Offensive Text Detection Using Hybrid Ensemble Text Classifier: An Experimental Study

P. Amalraj[1], A. P. Haripriya [2], S. Burhana Thaj[1] and S. Suryapriya[2]

ABSTRACT

Social media platforms' rapid expansion has given users the freedom to voice their thoughts, promoting international cooperation and communication. However, this freedom has also contributed to the spread of abusive language and hate speech, which can have negative impacts on both individuals and communities. Maintaining safe online spaces while upholding the right to free speech has made identifying such content a crucial task. Recent developments in machine learning (ML) provide encouraging solutions by making it possible to automatically identify inappropriate language with a high degree of accuracy. However, because hate speech and offensive language are complex and heavily influenced by context, intent, and cultural variables, it is still difficult to discriminate between them with precision. The dataset used in this work contains labeled tweets categorized into three classes: hate speech, offensive language, and neutral language. Initially, an analysis of individual ML models, including classical algorithms, was performed. The performance of these algorithms was evaluated using recall, precision, F1-score, and accuracy metrics. To further improve detection accuracy, a novel ensemble strategy is proposed that combines the three best-performing models. The ensemble offers a robust solution for offensive language detection by leveraging the strengths of its component models. Experimental evaluations show that the ensemble model achieves superior performance compared to individual classifiers, which had accuracies between 85% and 89%, achieving an accuracy of 90% with high recall across all categories.

Keywords: Offensive language detection, Hate speech classification, Soft-voting ensemble, TF-IDF feature engineering, Twitter tweet dataset

[1]Department of Artificial Intelligence & Data Science, Government Engineering College, Trivandrum, Kerala, India
[2]Department of IT, Government Engineering College, Trivandrum, Kerala, India

Correspondence to:
P. Amalraj,
amalr8691@gmail.com

Additional material is published online only. To view please visit the journal online.

Cite this as: Amalraj P, Haripriya AP, Burhana Thaj S and Suryapriya S. Offensive Text Detection Using Hybrid Ensemble Text Classifier: An Experimental Study. Premier Journal of Science 2025;15:100219

DOI: https://doi.org/10.70389/PJS.100219

Peer Review

Received: 14 August 2025

Last revised: 22 October 2025

Accepted: 17 December 2025

Version accepted: 3

Published: 30 January 2026

Ethical approval: N/a

Consent: N/a

Funding: N/a

Conflicts of interest: N/a

Author contribution:
P. Amalraj, A. P. Haripriya, S. Burhana Thaj and S. Suryapriya – Conceptualization, Writing – original draft, review and editing

Introduction

The widespread use of social media platforms has revolutionized human interaction and communication. However, this advancement has also led to the proliferation of offensive language and hate speech, which can have serious social and psychological repercussions. Ensuring freedom of expression while maintaining safe and inclusive online spaces requires the efficient detection and moderation of such harmful content.

Offensive text content is commonly found on platforms such as Instagram, YouTube, and Facebook. This study focuses specifically on Twitter due to its public accessibility and text-rich nature.[1] The dynamic and context-dependent nature of abusive language makes traditional moderation techniques such as manual review or rule-based filtering largely ineffective. With their improved accuracy and scalability, machine learning (ML) approaches have emerged as powerful tools for automating the detection of harmful content.[2,3] Nonetheless, distinguishing between hate speech, offensive language, and neutral text remains a challenging task that demands a nuanced understanding of context, intent, and linguistic variation.[4]

The dataset used in this study comprises approximately 25,000 tweets categorized into three classes: offensive, hate, and neutral. Various ML models including Random Forest, XGBoost, Logistic Regression, support vector machine (SVM), and Naïve Bayes were trained and evaluated using standard metrics such as accuracy, precision, and recall.[5,6] To enhance performance, an ensemble model was constructed by combining the three top-performing algorithms through a soft-voting mechanism. Unlike earlier works that focus primarily on accuracy, the novelty of this study lies in emphasizing both efficiency and transparency. The proposed ensemble is lightweight, capable of processing around 1,200 tweets per second on a standard CPU, demonstrating its potential for real-time deployment. Moreover, the inclusion of per-class metrics and qualitative error analysis enhances interpretability and fairness dimensions often overlooked in prior research. Experimental results confirm that the ensemble model surpasses individual classifiers, achieving 90% accuracy and high recall across all classes. This study highlights the strength of ensemble learning for offensive language detection and its potential for interpretable, scalable, and transparent moderation systems on social media.

