate speech and 0 otherwise), using a threshold of 0.5 to assign each text a category.

Finally, in the unification phase, the data were combined into a single file with two columns: Content (comment text) and Label (hate speech indication). After the ETL process, the final dataset was reduced to 440,380 records. Subsequently, the data set was divided into 80% for training and 20% for testing, ensuring a suitable distribution for model training.

BERT: BERT was chosen as the basis for developing the NLP model. BERT is a pretrained model developed by a group of Google researchers to improve next-generation language representation. BERT has surpassed models such as the "Generative Pretrained Transformer" (GPT) and "Embeddings from Language" (ELMo). It is characterized by its transformer-based technology, which transforms one sequence into another, eliminates recurrences, and replaces them with attention mechanisms to handle dependencies between inputs and outputs. This means that BERT analyzes all words in a sentence simultaneously, allowing it to identify the relationships between them and better understand the context. [27].

Unlike previous NLP models, which use unidirectional training, BERT employs bidirectional training, reading the entire sentence at once and analyzing the context surrounding a word to better understand its meaning. BERT also uses masking techniques, replacing parts of a sentence with tokens called masks, and then predicting those words based on the context of the unmasked words, thus optimizing the performance of the model [27].

Training BERT is computationally expensive. However, once trained, the model can be easily applied to text classification tasks [27].

BERT training was carried out as follows: upon loading the datasets, they were batched for improved data handling performance. The data were then tokenized; that is, the sentences were converted into word-level vectors, each sentence separated by a word. Subsequently, each batch was fed into the pre-trained model in a loop. Through bidirectional passing, a prediction was obtained for each data point (1 or 0), which was then compared with the actual label to calculate the loss. This value was then used for the evaluation phase. To achieve better results in terms of model accuracy, epoch training was applied, which means that the aforementioned process was repeated several times (i.e., in this case 3) (see Fig. 2).

Random Forest: The following hyper parameters were defined for training the Random Forest model:

- **Node size:** The default values were used, setting a minimum of two samples to split a node and one sample for leaf nodes, allowing the capture of specific patterns in the data.
- **Number of trees:** Training was performed over 10 epochs, increasing the number of trees by 10 in each epoch, until a total of 100 trees were reached.
- **Number of features sampled:** 5000 features were defined during text vectorization, and each tree used approximately $\sqrt{5000} \approx 71$ features.

Since machine learning models cannot directly process text, a pre-processing step was applied that converted the texts into numerical representations using the TF-IDF technique, generating a 5000-feature matrix.

Training started with 10 trees and was gradually increased to 100. Each tree made a prediction and the final result was obtained by majority vote, consolidating a robust decision based on the combination of multiple individual predictions (see Fig. 3).

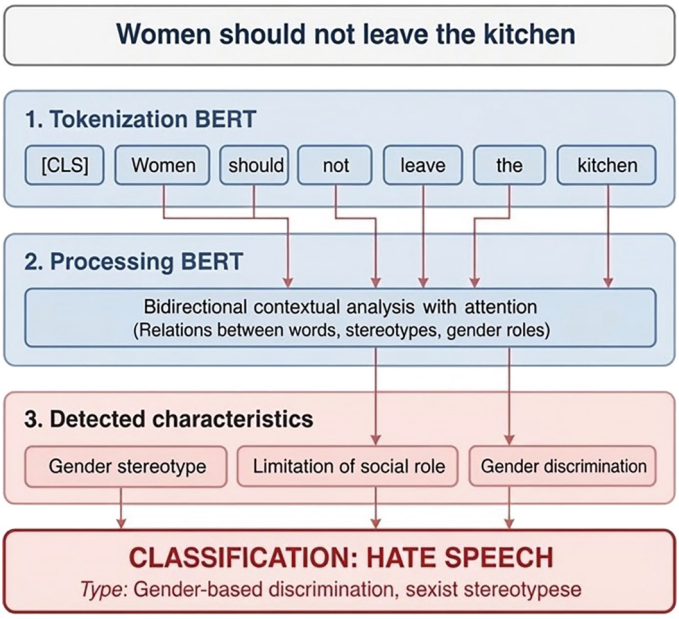

Fig. 2. BERT Training

Model Combination: To improve the accuracy of hate speech detection, the **Random Forest (RF)** and **BERT** models were combined using probability weighting. Each model was assigned a weight based on its individual performance.

- **Random Forest (RF)**: 40%
- **BERT**: 60%

The weighting was not based solely on precision (94% BERT, 93% RF), but also on theoretical foundations: BERT has consistently demonstrated a greater capacity to capture semantic context in implicit hate speech [18,27], while Random Forest is more sensitive to superficial lexical features. This asymmetry justifies the favoring BERT in the ensemble. Regarding the 0.5 threshold, it corresponds to the standard decision boundary in binary classification under balanced distributions [21], a condition that is met in the constructed corpus.

