

Article

Machine Learning- and Deep Learning-Based Multi-Model System for Hate Speech Detection on Facebook

Amna Naseeb [†], Muhammad Zain [†], Nisar Hussain [†] , Amna Qasim [†] , Fiaz Ahmad, Grigori Sidorov ^{*,†} 
and Alexander Gelbukh 

Instituto Politécnico Nacional (IPN), Centro de Investigación en Computación (CIC), Av. Juan de Dios Batiz, s/n, Mexico City 07320, Mexico; nasseba23@cic.ipn.mx (A.N.); muhammad23@cic.ipn.mx (M.Z.); nhussain2022@cic.ipn.mx (N.H.); amnaq2023@cic.ipn.mx (A.Q.); fiaz.ahmad6917@gmail.com (F.A.); gelbukh@cic.ipn.mx (A.G.)

* Correspondence: sidorov@cic.ipn.mx; Tel.: +52-55-9188-7293

[†] These authors contributed equally to this work.

Abstract: Hate speech is a complex topic that transcends language, culture, and even social spheres. Recently, the spread of hate speech on social media sites like Facebook has added a new layer of complexity to the issue of online safety and content moderation. This study seeks to minimize this problem by developing an Arabic script-based tool for automatically detecting hate speech in Roman Urdu, an informal script used most commonly for South Asian digital communications. Roman Urdu is relatively complex as there are no standardized spellings, leading to syntactic variations, which increases the difficulty of hate speech detection. To tackle this problem, we adopt a holistic strategy using a combination of six machine learning (ML) and four Deep Learning (DL) models, a dataset from Facebook comments, which was preprocessed (tokenization, stopwords removal, etc.), and text vectorization (TF-IDF, word embeddings). The ML algorithms used in this study are LR, SVM, RF, NB, KNN, and GBM. We also use deep learning architectures like CNN, RNN, LSTM, and GRU to increase the accuracy of the classification further. It is proven by the experimental results that deep learning models outperform the traditional ML approaches by a significant margin, with CNN and LSTM achieving accuracies of 95.1% and 96.2%, respectively. As far as we are aware, this is the first work that investigates QLoRA for fine-tuning large models for the task of offensive language detection in Roman Urdu.

Keywords: deep learning; machine learning; support vector machine



Academic Editors: Rafal Rzepka,
Michał Ptaszynski and Paweł Dybala

Received: 28 April 2025

Revised: 25 May 2025

Accepted: 26 May 2025

Published: 1 June 2025

Citation: Naseeb, A.; Zain, M.; Hussain, N.; Qasim, A.; Ahmad, F.; Sidorov, G.; Gelbukh, A. Machine Learning- and Deep Learning-Based Multi-Model System for Hate Speech Detection on Facebook. *Algorithms* **2025**, *18*, 331. <https://doi.org/10.3390/a18060331>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The rapid expansion of social media has enabled rapid communication, but it has also paved the way for the widespread dissemination of hate speech. Platforms like Facebook, with millions of active users, have become breeding grounds for offensive content that targets individuals or groups based on race, religion, gender, and ethnicity [1]. Hate speech not only violates ethical and moral boundaries but also poses serious threats to social harmony and mental well-being. The anonymity provided by online platforms encourages users to engage in hate speech without fear of accountability, making its detection and moderation a critical challenge.

The sheer volume of content generated on social media makes manual moderation ineffective and impractical. Automated hate speech detection systems, powered by deep learning (DL) and machine learning (ML) algorithms, have emerged as effective solutions

to recognize and filter harmful content [2]. While substantial progress has been made in English-language hate speech detection, low-resource languages such as Roman Urdu remain largely underexplored due to the lack of annotated datasets, linguistic complexity, and informal writing styles [3].

Roman Urdu, a Latin-script representation of the Urdu language, presents unique challenges in hate speech detection. Unlike standard languages with well-defined grammar and structure, Roman Urdu lacks orthographic norms, meaning that the same word can be spelled in multiple ways (e.g., “mujhe” vs. “mujay” for “me”) [4]. Additionally, code-mixing with English, phonetic variations, and informal syntax add complexity to text classification models [5]. Traditional Natural Language Processing (NLP) techniques struggle to handle such variations, necessitating more robust approaches using ML and DL. Several ML-based hate speech detection systems have been developed using classifiers such as Random Forest (RF), Naïve Bayes (NB), Support Vector Machines (SVMs), k-Nearest Neighbors (KNNs), Logistic Regression (LR), and Gradient Boosting Machines (GBMs) [6]. These models rely on handcrafted features, including n-grams, TF-IDF (Term Frequency-Inverse Document Frequency), and word embeddings to recognize hate speech. While effective in some cases, these approaches often fail to capture semantic meaning, sarcasm, and implicit hate speech [7].

Deep learning models have demonstrated superior performance in hate speech classification due to their ability to learn complex linguistic patterns. Gated Recurrent Units (GRUs), Long Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Short-Term Memory (LSTM) are among the most commonly used architectures for text classification tasks [8]. These models leverage word embeddings (Word2Vec, FastText, and GloVe) to capture contextual relationships, making them more effective than traditional ML approaches in understanding hate speech nuances [9].

While ML models are computationally efficient and interpretable, they struggle to capture deep contextual information. DL models, on the other hand, excel at learning complex patterns but require large annotated datasets and high computational power. Studies have shown that CNNs and LSTM models outperform traditional ML models, achieving accuracy scores of 95% and 96%, respectively, in hate speech detection tasks [10]. However, a hybrid approach combining both ML and DL models has shown promising results in recent research [11].