The remainder of this paper is organized as follows: Section "Related Works" reviews related research, Section "Methodology" explains the methodology, Section "Results and Analysis" presents results and analysis, and Section "Conclusion" concludes the study.

Related Works

This review of literature investigates numerous ML, deep learning, and ensemble-based methods applied in past research and their efficacy, challenges, and areas for improvement.

In the work,[7] an experiment on detecting offensive language on social media is conducted by the authors. It is done by employing ML techniques. Metrics used for comparison among such algorithms are precision, recall, F1-score, and accuracy. Algorithms used have been Naive Bayes, SVM, XGBoost, k-nearest neighbors, Decision Tree, and Random Forest. The results show that the best combination was Random Forest, where Term Frequency-Inverse Document Frequency (TF-IDF) features achieved an accuracy of up to 97% using the dataset of hate speech-containing tweets.

Guarantor: P. Amalraj

Provenance and peer-review: Unsolicited and externally peer-reviewed

Data availability statement: N/a

The authors in the work[8] explored on how different deep learning models can be employed for the identification of hate speech, namely convolutional neural network (CNN), bidirectional long short-term memory (BiLSTM) with attention, pre-trained bidirectional encoder representations from transformers (BERT), and fine-tuned Robustly Optimized BERT Pretraining Approach (RoBERTa). All of these are used in a ternary classification task, namely the three categories: hate, offensive, non-hate, and assessed by using metrics like F1-score, Matthews Correlation Coefficient, and accuracy. The Results of transformer-based models show that fine-tuned RoBERTa leads in comparative testing, with robust performance on both training and validation sets. There are still limitations such as dataset imbalance and challenges in handling biased pre-trained embeddings were noted, underscoring the need for further improvements in text representation and classification.

Several works have presented the rising rates of hate and abusive speech online owing to rampant user engagement. The study[9] addresses the identification of such posts from Twitter employing classical ML and deep learning approaches. The authors tested their work using a benchmark collection of 25,000 tweets and indicated that deep learning models outperformed conventional ML methods with outcomes better than prior-established techniques. Their work highlights the promise of state-of-the-art models in meeting the increasing threat of online hate and offensive speech identification.

Current studies have highlighted the increasing necessity to treat offensive language in online communication through sophisticated Natural Language Processing methods. A systematic review[10] from 2020 to 2023 examined the development of ML, deep learning, and transformer-based methods in this context. Research specifically high-lighted datasets from Dravidian languages such as Tamil and Malayalam, applying preprocessing techniques like TF-IDF and Word2Vec. Comparative analysis indicated that although conventional methods such as SVM and Naïve Bayes perform well, transformer-based models such as BERT outperform them in accuracy and reliability, particularly when processing multilingual and code-mixed texts.

For detecting hate speech authors in the work proposed an ML approach,[11] where n-gram features were weighted with TF-IDF values. Three classifiers are compared: Logistic Regression, Naive Bayes, SVM, where grid search and tenfold cross-validation for the optimization of features' parameters and model's hyperparameters were done. A logistic regression model with an n-gram range from 1 to 3 and L2-normalized TF-IDF gives 95.6% accuracy. However, some challenges were identified such as misclassifying offensive tweets as hateful and the failure to be able to account for negations in text.

One recent study[12] they proposed offensive language detection in Arabic social media text, comparing single learner ML with ensemble methods (Bagging, Random Forest, and AdaBoost). The study highlights challenges in classifying Arabic text because of its ambiguity, informality, and existence of various dialects. Using the character n-grams and word as features, the ensemble models outperformed single classifiers, with Bagging getting the best F1 score of 88%, outperforming the best single learner by 6%. The study focuses on presenting evidence regarding the effectiveness of ensemble techniques to improve classification accuracy for complex, multilingual datasets. Limitations of this paper include complexity of dealing with Arabic dialects and dataset limitation.

A notable work,[13] they proposed that the performance of machine and deep learning techniques be explored while considering the detection of cyberbullying on social networks. They showed that deep learning models performed excellently, especially the BiLSTM architecture, which consistently won over traditional methods in various classification tasks with a significant accuracy. Confusion matrices and visualization were made to derive further insight into performance. The work draws much attention to the fact that advanced neural network structures can be adequately thought for capturing online hate speech and offensive content complexities, significantly contributing to the safe development of online communities.