From a mathematical perspective, the weights can be approximated proportionally to the accuracy of each model as follows:

$$w_{\text{BERT}} = \frac{94}{94+93} \approx 0.5027, \quad w_{\text{RF}} = \frac{93}{94+93} \approx 0.4973 \tag{1}$$

However, it was decided to manually adjust the proportion to 60%-40% to reflect BERT's qualitative advantage and its greater ability to generalize to unseen data.

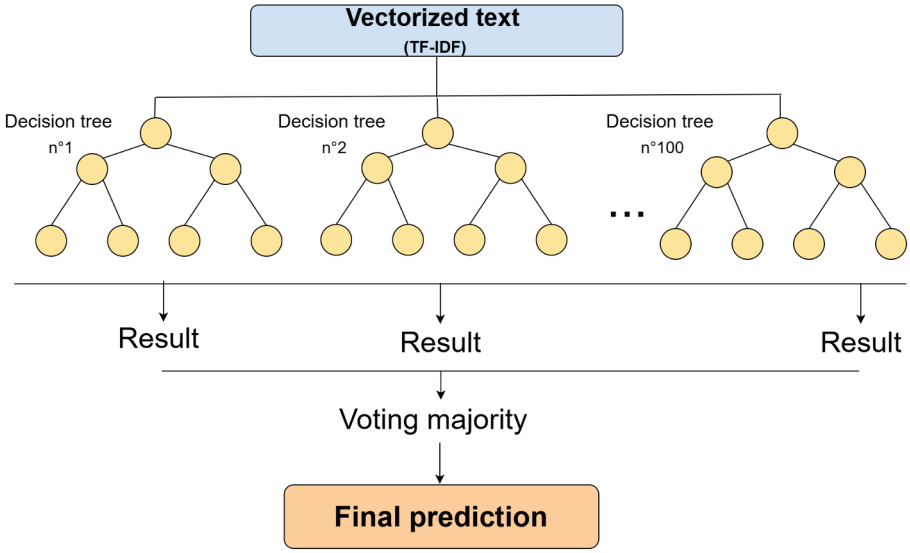

Fig. 3. Random Forest Training

Each model processes the text independently: RF uses TF-IDF vectorization, while BERT employs tokenization. Once the prediction probabilities of each model are obtained, the weighted combination is performed (see Fig. 4).

As shown, the individual probabilities of each model are multiplied by their respective weights and then summed to obtain the weighted probability. The class with the highest final probability determines the system's prediction. This combination leverages BERT's ability to understand language and RF's ability to capture specific patterns, thus improving the accuracy of the hate speech detection system.

Gemini Flash: Gemini Flash 1.5 acts as a contextual specificity filter in the third stage of the pipeline, applying exclusively to texts already classified as hate speech by the hybrid model. Its role is not the general detection of hate speech, but rather determining whether the offensive content is specifically directed at women. This separation is architecturally relevant, as general-purpose models tend to lack sensitivity to gender-specific targeting. The Gemini Flash API is queried via a structured prompt that identifies explicit references, generalizations, or stereotypes about women, returning 'yes' to block the content or 'no' otherwise. This approach does not require fine-tuning on gender-specific corpora and handles implicit expressions that rule-based filters miss.

3.3 Web Interface Implementation

We develop a web prototype of a posting wall to test the model's functionality in a near-real-world environment and evaluate it from the user's perspective. This