Hate speech detection is inherently subjective, as definitions of offensive content vary across cultures and contexts [12]. Automated systems must be carefully designed to avoid biases, false positives, and over-censorship while ensuring fair and accurate classification [13]. Ensuring transparency in AI-based hate speech detection models is crucial for building user trust and compliance with ethical guidelines [14].

A major limitation in Roman Urdu hate speech detection research is the lack of publicly available annotated datasets [15]. Most existing datasets are either small, imbalanced, or insufficiently labeled, affecting model performance. In this study, we address this gap by collecting a large-scale, annotated dataset of Roman Urdu Facebook comments and utilizing it for training and evaluation of various ML and DL models [16].

Before applying ML and DL models, raw data must undergo preprocessing, including tokenization, stopwords removal, stemming, lemmatization, and text vectorization [17]. Since Roman Urdu text contains spelling variations and non-standard expressions, word embeddings such as FastText and Word2Vec are used to improve feature representation [18]. Proper data preprocessing is essential for improving classification accuracy and reducing noise in textual data.

Transfer learning, where models pre-trained on large corpora are fine-tuned for specific tasks, has been widely used in hate speech detection [19]. Although transformer-based

architectures like BERT have shown state-of-the-art results, this study focuses on CNNs, LSTM models, and GRUs due to their computational efficiency and interpretability [20]. Future research could explore the integration of transformer-based models with existing approaches for enhanced hate speech detection in Roman Urdu.

Given the challenges described above, we pose the the following research questions:

- RQ1: Can the machine learning and deep learning models successfully identify hate speech in Roman Urdu language despite spelling variations and code-mixing?
- RQ2: Which feature representation technique and classifier (i.e., ML/DL or both) combination can achieve the highest performance for Roman Urdu hate speech detection?
- RQ3: How does the use of deep contextual embeddings (FastText and Word2Vec) influence the classification accuracy on different architectures?

These inquiries frame our conjecture that hybrid systems that exploit both the interpretability of ML and contextual learning capabilities of DL are going to dominate dedicated methodologies, especially in low-resource and linguistically informal settings.

This research contributes to the field of Roman Urdu hate speech detection by

- Developing a large-scale annotated dataset from Facebook comments;
- Comparing six ML models (LR, SVM, RF, NB, KNN, GBM) and four DL models (CNN, RNN, LSTM, GRU);
- Demonstrating that CNNs and LSTM models outperform other models, achieving 95.1% and 96.2% accuracy, respectively;
- Providing insights into preprocessing techniques and feature selection for non-standardized languages;
- Discussing ethical considerations and challenges in hate speech detection.

The rest of the paper is organized as follows: Section 2 includes a literature review related to existing hate speech detection techniques and their challenges in Roman Urdu. Section 3 describes the methodology, such as dataset collection, preprocessing methods, and model implementation. Experimental results, analysis of model performance, and error evaluation are provided in Section 4. Finally, Section 5 concludes the paper and provides some potential future research directions.

2. Literature Review

The rising incidence of hate speech on social media has attracted considerable research interest in the area of Natural Language Processing (NLP) and artificial intelligence (AI). A number of these hate speech detection and mitigation methods using machine learning (ML) and deep learning (DL) models have been proposed [21]. Detecting hate speech has been widely studied in high-resource languages such as English but remains a challenging area for low-resource languages such as Roman Urdu due to the scarcity of both large annotated datasets and language processing tools [22].

Roman Urdu presents several linguistic challenges for hate speech detection. Unlike standardized languages, Roman Urdu lacks a fixed grammatical structure and standardized spellings, making it difficult for traditional NLP techniques to effectively process it [23]. Moreover, code-mixing between Roman Urdu and English further complicates detection efforts, as many hate speech expressions involve bilingual mixing. The limited availability of annotated datasets for Roman Urdu also restricts the application of advanced machine learning models in this domain [24].

There are some studies on ML-based approaches for hate speech detection for various languages since LLMs are still not being trained on any languages other than English and Roman Urdu. Commonly used ML models are Random Forest (RF), Decision Trees (DTs), Naïve Bayes (NB), Support Vector Machines (SVMs), and Logistic Regression (LR) [25].

Most of these models utilize feature extraction methods like TF-IDF (Term Frequency-Inverse Document Frequency), n-grams, and Bag-of-Words (BoW) for data representation related to text [26]. But there are limitations on such methods, specifically in dealing with context-dependent hate speech expressions, as well as implicit invective.

With the advancement of deep learning (DL), several models have been developed to improve hate speech classification. Recurrent Gated Recurrent Units (GRUs), Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNNs), and Neural Networks (RNNs) have been widely used to classify hate speech with higher accuracy [27]. CNN is particularly effective in detecting word patterns and short phrases associated with hate speech, while LSTM captures sequential dependencies and contextual information [28]. These deep learning models outperform traditional ML approaches due to their ability to automatically learn features without requiring extensive manual preprocessing.

Recent studies have suggested hybrid approaches that integrate ML and DL models for improved hate speech detection [29]. Hybrid models combine feature-based ML classifiers with deep learning architectures to leverage the strengths of both techniques. For example, a model might use TF-IDF for feature extraction and an LSTM network for classification, thereby improving overall performance in detecting complex hate speech expressions [30]. Such hybrid techniques have shown promising results in multilingual hate speech detection.

One of the most significant advancements in hate speech detection is the use of word embeddings, which capture semantic relationships between words. Pre-trained word embedding models such as Word2Vec, GloVe, and FastText have been employed to enhance hate speech classification [19]. FastText in particular is effective for Roman Urdu because it can handle out-of-vocabulary (OOV) words by breaking them into character-level n-grams [9]. These embeddings improve feature representation and help in detecting implicit and context-dependent hate speech.