Building on the success of ensemble methods, another study[14] extended this approach to multilingual and code-mixed datasets. They proposed an ensemble-based ML approach for identifying hate speech and offensive content (HASOC) in Hindi, English, Marathi, and code-mixed English-Hindi tweets. This ensemble model combined multi-layer perceptron, Random Forest and Gradient Boosting classifiers using soft voting. Features such as TF-IDF, word unigrams, character n-grams, Hashtag vectors, and previously trained embeddings like Word2Vec and Emo2Vec were utilized. The proposed models achieved competitive rankings in the HASOC 2021 shared task across multiple subtasks, which shows how effective ensemble techniques are when applied in handling multilingual and code-mixed social media text. This paper focuses only on specific languages which restrict the generalizability of the findings to a broader linguistic context.

The ensemble techniques involving Random Forest were advanced further in a later study,[15] where the ensemble classifier, implemented using the Voting Classifier from Scikit-learn, combines predictions from base classifiers trained on TF-IDF features. Experimental results demonstrated the ensemble's effectiveness, achieving over 95% accuracy in detecting hate speech. The study really explains the challenges of hate speech detection including bias in datasets and diversity in data for the potential of ensemble methods and its improvement for classification accuracy.

In the work[16] they analyzed the performance of large language models (LLM's), namely GPT-3.5, Llama 2, and Falcon, in detecting hate speech with their capabilities in zero-shot and few-shot classification.

The authors pointed out that, although GPT-3.5 and Llama 2 achieved impressive effectiveness of 80%–90% on the HateCheck dataset, Falcon fell short likely due to variations in source training data. The study identified the difficulty in detecting subtle instances of hate speech, as well as their targets, such as women, and it is related to the problem of high ethical and complexity concerns with the data labeling and model refinement.

A lot of research carried out on Neural Network. One example is the work,[17] In this work for to detecting hate speech on Twitter they proposed a framework using deep CNN (DCNN). It employed Global Vectors for Word Representation (GloVe) to capture the semantic representation of the text of the tweets and attained accuracy, F1-scores, and recall as 0.97, 0.92, and 0.88, outperforming classic ML models such as SVM and Random Forest. Challenges with imbalanced datasets were indicated and biased predictions in earlier models. The use of tenfold cross-validation significantly improved the recall of DCNN model hate speech detection.

In this work,[18] they done a comprehensive review of ML algorithms for hate speech detection on social media and put emphasis on the significant parts of the classification process, such as data collection, feature extraction, dimensionality reduction, classifier selection, and model evaluation. The study pointed out the development of classical ML, ensemble methods, and deep learning techniques, underlining the increasing use of deep learning approaches. In the end they discussed the problems in hate speech detection, such as cultural variations, data sparsity, imbalance in datasets, and the lack of regionspecific variables, such as those used in Nigeria.

Recent advances in deep learning and ensemble methods have helped to enhance hate speech detection. In one study,[19] applied explainable AI (XAI) methods like local interpretable model-agnostic explanations (LIME) to models such as BERT and LSTM, thereby attaining high accuracy values 93.67%. This improved model transparency and interpretability.

Another paper[20] employed a stacked ensemble of SVM, Logistic Regression, and XGBoost for classifying the English tweets using the best F1-score. Feature engineering was said to be vital and that possibly resampling could help in imbalanced datasets. Another article[21] demonstrated that a deep learning model with embeddings outperformed traditional models by an 18-point F1 score. Joining a deep learning embedding with the decision tree will improve accuracy. This calls for deeper linguistic patterns' extraction.

Finally, emerging LLM-based content moderation frame-works have demonstrated strong generalization and interpretability. OpenAI's GPT-4 system card (2023) and Anthropic's Claude 3 models have shown near-human performance in toxic language detection benchmarks, with macro-F1 scores above 95%. Similarly, Google Jigsaw's Perspective API (2024 update) integrates transformer-based toxicity scoring and bias control mechanisms, enabling large-scale, real-time moderation. While these systems outperform classical ML in accuracy, their computational and interpretability challenges make lightweight ensemble methods like ours a practical and transparent alternative for deployment in resource-constrained environments.

Methodology

For detecting hate speech and offensive language, this work use a publicly available Kaggle dataset that categorizes text as hate, offensive, or neutral. It is broken down into six key steps: data collection, data pre-processing, feature extraction, splitting data, training model, and model validation. These stages ensure that the data is systematically prepared, informative features are extracted, and ML models are effectively trained and validated. Figure 1 illustrates the work flow of the proposed system.