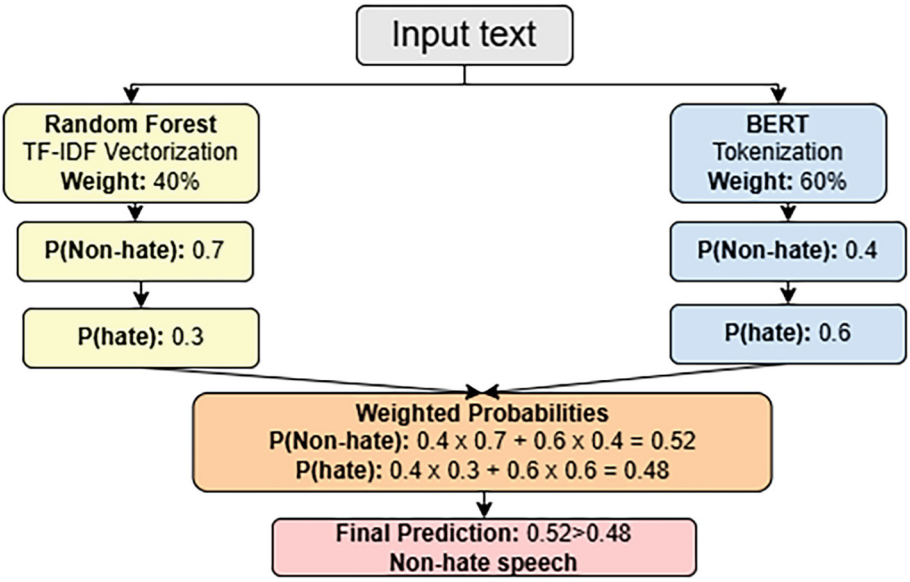

Fig. 4. Weighted scoring

application was developed using the Model-View-Controller (MVC) architectural pattern, which separates the user interface from the business logic and user interaction handling. We also used the Object-Oriented Programming (OOP) paradigm to generate more structured and understandable code by abstracting the different concepts related to the website's functionality (see Fig. 5).

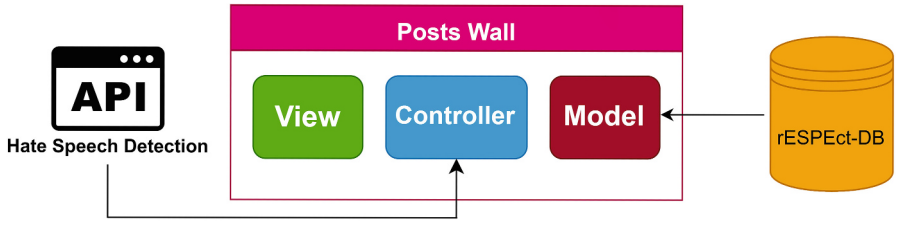

Fig. 5. Web application software architecture

We used Python as the programming language to develop the application. We used Flask, a Python framework known for its ease of use in developing web applications with minimal code. A non-relational database was chosen to store the application's data, as it enables efficient handling of the large volumes of data typically required by social media platforms. Additionally, this type of database offers structural flexibility, facilitating modifications to the business

logic if needed. For this purpose, the Firebase Firestore database was used to store the data in the cloud.

The application consists of a posting wall where users can register, log in, and access the main page. There, they can post short text messages, edit, delete, read other users' posts, and like them. When a user attempts to post on the wall, the content is sent via a REST API to a developed detection model. This model identifies whether the text constitutes hate speech against women. If it is not hate speech, the post is published normally. Otherwise, the post is blocked, and the user is notified that they attempted to publish offensive text against women.

4 Results

4.1 Trained Model Results

To evaluate the performance of trained models, we use the Holdout Set method, dividing the data set into 80% for training (352,304 data points) and 20% for testing (88,076 data points). Within the test set, 44,478 comments were labeled as 0 and 43,598 as 1.

We evaluated three approaches: BERT, Random Forest, and a weighted combination of both models. The evaluation used Accuracy, Recall, and F1-score metrics to compare the performance of each model in detecting hate speech. Detailed results are presented in Table 1.

Table 1. Evaluation results of the models

Label	Model	Precision	Recall	F1-Score
0	BERT	0.94	0.95	0.94
0	Random Forest	0.95	0.91	0.93
0	Combined	0.97	0.95	0.96
1	BERT	0.93	0.91	0.93
1	Random Forest	0.95	0.97	0.93
1	Combined	0.97	0.96	0.96
Support (0)				44478
Support (1)				43598

The weighted fusion (BERT 60% and Random Forest 40%) performs better than the individual models because it combines BERT's contextual semantic understanding with Random Forest's lexical pattern detection. The standard 0.5 decision threshold is maintained in binary classification with a balanced distribution, ensuring an optimal balance between false positives and negatives and preventing overfitting. This compensates for the individual shortcomings of each model and reduces classification errors. (See Fig. 6).