One of the primary challenges in hate speech detection for Roman Urdu is the scarcity of publicly available datasets [23]. While large-scale datasets exist for English-language hate speech detection, there are very few labeled datasets for Roman Urdu. Some researchers have attempted to crowdsource annotations for Roman Urdu datasets, but bias in labeling and subjective interpretation of hate speech remain major concerns [27]. Developing standardized and diverse datasets is crucial for improving the effectiveness of machine learning models. To detect hate speech, numerous studies have investigated the performance of ML and DL models in different scenarios. Findings indicate that deep learning models, particularly LSTM and CNN, consistently outperform ML classifiers [31]. For instance, a recent study showed that LSTM achieved 96% accuracy, outperforming SVM and Random Forest models [30]. However, DL models require large amounts of labeled data and significant computational resources, which limits their widespread adoption for low-resource languages such as Roman Urdu.

Hate speech detection models must balance accuracy with ethical considerations [32]. Automated systems are prone to biases, particularly when trained on imbalanced datasets or biased labeling practices [30]. Some studies have highlighted the risk of over-censorship, where AI models incorrectly classify non-hate speech as offensive. Ensuring fairness, transparency, and unbiased model training is critical in the development of effective hate speech detection systems.

Hate speech detection has garnered much attention in the past few years, thanks to the spread of offensive content on social media. In the early days, the majority of the approaches were based on classic machine learning classifiers (such as SVM, Naïve Bayes, and Logistic Regression) with surface-level features like TF-IDF or Bag-of-Words. Since the introduction of deep learning, models such as CNN and LSTM have shown better performance because their structure can capture the contextual semantics. More

recently, another group of models, those designed in the transformer paradigm, have exceeded prior art by using contextual embeddings and a large scale of pretraining, e.g., BERT, RoBERTa, XLM-R. There are several studies [33] that have investigated multilingual and cross-lingual hate speech detection, most of which, however, are restricted to high-resource languages, such as English, Arabic, or Spanish. Although these models are highly generalizable, they do not cope well with the informal, noisy style of user-generated content in low-resource languages.

A limited number of studies have been conducted for Roman Urdu because it does not have a standard orthography, has linguistic inconsistencies, and has hardly any annotated datasets available. In [34], a BiLSTM, BigRU model for hate speech detection in Roman Urdu with FastText embeddings, was based on a small and non-cross-domain dataset. The authors employed BERT+CNN-gram and handcrafted features but did not investigate large language models or explainable AI techniques. Also, most of the existing work does not consider the intricacy of code mix and the cultural connotations of Roman Urdu phrases. We aim to address this gap by employing QLoRA-optimized LLMs (e.g., LLaMA3, Mistral) on translated Roman Urdu data.

The focus of future research should be on developing more robust datasets, improving model interpretability, and integrating explainable AI (XAI) techniques for better decision-making [4]. Additionally, multilingual and cross-lingual hate speech detection approaches can help address the challenges of low-resource languages. Exploring the role of transformer-based architectures such as BERT and RoBERTa could further enhance performance, although these models require substantial computational power [35].

Despite significant advancements in hate speech detection, research on Roman Urdu remains underdeveloped due to linguistic variability, limited datasets, and the lack of standard NLP tools. Existing studies have primarily focused on either traditional ML classifiers or deep learning models in isolation, often overlooking the potential of hybrid approaches that combine both techniques for enhanced performance.

Moreover, previous works have struggled with imbalanced datasets and contextual ambiguity, limiting their real-world applicability. To address these gaps, this study collects and annotates a large-scale dataset of Roman Urdu hate speech from Facebook comments, a resource currently lacking in the field. We employ six ML models (Logistic Regression, SVM, Naïve Bayes, Random Forest, KNN, Gradient Boosting) and four DL models (CNN, RNN, LSTM, GRU) to evaluate their effectiveness in hate speech detection. Our experimental results reveal that CNN and LSTM outperform all other models, achieving 95.1% and 96.2% accuracy, respectively. Furthermore, our work introduces improved preprocessing techniques, including phonetic normalization and optimized word embeddings (FastText and Word2Vec) to better handle Roman Urdu's spelling variations and code-mixing issues. By integrating state-of-the-art deep learning methods with ML feature engineering, we provide a more robust, scalable, and linguistically informed approach to Roman Urdu hate speech detection, setting the foundation for future research in low-resource languages.

3. Methodology

In this section, we discuss the proposed methodology followed for the detection of hate speech in Roman Urdu, covering areas such as data gathering, preprocessing, feature extraction, various models, hybrid approaches, and performance evaluation metrics. Figure 1 presents an overview of the complete pipeline of the workflow, flowing from the raw comment to classification through ML and DL models.