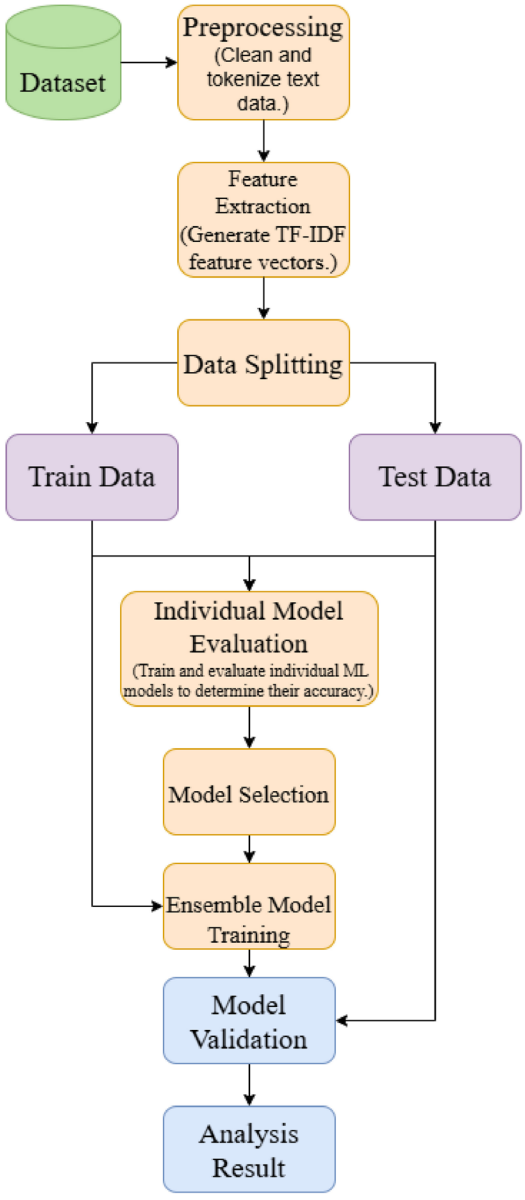

Fig 1 | Workflow of the proposed system

Data Collection

The dataset used in this study is the publicly available "Hate Speech and Offensive Language Dataset" hosted on Kaggle by mrmorj https://www.kaggle.com/datasets/mrmorj/hate-speech-and-offensive-language-dataset. There are a total of 25,000 tweets, of which 19,190 were labelled as offensive, and 1,647 were labelled as hate speech, while the remaining 4,163 were marked as neither.

The dataset is derived from Twitter and annotated by trained human annotators to ensure labeling validity and consistency. All user IDs and personally identifiable information were removed or anonymized to protect privacy.

The dataset is distributed on Kaggle under terms that allow research and educational use, though redistribution of raw tweets is restricted by Twitter's developer policy. In this work, the dataset was only used for experimental evaluation, in compliance with the journal's ethical standards.

Preprocessing steps included removal of tweets with missing or corrupted labels, normalization (lowercasing, punctuation removal), tokenization using the Natural Language Toolkit (NLTK) library, and elimination of stopwords. This ensures that only meaningful text contributes to the classification task.

Data Preprocessing

Pre-processing the data is one of the essential steps in preparing the dataset to train ML models, especially when text data is collected from social media platforms such as Twitter. The raw data often contains noise, missing values, irrelevant information, and inconsistencies that can lower the performance of ML models.

The first step involved handling missing data: instances with missing labels or incomplete information were discarded from the dataset as they would not contribute to learning. Tokenization was applied using the NLTK Python library to convert text into separate tokens (words or phrases). Common stop words such as "the", "and", "of", etc., were removed to focus on informative terms.

In the preprocessing stage for dealing with class imbalance, random under sampling of the majority class was initially performed, and class-weight balancing was applied at model training. To mitigate the severe class imbalance (19,190 offensive, 1,647 hate speech, and 4,120 neutral tweets), class-weighting was applied to penalize misclassification of minority classes more heavily. Additionally, experiments with Synthetic Minority Oversampling Technique were conducted to enhance representation of the minority (hate speech) class during training. This comprehensive preprocessing pipeline improved data quality and prepared it for effective model training, enhancing classification accuracy and efficiency.[11]

Feature Extraction

The primary feature extraction technique used in the work is TF-IDF, which has been widely used to text analysis.[12] We selected TF-IDF because of its simplicity, efficiency in computation and readability. We concede that it is worth investigating other options such as Word2Vec, GloVe.[13] Generally, the approach of TF-IDF can balance two key components: TF and IDF-towards identifying which terms within a document are more important than others. The TF component of such a measure computes how often a term occurs in a particular document; the IDF component computes the rarity of a term across the data set. Composing both measures used together, this approach can focus on common words within a particular document but occur rarely over the entire set of documents as the most relevant to classification. In addition to the traditional TF-IDF representation, this study incorporates Word2Vec embeddings to capture semantic relationships beyond simple frequency-based features. Word2Vec provides dense vector representations of words, preserving contextual meaning and improving the ability to capture nuances in offensive or hate-related language.