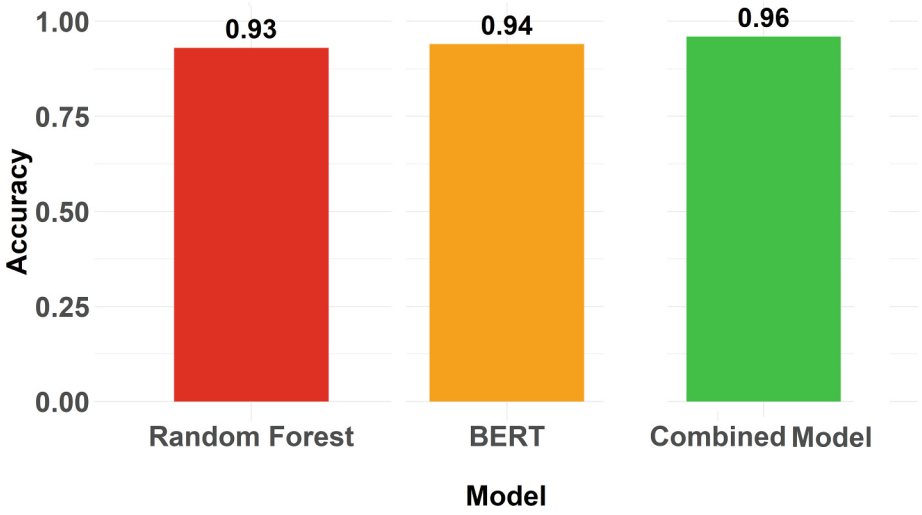

Fig. 6. Comparative chart of model accuracy.

To validate these results, we present the confusion matrix of the evaluated models (see Fig. 7), which shows the values corresponding to True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN).

The comparative confusion matrix between the BERT, Random Forest and combined models is shown in Fig. 7. The combined model exhibits the best overall performance, as evidenced by the lowest number of false negatives and false positives compared to BERT and Random Forest. These results indicate that integrating both approaches improved recall and accuracy by reducing errors associated with failing to detect hate speech when it is indeed hate speech (FN) and with misclassifying non-offensive messages as hate speech (FP). Furthermore, the combined model achieved the highest values for true positives and true negatives, indicating a greater capacity to correctly distinguish between offensive and non-offensive messages.

On the other hand, Fig. 8 shows the comparative behavior between the ROC curves (Receiver Operating Characteristic) and the AUC (Area Under the Curve) of the three models evaluated. As shown in the AUC, BERT demonstrates the greatest effectiveness in discriminating between messages with and without hate speech, regardless of the applied threshold. As shown, with a difference of only 0.004, it is followed by the model that combines BERT with Random Forest, while the latter shows the lowest overall capacity among the three models, although it remains quite favorable, above 0.9.

When comparing the results from the Confusion Matrix and the ROC-AUC plot, the combined model continues to perform better. Although BERT achieved a higher AUC, the difference between its AUC and the combined model's is minimal. Furthermore, it is important to note that this graph calculates the overall performance of the models. However, under the calibration metrics used in model

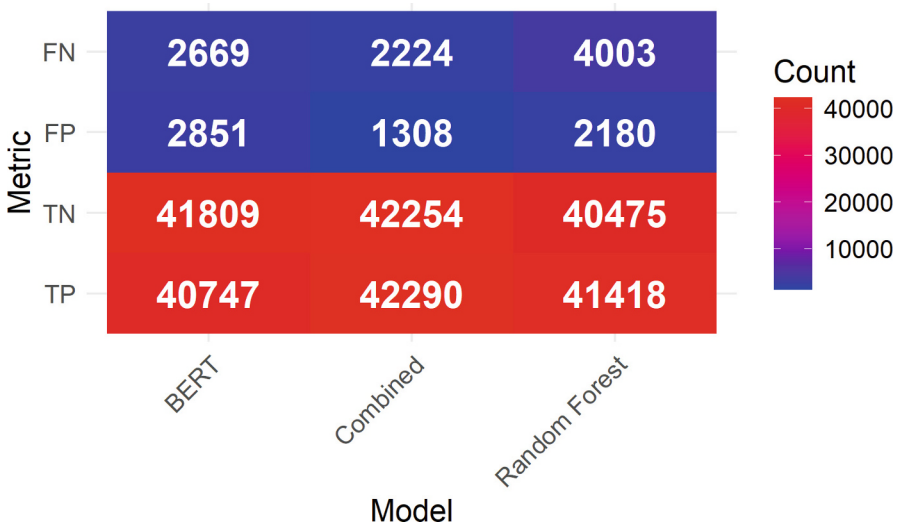

Fig. 7. Confusion matrix for evaluating model performance.

construction, specifically, the conditions of this case study, the combination of BERT and Random Forest continues to yield better results, as demonstrated by the confusion matrix. This matrix shows a clear superiority in the combined model, as it not only identifies more true positives and negatives than the other models but also produces fewer errors. Contrary to the AUC figures, the difference between the models' confusion matrices is more significant, favoring the combined model.

4.2 API Performance Evaluation

To verify the behavior of the REST API under different load levels, performance tests were conducted using JMeter. This tool allows for the simulation of multiple simultaneous users interacting with the system, facilitating the measurement of response times, error rates, and the number of requests processed per second. Such analyzes help identify potential bottlenecks and optimize the system's infrastructure.