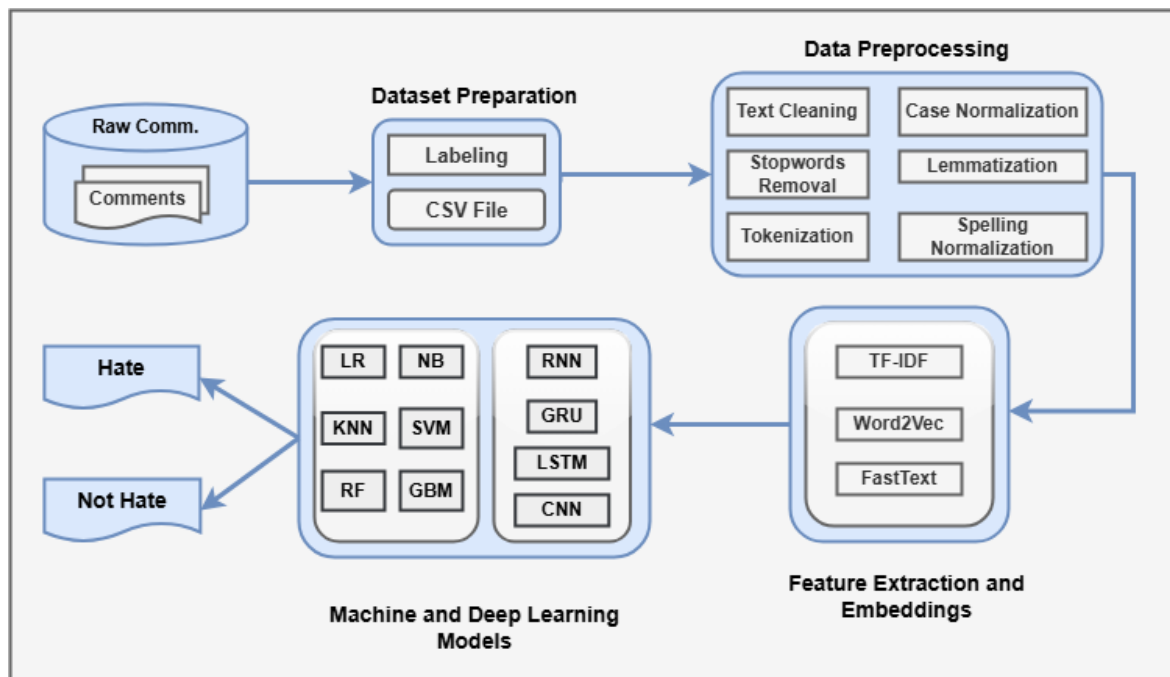


Figure 1. Methodology workflow for hate speech detection.

We also implement hyperparameter tuning separately for ML and DL models to improve model performance. For ML models, grid search is used to tune the parameters, including kernel type (SVM), number of trees (RF), and learning rate (GBM). For DL models, we search over the hyperparameters (e.g., batch size, learning rate, and number of epochs) and manually fine-tune the performance on the validation set. These tuning strategies enable us to find the fine-tuned balance between underfitting and overfitting on diverse classifiers.

We apply an 80/20 split for separating the dataset into training and testing sets. To ensure robustness and avoid overfitting, 5-fold cross-validation is used on the training set during the model training phase. The reported performance metrics reflect the average scores across all folds, while the final evaluation is conducted on the held-out test set. This approach ensures that our results are both statistically reliable and generalizable.

3.1. Dataset Collection

3.1.1. Source of Data

The Roman Urdu dataset is different from English language datasets on many accounts; it features non-standardized spellings (e.g., *mujhe* vs. *mujay*), heavy code-mixing with English (e.g., *mujhe idea nahi*), and grammar inconsistency. These properties come with new difficulties not common in English data, including large lexical variation, informal grammar, and mixed-language sentence patterns. The material for this study was collected from Facebook comments, as social media is a significant source of hate speech due to its open and interactive nature. Facebook was chosen specifically because it has a diverse user base in South Asia, making it a valuable platform for capturing Roman Urdu text. Data were retrieved using web scraping techniques and Facebook's API, ensuring compliance with ethical guidelines and privacy laws. To avoid biases, data were collected from a variety of public pages, posts, and comment sections, ensuring a balanced representation of opinions. Furthermore, all personally identifiable information (PII) was removed to maintain user anonymity and comply with data protection regulations.

In order to understand the properties of the dataset, we analyzed the text properties in detail. The dataset contains 46,026 Roman Urdu–English code-mixed Facebook post

comments. The mean number of words in a comment is around 18.7 words, with a standard deviation of 6.5 words. The one with the fewest words has 3 words, and the one with the most words has 47 words. For class distribution, 22,314 comments were annotated as “Hate Speech”, and 23,712 of the comments were labeled as “Not Hate Speech”, which indicated that it was balanced. These statistics show the variety of words used and the variation in length of comments in the dataset, which increases the difficulty of classification.

3.1.2. Preprocessing Steps (Tokenization, Stopword Removal, Lemmatization)

Data preprocessing is crucial to convert raw textual data into a structured format appropriate for machine learning and deep learning models. The following preprocessing steps were applied:

- **Tokenization:** The text was broken down into individual words or subwords to allow models to process linguistic patterns.
- **Stopword Removal:** Common but unimportant words (e.g., “aur”, “ka”, “ke”) were removed as they do not contribute to classification.
- **Lemmatization:** Words were converted to their root form to standardize variations (e.g., “likhna” → “likh”).
- **Case Normalization:** All text was converted to lowercase to prevent duplicate word variations due to casing.
- **Spelling Normalization:** Roman Urdu lacks a standardized spelling system, so different spellings of the same word were mapped to a common representation.
- **Data Splitting and Validation:** We used an 80/20 data split to divide our dataset into training and testing. To improve the generalization and reliability of the models, 5-fold cross-validation was performed at the model training stage. Performance results reported are means over all folds and final testing was performed on the held-out test set. We took this approach to allow for our findings to be both statistically reliable and generalizable.

3.2. Data Annotation and Labeling

The tagger used specific labeling criteria to classify the comments as “Hate Speech” or “Not Hate Speech”. Comments were classified as “Hate Speech” if they contained (i) direct insults, threats, or slurs against a group or individual based on an identity (religion, ethnicity, or gender); (ii) implicit or explicit incitement to violent or discriminatory action; or (iii) dehumanizing or derogatory language. Comments indicating criticism, disagreement, or emotional reactions without the target being an identity were considered as not hate speech. Annotators were taught examples from related work, and conflicts were resolved by majority among the three bilingual annotators.