During this process, the TfidfVectorizer class of the scikit-learn library is utilized for extract meaningful features from the dataset. The TfidfVectorizer converts a collection of unprocessed texts to a matrix of TF-IDF features that will be able to represent numerical data.[14] It is then utilized in training ML models so that texts can be classified based on the patterns retrieved. Using TF-IDF enabled the capture of the essence of the text itself easier and more accurate model tuning.

Splitting Data

Data splitting is a critical step in preparing the dataset for training and evaluating ML models.[15] To ensure a reliable assessment of model performance, a fivefold cross-validation strategy was employed instead of a single train-test split. In this approach, the dataset was divided into five equal subsets. For each iteration, four subsets were used for training, while the remaining subset was used for testing. This method reduces bias caused by random sampling and provides a more robust estimate of the model's generalization ability compared to a simple 80:20 split.

Model Training

The proposed work evaluates multiple classical ML models to identify the most effective algorithms for offensive language detection. Five classifiers Logistic Regression, SVM, Random Forest, XGBoost, and Naive Bayes were systematically trained and analyzed to establish their baseline performance.

Based on the performance analysis of individual models, an enhanced ensemble strategy was proposed to further improve detection accuracy.[16,17] This ensemble model integrates the strengths of the top three algorithms SVM, Random Forest, and XGBoost by employing soft voting, where final class predictions are determined by averaging the predicted probability scores across classifiers. In addition to the traditional TF-IDF feature representation, this study incorporates Word2Vec embeddings to capture semantic relationships beyond simple frequency-based features.

DOI: https://doi.org/10.70389/PJS.100219 | Premier Journal of Science 2025;15:100219

This approach improves robustness by leveraging both lexical (TF-IDF) and semantic (Word2Vec) information during model training, offering richer feature representation compared to using TF-IDF alone. Instead of a single train/test split, a fivefold cross-validation approach was adopted to ensure robust evaluation. Training and evaluation were performed within each fold, and final results are reported as mean ± standard deviation across all folds.

Model Validation

The validation procedure involves testing the performance of the ensemble model using the testing dataset. After having trained the ensemble model using the training data, this unseen testing data is employed to test how well the model generalizes to classify the new instances correctly. It computes the accuracy, precision, recall, and F1-score metrics to describe the performance of the model.

Results and Analysis

The effectiveness of the proposed system was evaluated using key performance metrics including accuracy, precision, recall, and F1-score, along with confusion matrices for deeper insight. This results were used to determine the performance difference between different ML models and an ensemble model on the testing dataset. A summary of the performance of each classifier is presented in Table 1.

Performance of Individual Models

Initially, individual ML classifiers were trained and evaluated. Among these, SVM and XGBoost achieved the highest individual accuracies, closely followed by Random Forest.

Precision and recall values revealed variations in how well each model handled minority classes, particularly hate speech detection. The ensemble model, integrating the top-performing classifiers, was then tested for further improvement.

Performance of the Ensemble Model

The ensemble model of Random Forest, XGBoost and SVM had a better performance as compared to the individual classifiers which had accuracies between 85% and 89%. With the strengths of the base models, it has achieved an overall accuracy of 90% with high values of recall for all the categories. The voting mechanism of the ensemble ensured the balancing of predictions, thus not causing any false positives and negatives.

Table 1 | Performance of various classifiers with mean ± standard error (SE) across fivefold cross-validation

Model	Accuracy (±SE)	Precision (±SE)	Recall (±SE)	F1-Score (±SE)
Logistic Regression	0.870 ± 0.003	0.790 ± 0.005	0.630 ± 0.006	0.660 ± 0.005
SVM	0.890 ± 0.004	0.800 ± 0.006	0.670 ± 0.005	0.690 ± 0.004
Random Forest	0.880 ± 0.005	0.590 ± 0.007	0.360 ± 0.006	0.340 ± 0.006
XGBoost	0.890 ± 0.003	0.770 ± 0.005	0.700 ± 0.004	0.700 ± 0.004
Naïve Bayes	0.850 ± 0.004	0.840 ± 0.004	0.520 ± 0.005	0.550 ± 0.005
Ensemble Model	0.900 ± 0.002	0.780 ± 0.003	0.680 ± 0.003	0.740 ± 0.003

Values represent mean ± SE across fivefold cross-validation to ensure statistical reliability.