During the tests, different load scenarios were established, varying the number of concurrent users and the request frequency to examine the API's performance under high-traffic conditions. Indicators such as average response time, latency, and CPU and memory consumption on the server, as well as system stability under a progressive increase in requests, were evaluated.

As shown in Table 2, the tests started with a concurrency of 100 users and were progressively increased to 600, at which point failures in request execution were detected.

The performance tests presented show a notable decrease in responsiveness as the number of concurrent users increases. With up to 300 users, the system

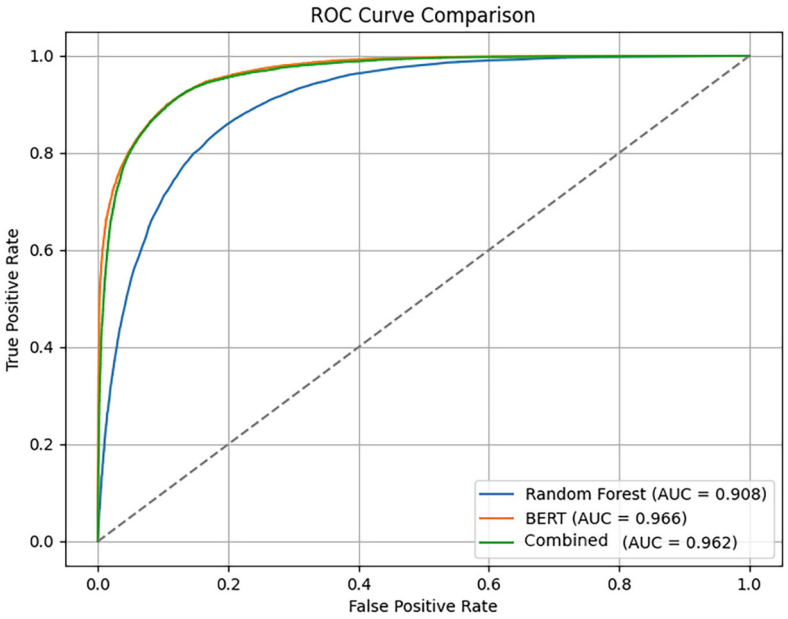

Fig. 8. ROC-AUC graph to evaluate and compare model performance.

Table 2. Performance Test Results

Users	Std. Deviation	Error (%)	Throughput (req/sec)
100	4927.90	0.00	5.5
200	9731.98	0.00	5.7
300	15191.54	0.00	5.6
400	17303.65	24.73	7.3
500	16822.24	40.40	9.3
600	16928.18	50.83	11.2
Total	16303.78	28.71	4.8

maintains a stable standard deviation and a 0% error rate, indicating reliable performance under moderate load. However, starting at 400 users, the error rate increases considerably, reaching 50.83% with 600 users, suggesting that the API struggles to handle high traffic.

4.3 Web Client Quality Assessment

Quality testing focused on the developed web client, as it was considered prudent to ensure a pleasant user experience when using the application, since this is where the user will interact with the model.

To this end, the tests were performed using Lighthouse. This is an open-source Google tool that improves the quality of web pages. Lighthouse enables auditing several essential aspects, including performance, accessibility, Progressive Web Apps (PWAs), and Search Optimization (SEO).

Performing quality tests with Lighthouse is straightforward as it is accessible through the developer tools in the Google Chrome browser. After properly configuring, Lighthouse generates a diagnostic report on the website's quality and identifies areas for improvement.

At the time of the first test, the resulting diagnostic report showed metrics that, while not entirely low, indicate a need for improvement to reach the expected quality standards. The tool suggested several steps to improve the quality and best practices of the application's development, such as replacing GIF animations with WebM, replacing PNG or JPEG images with WebP, providing accessible button names and including alternative text for images, and adding meta descriptions so search engines can find and recommend the website.

After implementing these changes, the website was re-evaluated with Lighthouse, resulting in the highest quality scores, as shown in Fig. 9.