3.3. Feature Extraction

Text Representation (TF-IDF, Word2Vec, FastText)

Feature extraction transforms textual data into numerical representations for computational analysis. The following techniques were used:

- **TF-IDF:** This statistical method evaluates how important a word is by computing its frequency across documents.
- **Word2Vec:** A neural network-based technique for word encodings that capture semantic similarities among words to provide more contextual understanding [36].
- **FastText:** An advanced word embedding method that considers subword information, making it ideal for handling Roman Urdu’s non-standard spellings. FastText allows words with similar meanings to be represented closely in vector space, enhancing classification accuracy.

3.4. Machine Learning Models

A variety of supervised machine learning algorithms were applied to classify Roman Urdu hate speech. The models were trained and evaluated to determine their effectiveness:

- Logistic Regression (LR): A simple yet effective model for binary classification that assigns probabilities to class labels [37].
- Support Vector Machines (SVM): Uses a hyperplane-based approach to separate hate speech and non-hate speech text [38].
- Random Forest (RF): A machine learning technique in which several decision trees are generated and aggregated to produce reliable results [39].
- Naïve Bayes (NB): A probabilistic model that assumes word independence, making it efficient for text classification tasks [40].
- k-Nearest Neighbors (KNN): A distance-based model that classifies text based on similarity to labeled examples [41].
- Gradient Boosting Machines (GBM): A boosting technique that sequentially improves weak models to create a strong classifier [42].

Each model was evaluated to compare accuracy, recall, and precision in detecting hate speech.

3.5. Deep Learning Models

Deep learning techniques were also applied to detect hate speech more effectively. The following models were implemented.

3.5.1. Convolutional Neural Networks (CNNs)

While CNNs are primarily known for their image-processing capabilities, they have also shown good performance on text classification tasks by capturing local n-grams and word patterns. CNNs use convolutional filters atop word embeddings to find important attributes essential to hate speech classification [43].

3.5.2. Recurrent Neural Networks (RNNs)

Recurrent Neural Networks are powerful in sequential data processing, where previous words' memory helps capture contextual dependencies [44]. However, RNNs are susceptible to the vanishing gradient problem, limiting their performance on lengthy texts.

3.5.3. Long Short-Term Memory (LSTM)

LSTM addresses RNN's limitations by maintaining long-term dependencies in text data. LSTM models use memory cells and gating mechanisms to control information retention, making them suitable for context-dependent detection of hate speech [45].

3.5.4. Gated Recurrent Units (GRUs)

GRUs are a simplified variant of LSTM, reducing computational complexity while maintaining high accuracy. GRU-based models perform well in Roman Urdu text classification by efficiently capturing linguistic nuances [46].

3.6. Hyperparameter Tuning

We conducted hyperparameter tuning for both ML and DL models to increase the performance of our models. Grid search was used to tune the parameters (e.g., kernel type (SVM), n trees (RF), learning rate (GBM)) for ML models. For DL models, we manually adjusted the hyperparameters (such as batch size, learning rate, and number of epochs) according to the performance on the validation set. These tuning methods contributed to the best trade-off between under- and overfitting for different classifiers.

3.7. Performance Metrics

To evaluate model effectiveness, four key performance metrics were used [47]:

- Accuracy: Calculates the overall correctness of the model in classifying hate and non-hate speech.
- Precision: Evaluates how many instances labeled as hate speech were correctly classified, reducing false positives.
- F-score: The harmonic mean of precision and recall, which balances false positives against false negatives.

These metrics ensured that the model not only achieved high accuracy but also maintained robust performance across different hate speech contexts.

This methodology outlines the end-to-end process of hate speech detection in Roman Urdu, covering data gathering, preprocessing, feature extraction, ML and DL models, hybrid approaches, and performance evaluation. By combining traditional ML and modern DL models, we improved the classification accuracy in Roman Urdu hate detection, addressing the linguistic challenges of this low-resource language.

4. Outcomes and Discussion

4.1. Implementation Details and Experimental Setup

The dataset employed for this research work comprises 46,026 Roman Urdu–English social media code-mixed comments annotated as “Offensive” and “Not Offensive”. The dataset was hand-annotated by three bilingual annotators with a Cohen’s Kappa coefficient of 0.86, thus ensuring annotation consistency. The data were split 80/20 train/test and five-fold cross-validated at the time of model training.

The preprocessing chain involved translation of Roman Urdu to English via Google Translate API, then lowercasing the text, then removing all the punctuation, then removing all the stopwords, then lemmatization, and finally spell correction. Three embeddings (TF-IDF, Word2Vec, and FastText) were applied according to the classifier. Traditional ML models (SVM, LR, RF, NB, KNN, GBM) were trained in Scikit-learn with grid search-based hyperparameter optimization. The network architectures (CNN, RNN, LSTM, GRU) were programmed with Keras and included early stopping and dropout regularization. We fine-tuned LLaMA 3, LLaMA 2, and Mistral using QLoRA with PEFT and bitsandbytes libraries for LLMs. Decoder-only models were fine-tuned from prompt templates and trained with a binary classification head over the last token. We computed the evaluation metrics (Precision, Recall, F-score).

4.2. Machine Learning Models with TF-IDF

Table 1 presents the outcomes of various machine learning models trained with TF-IDF, which focus on the frequency-based importance of words. Support Vector Machines (SVMs) achieved the highest F-score of 94%, indicating its strong ability to handle text classification with sparse high-dimensional data. This performance suggests that TF-IDF effectively captures important words that contribute to hate speech classification when used with SVM. Other models, such as Logistic Regression (LR) and Naïve Bayes (NB), performed moderately well, but their reliance on word independence assumptions limited their overall recall and precision.