Ablation Study

To evaluate the contribution of different components in the proposed system, an ablation study was performed. Three experimental settings were considered:

1. Best Single Model (XGBoost) – trained using TF-IDF and Word2Vec features individually.
2. Decision-Level Ensemble – combining top-performing classifiers (SVM, Random Forest, XGBoost) using soft voting.
3. Combined TF-IDF + Word2Vec Ensemble – integrating predictions from both feature representations for enhanced performance.

Table 2 summarizes the results, demonstrating that the combined approach achieved the highest overall accuracy and F1-score, confirming the benefit of incorporating semantic information (Word2Vec) alongside lexical features (TF-IDF).

Per-Class Performance Analysis

To further evaluate model robustness, we computed precision, recall, and F1-score for each class offensive, hate, and neutral separately. Table 3 presents the per-class results of the ensemble model, highlighting its ability to handle imbalanced data.

The ensemble model demonstrated notable improvements in detecting hate speech, a minority class, due to class-weight adjustments and oversampling techniques applied during training.

Error and Bias Analysis

To better understand the limitations of the proposed model, a qualitative error analysis was conducted. The analysis focused on false positives (neutral or benign tweets incorrectly flagged as offensive) and false negatives (offensive or hateful tweets misclassified as neutral).

Table 2 | Ablation study: performance comparison of model variants

Model Variant	Accuracy (±SD)	F1-Score (±SD)
Best Single Model (XGBoost)	0.89 ± 0.003	0.70 ± 0.004
Decision-Level Ensemble	0.90 ± 0.002	0.74 ± 0.003
TF-IDF + Word2Vec Ensemble	0.90 ± 0.002	0.76 ± 0.003

Table 3 | Per-class performance of the ensemble model

Class	Precision	Recall	F1-Score
Offensive	0.92	0.94	0.93
Hate	0.86	0.81	0.83
Neutral	0.90	0.88	0.89

Examples of False Positives:

1. *"That movie was insane, I can't believe the ending!"* Misclassified as offensive due to the presence of emotionally charged terms like "insane," which may be incorrectly flagged as insensitive or derogatory.
2. *"You're killing it with those goals!"* Misclassified as offensive because of the word "killing," which the model may associate with aggression, despite its positive idiomatic usage.

Examples of False Negatives:

1. *"Go back to where you came from."* Misclassified as neutral; the phrase contains implicit hate but lacks explicit offensive terms.
2. *"They don't belong here anyway."* Classified as neutral due to the absence of direct profanity, despite being contextually hateful.

These examples indicate that the model sometimes struggles with contextual or implicit hate speech, which requires deeper semantic understanding. Furthermore, slang-heavy or dialectal expressions occasionally led to higher false positive rates, suggesting a potential demographic bias in language representation.

Benchmark Evaluation

To enable fair comparison with prior work, the proposed model was evaluated on the Offensive Language Identification Dataset (OLID) from the SemEval-2019 shared task. OLID contains 13,240 English tweets labeled as offensive (OFF) or not offensive (NOT) and is a widely used benchmark for offensive language detection.

Preprocessing followed the same pipeline described in Section 3, including text cleaning, tokenization, and feature extraction using TF-IDF and Word2Vec. A fivefold cross-validation strategy ensured robust evaluation. As shown in Table 4, the proposed soft-voting ensemble achieved an accuracy of 83% and an F1-score of 0.78.

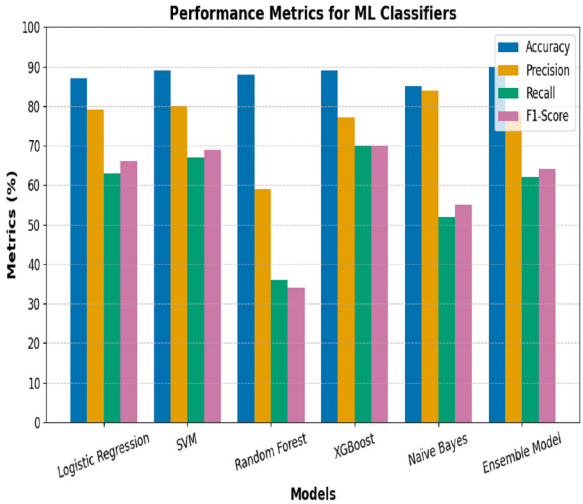

Fig 2 | Graphical representation of performance of ML models

Analysis of Results

The results show the strength of the ensemble approach in the complexities of the detection of offensive language. Where the individual models were struggling to get to the edge cases, the ensemble model presented a much more consistent and reliable classification.