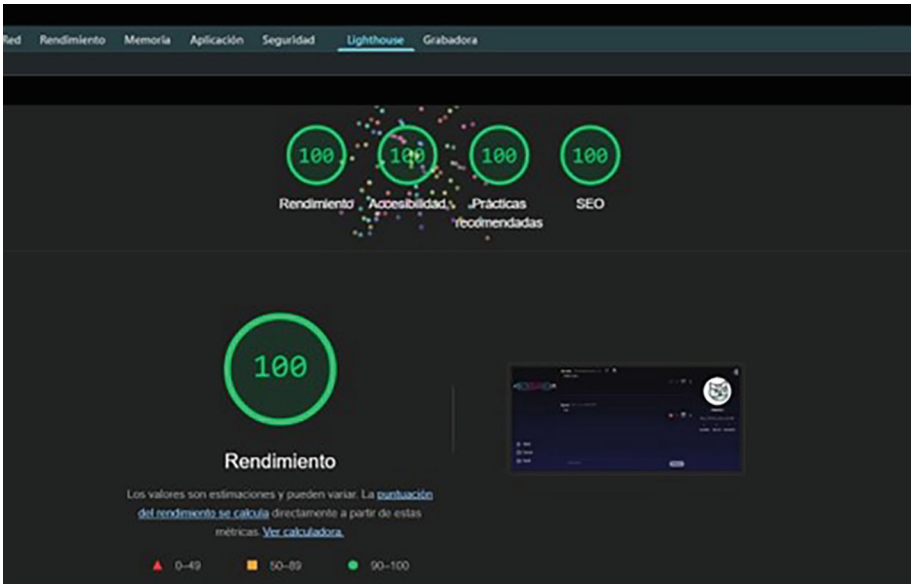

Fig. 9. Second Web Quality Lighthouse Test

4.4 Discussion

The results demonstrate the effectiveness of the hybrid architecture in detecting hate speech against women. The precision of the combined model (96%) exceeds

that of BERT (94%) and Random Forest (93%) separately. BERT's contextual semantic understanding, along with Random Forest's lexical pattern detection, significantly reduces errors associated with content review and filtering costs. The combined model registered 2,224 false negatives, compared to 2,669 for BERT and 4,003 for Random Forest, for a 44.5% reduction. However, the qualitative analysis of residual errors reveals shortcomings in three areas: (i) subtle sarcasm (e.g., "it is good that women don't drive," a comment interpreted as neutral due to the lack of explicitly offensive vocabulary), (ii) contextual ambiguity (e.g., phrases from academic debates or news reports with sensitive terms that are not intended to be hostile), and (iii) cultural nuances not reflected in the training dataset. The AUC of BERT (0.966) slightly outperforms the combined model (0.962). However, it does not reflect the behavior of the model under a fixed threshold of 0.5. In this scenario, the hybrid model demonstrates superiority, as it prioritizes gender-specificity over overall toxicity.

Compared with the existing literature, the proposed model performs competitively. Reference models such as FAST-RNN (98%) and fine-tuned BERT(97%) achieve higher scores, but were trained on narrow domains and with specialized corpora, conditions that favor high metrics without guaranteeing generalization. In contrast, hybrid architectures such as Hashmi and Yayilgan [18], which combine Nor-BERT with LSTM, corroborate the trend that integrating models improves performance in complex tasks, consistent with the findings of this study. The main differentiation contribution lies in the incorporation of Gemini Flash 1.5 as a contextual specificity layer, which adds the ability to distinguish whether hate speech is specifically directed at women, a dimension that most comparable studies do not explicitly address.

However, it should be noted that the model was initially developed in English, so its application in other languages would benefit from adaptation or translation. Training exclusively in English requires translation into other languages, which introduces noise that degrades the detection of implicit or colloquial expressions [14]. Furthermore, performance tests revealed that the system remains stable only up to 300 concurrent users, with an error rate of 50.83% at 600 users, limiting its applicability on high-traffic platforms. Finally, constructing the data set from generic Kaggle sources can introduce labeling biases that affect the detection of subtle gender-based hate.

5 Conclusions and Future Work

This study developed and implemented a multi-stage moderation pipeline that detects and prevents hate speech against women by integrating natural language processing and machine learning. This hybrid architecture achieved 96% accuracy by combining BERT's semantic understanding with Random Forest's pattern-detection capabilities. This resulted in a 44.5% reduction in false negatives compared to Random Forest alone. We include Gemini Flash as a filter to improve the detection of comments targeting women provides a competitive advantage over more traditional models. This leads to the conclusion that a

multi-stage approach is more effective against online violence by improving the contextualization and specificity of the processing.

In future work, we propose incorporating multilingual models such as mBERT or XLM-R to reduce the noise introduced by machine translation in input from other languages. To improve scalability, promising approaches include model distillation, asynchronous request queues, and container-based horizontal scaling. We also propose evaluating deep neural networks based on Transformers, applying fine-tuning techniques. Finally, we could explore the integration of emotion analysis to improve hate speech detection from an affective perspective.

Acknowledgements. The authors express their gratitude to the Distributed Systems, Cybersecurity, and Content Research Group (RACKLY) for the use of the High-Performance Systems Research Laboratory and its specialized software and hardware.