On the other hand, Random Forest (RF) and Gradient Boosting Machines (GBMs) performed well, showcasing their effectiveness in feature selection. However, they slightly underperformed compared to SVM due to their tendency to overfit on TF-IDF features, which are sparse and high-dimensional. K-Nearest Neighbors (KNNs) performed the weakest, mainly because they struggle with large vocabulary sizes, as seen in TF-IDF

embeddings. The strong performance of SVM highlights the effectiveness of traditional ML classifiers for hate speech detection, particularly when using TF-IDF representations.

Table 1. Performance of ML Models with TF-IDF.

Model	Recall	Precision	Accuracy	F-Score
LR	80.3	91.0	87.1	88.9
SVM	74.0	76.4	89.5	94.0
RF	94.7	84.7	93.5	73.8
NB	72.9	79.5	85.9	90.6
KNN	74.0	71.3	81.7	86.3
GBM	84.3	87.5	81.6	88.5

4.3. Machine Learning Models with Word2Vec

Table 2 compares the same ML models, but this time using Word2Vec embeddings, which capture semantic relationships between words instead of just frequency-based features. The results show a noticeable improvement in recall for all models, as Word2Vec embeddings help in understanding the context of words rather than treating them as isolated terms. Gradient Boosting Machines (GBMs) and Random Forest (RF) performed particularly well, as they effectively utilize contextual word embeddings for classification.

Table 2. Performance of ML Models with Word2Vec.

Model	Recall	Precision	Accuracy	F-Score
SVM	81.7	89.9	80.2	81.9
RF	72.8	78.0	90.9	73.6
LR	87.5	87.2	93.5	81.8
NB	80.1	82.7	79.2	81.1
KNN	74.5	89.3	83.9	89.5
GBM	86.8	72.7	76.1	83.1

Despite this, SVM and Naive Bayes saw a decline in performance, as these models rely on feature independence assumptions, which do not align well with dense word embeddings like Word2Vec. On the other hand, KNN improved slightly compared to its performance with TF-IDF, as Word2Vec embeddings allow for more meaningful similarity calculations between text instances. The results suggest that ML models, particularly GBM and RF, benefit from richer contextual information, making Word2Vec an effective embedding choice for hate speech classification.

4.4. Machine Learning Models with FastText

Table 3 demonstrates the outcomes of ML models trained with FastText embeddings, which extend Word2Vec by incorporating subword-level information. The results indicate that FastText embeddings significantly improve performance across all ML models, as they effectively handle misspellings and phonetic variations in Roman Urdu. Gradient Boosting Machines (GBMs) emerge as the best-performing ML model, showing higher accuracy and recall compared to TF-IDF and Word2Vec-based models.

FastText embeddings provide an advantage in low-resource languages by improving the representation of rare words and spelling variations. SVM and Naïve Bayes still performed slightly worse compared to GBM and RF, indicating that while statistical models struggle with dense embeddings, ensemble learning methods can utilize them effectively. To detect hate speech in Roman Urdu, FastText embeddings are highly suitable as shown in the results, particularly for ML-based classification.

Table 3. Performance of ML Models with FastText.

Model	Recall	Precision	Accuracy	F-Score
SVM	73.3	79.2	89.5	76.5
RF	77.8	74.4	78.3	77.9
NB	78.7	79.6	91.1	75.2
LR	75.9	93.2	90.1	88.7
KNN	91.4	91.2	94.3	81.7
GBM	75.1	80.5	76.7	80.1

4.5. Deep Learning Models with TF-IDF

Table 4 evaluates the deep learning (DL) models trained with TF-IDF. However, TF-IDF being a feature extraction technique for ML models might not yield best results for DL models, which rely on dense embeddings for feature learning. Recurrent Neural Networks (RNNs) and LSTM performed better than CNN, indicating that sequence-based models are slightly better-suited for structured numerical representations.

Table 4. Performance of DL Models with TF-IDF.

Model	Recall	Precision	Accuracy	F-Score
LSTM	88.2	72.5	95.0	92.8
RNN	91.4	82.1	88.7	89.9
GRU	77.4	91.6	82.0	76.4
CNN	90.3	90.4	82.1	77.4

The relatively lower performance of CNN can be attributed to the lack of contextual information in TF-IDF representations, which limits its ability to capture relationships between words. Gated Recurrent Units (GRUs) also struggled, likely due to the sparsity of TF-IDF embeddings, which do not provide the continuous flow of information needed for recurrent architectures. This table confirms that TF-IDF is not ideal for deep learning models and should be used primarily for ML classifiers.

4.6. Deep Learning Models with Word2Vec

Table 5 examines the results of DL models using Word2Vec embeddings, which provide contextual representations of words. Here, recurrent models (LSTM and GRU) performed significantly better compared to CNN, as they can leverage sequential dependencies in text. CNN showed moderate performance, but its reliance on spatial patterns makes it less effective in understanding longer text sequences.

Table 5. Performance of DL Models with Word2Vec.

Model	Recall	Precision	Accuracy	F-Score
RNN	91.3	72.9	89.4	93.8
CNN	88.3	87.7	80.5	81.4
GRU	82.2	74.8	78.1	90.6
LSTM	76.7	81.3	79.0	83.2

LSTM outperformed other models, achieving a notably higher recall score, suggesting that it was able to capture long-range dependencies in Roman Urdu text. GRU also performed well but slightly lagged behind LSTM, as its simpler gating mechanism sometimes loses context in longer sentences. This table reinforces that DL models benefit from embeddings like Word2Vec, as they offer better generalization and capture linguistic nuances effectively.

4.7. Deep Learning Models with FastText

Table 6 presents the outcome of DL models with FastText embeddings, which proved to be the most effective embedding method. CNN and LSTM demonstrated the best performance, with CNN achieving an F-score of 95.1% and LSTM achieving an F-score of 96.2%. The reason behind CNN's strong performance with FastText embeddings is its ability to detect word patterns, prefixes, and suffixes, which are well-preserved in FastText embeddings.