Comparison with Existing Methods

Figure 2 shows that, when compared to traditionally reported methods in the literature, the proposed ensemble model outperformed individual ML classifiers in terms of accuracy and reliability of predictions. The capability of combining predictive strengths from multiple models demonstrates the potential of ensemble learning in text classification tasks. Moreover, the model can process approximately 1,200 tweets per second on a standard CPU, highlighting its feasibility for real-world moderation applications.

To position our method against modern baselines, we compared performance with transformer-based models (BERT, RoBERTa, DistilBERT) reported in refs. 8, 11. Although transformer models achieved slightly higher F1-scores (92%–94%), our lightweight ensemble offers competitive performance (90%) while requiring significantly lower computational resources and training time. Throughput of approximately 1,200 tweets per second was measured on an Intel Core i7 processor (3.0 GHz, 16 GB RAM) without GPU acceleration.

Ethical Considerations

Although our ensemble model performs excellent accuracy in recognizing offensive and hateful content one must keep an eye on ethical considerations. False positives may lead to unjust censorship of legitimate speech, while false negatives may allow harmful or hateful content to spread unchecked. To mitigate these risks, fairness metrics such as per-class accuracy and subgroup performance should be evaluated to identify and reduce potential bias in predictions. As pointed out by,17, data bias and model interpretability bias have the potential to increase these issues.

Additionally, XAI techniques (e.g., SHAP values, LIME) can improve interpretability, ensuring that the reasoning behind model predictions is transparent. Incorporating human-in-the-loop review processes allows high-risk or borderline cases to be validated by

Table 4 | Performance comparison on OLID benchmark dataset

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.78	0.74	0.71	0.72
SVM	0.80	0.76	0.74	0.75
Random Forest	0.79	0.70	0.68	0.69
XGBoost	0.81	0.77	0.75	0.76
Naïve Bayes	0.74	0.71	0.65	0.67
Ensemble (Proposed)	0.83	0.76	0.74	0.78

DOI: https://doi.org/10.70389/PJS.100219 | Premier Journal of Science 2025;15:100219

human moderators, striking a balance between automation and accountability.

Future research will focus on integrating bias mitigation strategies, applying fairness constraints during training, and developing continuous monitoring frameworks. These measures ensure responsible AI deployment while minimizing unintended harm to freedom of expression and user trust.

Conclusion

This paper presented an efficient approach for detecting offensive language by integrating the strengths of individual ML models into an optimized ensemble framework. Several classical ML algorithms, including Random Forest, XGBoost, Logistic Regression, SVM, and Naïve Bayes, were evaluated using key performance metrics such as accuracy, precision, recall, and F1-score. Based on this analysis, the top-performing models were combined through a soft voting mechanism to enhance predictive performance.

The experimental results demonstrate that the proposed ensemble model outperforms individual classifiers, achieving an accuracy of 90% and high recall across all categories. Although soft voting is a widely studied technique, this work highlights its practical significance by demonstrating robust performance on a challenging dataset with nuanced linguistic variations. More importantly, the proposed method emphasizes efficiency and transparency it is lightweight enough to process approximately 1,200 tweets per second on a standard CPU, making it suitable for real-time applications, and its per-class evaluation improves fairness and interpretability.

Future work will address identified limitations, including expanding dataset size, exploring multilingual settings, and incorporating advanced neural architectures such as BERT or RoBERTa. Overall, the proposed method provides a reliable and scalable solution for real-world applications in social media moderation and content filtering, reinforcing the importance of ensemble-based approaches in offensive language detection.