References

1. Alhayan, F., et al.: Detection of cyberhate speech towards female sport in the Arabic Xsphere. PeerJ Comput. Sci. **10** (2024). https://doi.org/10.7717/peerj-cs.2138
2. Ali, M., Muhammad, A., Asad, M., Sajawal, M., Alexopoulos, C., Charalabidis, Y.: Towards Perso-Arabic Urdu language hate detection using machine learning: a comparative study based on a large dataset and time-complexity. In: Proceedings of the 26th Pan-Hellenic Conference on Informatics, pp. 317–321 (2023). https://doi.org/10.1145/3575879.3576011
3. Andrade, R.O., Fuertes, W., Cazares, M., Ortiz-Garcés, I., Navas, G.: An exploratory study of cognitive sciences applied to cybersecurity. Electronics **11**(11), 1692 (2022). https://doi.org/10.3390/electronics11111692
4. Badri, N., Kboubi, F., Habacha Chaibi, A.: Abusive and hate speech classification in Arabic text using pre-trained language models and data augmentation. ACM Trans. Asian Low-Resour. Lang. Inf. Process. (2024). https://doi.org/10.1145/3679049
5. Bandyopadhyay, R., Assenmacher, D., Alonso-Moral, J.M., Wagner, C.: Sexism detection on a data diet. In: Companion Publication of the 16th ACM Web Science Conference, pp. 94–102 (2024). https://doi.org/10.1145/3630744.3663609
6. Bashar, M.A., Nayak, R., Luong, K., Balasubramaniam, T.: Progressive domain adaptation for detecting hate speech on social media with small training set and its application to COVID-19 concerned posts. Soc. Netw. Anal. Min. **11**(1), 1–18 (2021). https://doi.org/10.1007/s13278-021-00780-w
7. Benavides-Astudillo, E., Fuertes, W., Sanchez-Gordon, S., Nuñez-Agurto, D., Rodríguez-Galán, G.: A phishing-attack-detection model using natural language processing and deep learning. Appl. Sci. **13**(9), 5275 (2023)
8. Calapaqui, G., Guarderas, D., Fuertes, W., López, A., Aules, H.: Detection of hate speech on on-line social platforms using machine learning and natural language processing - a literature review. In: Rocha, A., Peñalvo, F.G., Costa, C.J., Gonçalves, R. (eds.) Proceedings of 20th Iberian Conference on Information Systems and Technologies (CISTI 2025), pp. 441–452. Springer Nature Switzerland, Cham (2026)

9. Calderon-Suarez, R., Ortega-Mendoza, R.M., Montes-Y-Gomez, M., Toxqui-Quitl, C., Marquez-Vera, M.A.: Enhancing the detection of misogynistic content in social media by transferring knowledge from song phrases. IEEE Access **11**, 13179–13190 (2023). https://doi.org/10.1109/ACCESS.2023.3242965
10. Chiril, P., Pamungkas, E.W., Benamara, F., Moriceau, V., Patti, V.: Emotionally informed hate speech detection: a multi-target perspective. Cogn. Comput. **14**(1), 322–352 (2021). https://doi.org/10.1007/s12559-021-09862-5
11. Daouadi, K.E., Boualleg, Y., Guehairia, O.: Systematic investigation of recent pre-trained language model for hate speech detection in Arabic tweets. ACM Trans. Asian Low-Resour. Lang. Inf. Process. (2024). https://doi.org/10.1145/3674970
12. Dorris, W., Hu, R.R., Vishwamitra, N., Luo, F., Costello, M.: Towards automatic detection and explanation of hate speech and offensive language. In: Proceedings of the Sixth International Workshop on Security and Privacy Analytics, pp. 23–29 (2020). https://doi.org/10.1145/3375708.3380312
13. Escobar Díaz, A., Rivadeneira, R., Fuertes, W.: Emotional tone detection in hate speech using machine learning and NLP: methods, challenges, and future directions—a systematic review. Appl. Sci. **15**(23) (2025). https://doi.org/10.3390/app152312686
14. Fillies, J., Paschke, A.: Youth language and emerging slurs: tackling bias in BERT-based hate speech detection. AI Ethics **1**(1), 1–12 (2025). https://doi.org/10.1007/s43681-025-00701-z
15. Ganfure, G.O.: Comparative analysis of deep learning based Afaan Oromo hate speech detection. J. Big Data **9**(1), 76 (2022). https://doi.org/10.1186/s40537-022-00628-w
16. Han, H., Asif, M., Awwad, E.M., Sarhan, N., Ghadi, Y.Y., Xu, B.: Innovative deep learning techniques for monitoring aggressive behavior in social media posts. J. Cloud Comput. **13**(1), 19 (2024). https://doi.org/10.1186/s13677-023-00577-6
17. Hashmi, E., Yayilgan, S.Y.: Multi-class hate speech detection in the Norwegian language using FAST-RNN and multilingual fine-tuned transformers. Complex Intell. Syst. **10**(3), 4535–4556 (2024). https://doi.org/10.1007/s40747-024-01392-5
18. Hashmi, E., Yayilgan, S.Y., Abomhara, M.: Metalinguist: enhancing hate speech detection with cross-lingual meta-learning. Complex Intell. Syst. **11**(1), 179 (2025)
19. Kapil, P., Kumari, G., Ekbal, A., Pal, S., Chatterjee, A., Vinutha, B.N.: HHSD: Hindi hate speech detection leveraging multi-task learning. IEEE Access **11**, 101460–101473 (2023). https://doi.org/10.1109/ACCESS.2023.3312993
20. Kumbale, S., Singh, S., Poornalatha, G., Singh, S.: BREE-HD: a transformer-based model to identify threats on Twitter. IEEE Access **11**, 67180–67190 (2023). https://doi.org/10.1109/ACCESS.2023.3291072
21. Marshan, A., Nizar, F.N.M., Ioannou, A., Spanaki, K.: Comparing machine learning and deep learning techniques for text analytics: detecting the severity of hate comments online. Inf. Syst. Front. (2023). https://doi.org/10.1007/s10796-023-10446-x
22. Mehmood, F., Ghafoor, H., Asim, M.N., Ghani, M.U., Mahmood, W., Dengel, A.: Passion-Net: a robust precise and explainable predictor for hate speech detection in Roman Urdu text. Neural Comput. Appl. **36**(6), 3077–3100 (2024). https://doi.org/10.1007/s00521-023-09169-6
23. Mullah, N.S., Zainon, W.M.N.W.: Advances in machine learning algorithms for hate speech detection in social media: a review. IEEE Access **9**, 88364–88376 (2021). https://doi.org/10.1109/ACCESS.2021.3089515