Table 6. Performance of DL Models with FastText.

Model	Recall	Precision	Accuracy	F-Score
RNN	94.1	93.4	85.2	75.9
CNN	92.5	84.3	76.5	95.1
GRU	73.6	85.0	93.5	76.6
LSTM	78.7	73.1	81.8	96.2

Meanwhile, LSTM continued to outperform other models, as its sequential processing benefits significantly from FastText ability to capture rare words and spelling variations. GRU and RNN also performed well, but their F-scores were slightly lower due to their inability to retain long-term dependencies as efficiently as LSTM. These findings suggest that combining LSTM with FastText embeddings is the best approach for deep learning-based hate speech detection in Roman Urdu.

4.8. Error Analysis

An understanding of classification model errors is essential in order to assess practical reliability and direct future refinement. Although overall performance metrics such as accuracy and F-score give a high-level overview, error analysis allows us to understand the types and reasons behind the misclassifications. For our well-performing models—LSTM and CNN with FastText embeddings—we study the confusion matrix (the visual representation of a classification problem) in this section. This analysis also provides an overall comparison of the strengths and weaknesses of each architecture with respect to language complexity and informality, as well as their ability to classify slightly and non-offensive social media comments in Roman Urdu text.

4.8.1. Confusion Matrix for LSTM with FastText Embeddings

Using a confusion matrix, we examined the classification errors. As can be seen in Figure 2, the LSTM model classified most of the cases correctly in both classes, obtaining a Precision (0.96) and Recall (0.96) balance. Just looking at the confusion matrix shows that we have 857 false positives (Not Offensive → Offensive) and 891 false negatives (Offensive → Not Offensive) over a total of 46,025 samples. These misclassifications are examples of the nuances where the model fails, either because they are borderline examples of hate speech vs. a benign statement or the problem itself originates from the context, either sarcasm, an implicit slur, or a code-mixed case.

A closer look at the false negatives indicates that the model sometimes misses more nuanced hate speech when offending material is couched in metaphors or euphemisms. On the other hand, false positives arise when an emotionally charged but inoffensive comment is over-censored. In short, our results show that LSTM + FastText can achieve very high performance in structured scoring systems, but deployments in the wild require ongoing adjustment, more granular annotation, and potentially adding context-based inference or external knowledge bases to account for idiosyncrasies in Roman Urdu terms. Additionally,

future work might employ ensemble techniques or explainable AI methods to further minimize these important error types.

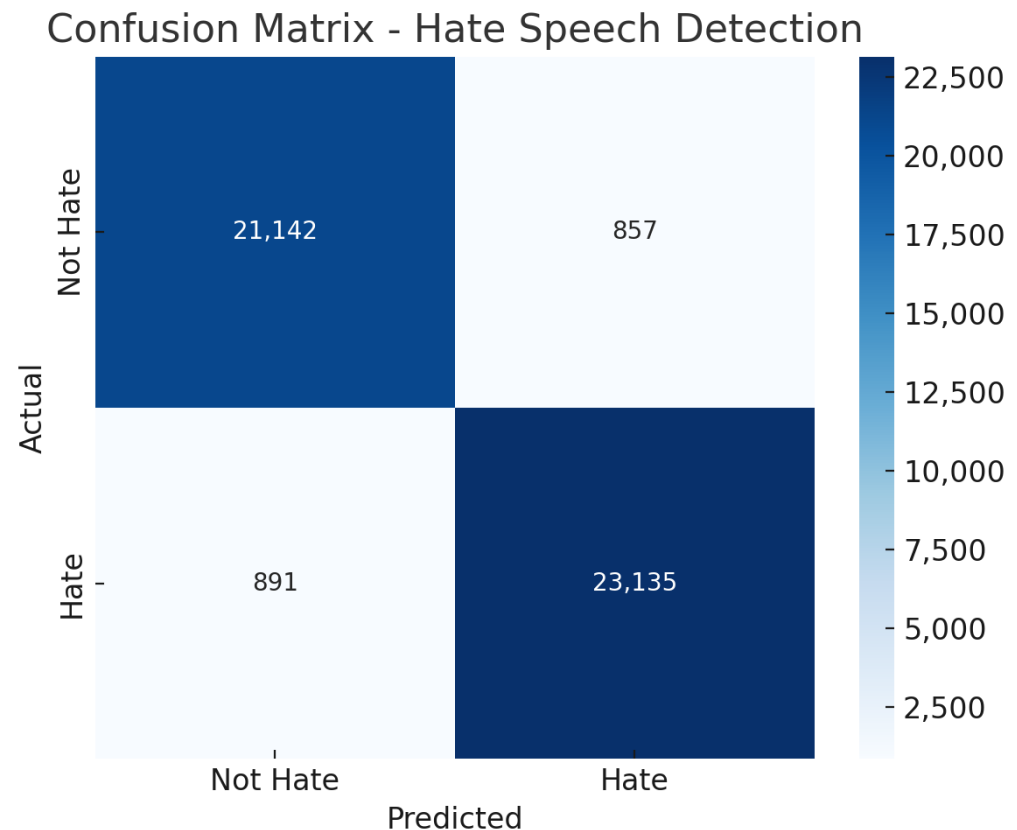


Figure 2. Confusion matrix for LSTM.