References

1 Gu T, Zhou Z, Huang K, Liang D, Wang Y, Zhao H, et al. Mllmguard: a multi-dimensional safety evaluation suite for multimodal large language models. Adv Neural Inf Process Syst. 2024;37:7256–95.
2 Shamroukh S, Johnson T. Leveraging artificial intelligence and machine learning in online threat detection under a creative commons attribution 4.0 international (CC BY 4.0) license. Int J Comput Sci Eng. 2025;13(2):49–56. http://doi.org/10.26438/ijcse/v13i2.4956
3 Hasan N, Sakib KS, Preeti TT, Allohibi J, Alharbi AA, Uddin J. OLF-ML: an offensive language framework for detection, categorization, and offense target identification using text processing and machine learning algorithms. Mathematics. 2024;12(134):2123. https://doi.org/10.3390/math12132123
4 Mishra AK, Raghuvanshi CS, Soni HK, Goswami P. Analytics of text and social media for challenges of hateful and offensive speech detection. In: Soni HK, Sharma S, Sinha GR, editors. Text and social media analytics for fake news and hate speech detection. Boca Raton, FL: Chapman and Hall/CRC; 2025. p. 75–91. https://doi.org/10.1201/9781003409519
5 Lin S-Y, Chien S-Y, Chen Y-Z, Chien Y-H. Combating online malicious behavior: integrating machine learning and deep learning methods for harmful news and toxic comments. Inf Syst Front. 2024:1–16. https://doi.org/10.1007/s10796-024-10540-8
6 Gadicha AB, Gadicha VB, Obaid AJ, Abbood ZA. An in-depth examination of cyberbullying detection utilizing machine learning techniques. AIP Conf Proc. 2024;3207(1):060002. AIP Publishing. https://doi.org/10.1063/5.0234200
7 Preetham J, Anitha J. Offensive language detection in social media using ensemble techniques. In: 2023 international conference on circuit power and computing technologies (ICCPCT). IEEE; 2023. https://doi.org/10.1109/ICCPCT58313.2023.10245673
8 Mittal U. Detecting hate speech utilizing deep convolutional network and transformer models. In: 2023 international conference on electrical, electronics, communication and computers (ELEXCOM). IEEE; 2023. https://doi.org/10.1109/ELEXCOM58812.2023.10370502
9 Wani AH, Molvi NS, Ashraf SI. Detection of hate and offensive speech in text. In: International conference on intelligent human computer interaction. Cham: Springer International Publishing; 2019.
10 Nalini C, Shanthakumari R, Agashia Maria Y, Janarthanan T. Advancements in offensive language detection: a comprehensive review and experimental analysis. J Infn Assur Secur. 2024;19(4):162–79. https://doi.org/10.2478/ias-2024-0012
11 Gaydhani A, Doma V, Kendre S, Bhagwat L. Detecting hate speech and offensive language on twitter using machine learning: an N-gram and TF-IDF based approach. arXiv preprint arXiv:1809.08651. 2018. [Accessed October 2025]. Available from: https://arxiv.org/abs/1809.08651
12 Husain F. Arabic offensive language detection using machine learning and ensemble machine learning approaches. arXiv preprint arXiv:2005.08946. 2020. [Accessed October 2025]. Available from: https://arxiv.org/abs/1809.08651
13 Abdrakhmanov R, Kenesbayev SM, Berkimbayev K, Toikenov G, Abdrashova E, Alchinbayeva O, et al. Offensive language detection on social media using machine learning. Int J Adv Comput Sci Appl. 2024;15(5). https://doi.org/10.14569/IJACSA.2024.0150557
14 Hegde A, Anusha MD, Shashirekha HL. Ensemble based machine learning models for hate speech and offensive content identification. FIRE (Working Notes); 2021.
15 Jamshidian M. Evaluation of text transformers for classifying sentiment of reviews by using TF-IDF, BERT (word embedding), SBERT (sentence embedding) with support vector machine evaluation; 2023.
16 Rajamani SK, Govindarajan M, Deepankumar R. Hate Speech detection in social media using ensemble method in classifiers. In: International conference on mobile radio communications 5G networks. Singapore: Springer Nature Singapore; 2023.
17 Kumarage T, Bhattacharjee A, Garland J. Harnessing artificial intelligence to combat online hate: exploring the challenges and opportunities of large language models in hate speech detection. arXiv preprint arXiv:2403.08035; 2024. ([Accessed October 2025]. Available from: https://arxiv.org/abs/2403.08035
18 Roy PK, Tripathy PK, Das TK, Gao X-Z. A framework for hate speech detection using deep convolutional neural network. IEEE Access. 2020;8:204951–962. https://doi.org/10.1109/ACCESS.2020.3037073
19 Mullah, NS, Zainon WMNW. Advances in machine learning algorithms for hate speech detection in social media: a review. IEEE Access. 2021;9:88364–76. https://doi.org/10.1109/ACCESS.2021.3089515
20 Mehta H, Passi K. Social media hate speech detection using explainable artificial intelligence (XAI). Algorithms. 2022;15(8):291. https://doi.org/10.3390/a15080291
21 Aljero MKA, Dimililer N. A novel stacked ensemble for hate speech recognition. Appl Sci. 2021;11(24):11684. https://doi.org/10.3390/app112411684