24. Polignano, M., Basile, P., de Gemmis, M., Semeraro, G.: Hate speech detection through ALBERTO Italian language understanding model. CEUR Workshop Proc. **2521** (2019)
25. Redondo, R.P.D., Vilas, A.F., Merino, M.R., Rodríguez, S.M.V., Guijarro, S.T., Hafez, M.M.: Anti-sexism alert system: identification of sexist comments on social media using AI techniques. Appl. Sci. (Switzerland) **13**(7) (2023). https://doi.org/10.3390/app13074341
26. Rodríguez-Sánchez, F., Carrillo-de Albornoz, J., Plaza, L.: Automatic classification of sexism in social networks: an empirical study on twitter data. IEEE Access **8**, 219563–219576 (2020). https://doi.org/10.1109/ACCESS.2020.3042604
27. Salminen, J., Hopf, M., Chowdhury, S.A., Jung, S., Almerekhi, H., Jansen, B.J.: Developing an online hate classifier for multiple social media platforms. HCIS **10**(1), 1–34 (2020). https://doi.org/10.1186/s13673-019-0205-6
28. Shaikh, S., Yayilgan, S.Y., Abomhara, M., Zoto, E.: Multilingual user perceptions analysis from twitter using zero shot learning for border control technologies. Soc. Netw. Anal. Min. **15**(19), 1–24 (2025)
29. Shanmugavadivel, K., Subramanian, M., Shri Durga, R., Srigha, S., Sree Harene, J., Yasvanth Bala, P.: KEC AI nlp@dravidianlangtech:abusive comment detection in Tamil language using machine learning techniques. In: DravidianLangTech 2023 - 3rd Workshop on Speech and Language Technologies for Dravidian Languages, Associated with 14th International Conference on Recent Advances in Natural Language Processing, RANLP 2023 - Proceedings, pp. 293–299 (2023). https://doi.org/10.26615/978-954-452-085-4_043
30. Thapa, S., et al.: Large language models (LLM) in computational social science: prospects, current state, and challenges. Soc. Netw. Anal. Min. **15**(4), 4 (2025). https://doi.org/10.1007/s13278-025-01428-9
31. Xu, Y., et al.: A survey on multilingual large language models: corpora, alignment, and bias. Front. Comput. Sci. **19**(11), 1911362 (2025). https://doi.org/10.3389/fcomp.2023.1911362. open Access at link.springer.com and journal.hep.com.cn
32. Yuan, L., Wang, T., Ferraro, G., Suominen, H., Rizoiu, M.A.: Transfer learning for hate speech detection in social media. J. Comput. Soc. Sci. **6**(2), 1081–1101 (2023). https://doi.org/10.1007/s42001-023-00224-9