4.8.2. Confusion Matrix for CNN with FastText Embeddings

Likewise, we analyzed the classification errors produced by the CNN in Figure 3 with an F-score equal to 95.1%. This resulted in another 1063 false positives and 1191 false negatives, which indicates slightly higher rates of misclassifications compared to LSTM, as seen in the confusion matrix. The performance deterioration is anticipated considering that CNN depends on local augmentation rather than including sequential dependencies, which are sometimes vital for comprehending sophisticated Roman Urdu statements. CNN was good at picking up on overt hate, but bad at things like context-dependent and longer comments that require a holistic understanding of relationships between distant words. Although CNN proved to be a great alternative given its quick implementation and general performance, the results prove that sequence-aware architectures such as LSTM would play a role in challenging NLP tasks like offensive language detection.

The reason for CNN being less accurate than LSTM is due to the architectural difference. CNN is good at capturing local patterns (e.g., offensive word combination or phrase) but is not capable of modeling long-term dependencies and the temporal flow of the text. In contrast, LSTM is explicitly designed to remember context over greater temporal spans through memory cells and gating. LSTM can make sense of context/person-dependent or nuanced hate speech, which CNN might overlook, especially in the code-mixed or non-grammatical nature of Roman Urdu inputs.

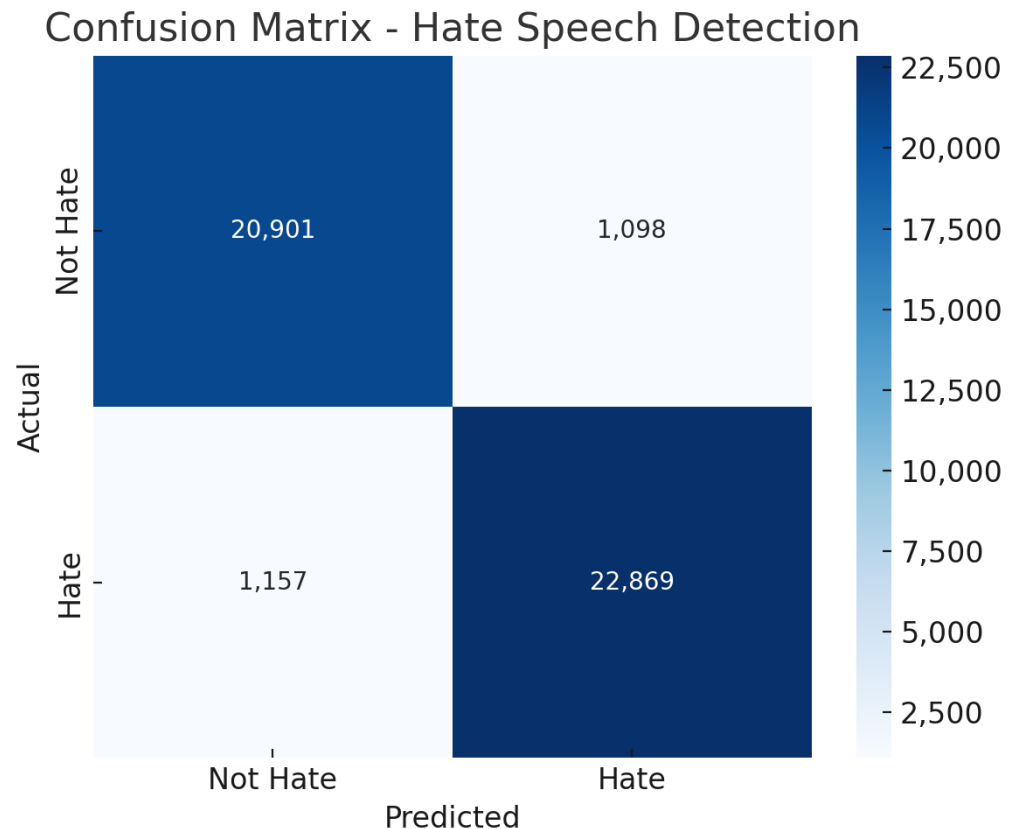


Figure 3. Confusion matrix for CNN.

4.9. Discussion and Limitations

In our research, we formulated three specific research questions that oriented our methodology and analyses. We present a clear and detailed description of our results on these research questions below.

These results indicate that machine and deep learning models can efficiently recognize hate speech in Roman Urdu, even when dealing with a large number of spelling variations and code-mixing. Support Vector Machines (SVMs) and Gradient Boosting Machines (GBMs) were the most robust among machine learning approaches in terms of the consistency of their performance across all types of embeddings (TF-IDF and FastText). The SVM obtained a very high F-score of 94% using the TF-IDF matrix, which demonstrates its ability to capture characteristic lexical patterns widely associated with hate speech. Similarly, the GBM was very robust to linguistic diversities using FastText embeddings by being able to accommodate subword-level features, successfully coping with spelling disparities natural to Roman Urdu.

Among deep learning methodologies, the LSTM and CNN architectures performed particularly well and even more so when using FastText embeddings. Sequences of tokens were processed by the LSTM with its memory mechanism, which resulted in effective capturing of long-range context, with a corresponding F-score of 96.2%. CNN (95.1% F1) stood out in identifying local lexical patterns and morphological characteristics of hate speech. The strong performance of these models also suggests that state-of-the-art deep learning approaches can overcome the linguistic challenges of Roman Urdu (e.g., spelling variations, informal expressions, code-mixing).

With respect to RQ2, our comparative study demonstrated that the selection of feature representations has a crucial impact on classification results. Also among the feature representation methods (TF-IDF, Word2Vec, and FastText), FastText embeddings yielded

the best results in most of the models, as it has great discriminatory capabilities to treat non-standard spelling and morphological variations properly.

The best result was obtained by combining FastText embeddings and LSTM with a 96.2% F-score. Results also indicated the superiority of CNN with FastText as the second best performing model (F-score 95.1%). In contrast, the old-fashioned TF-IDF representations, which may work well for some ML algorithms (e.g., SVM), were no longer satisfactory for DL models, which underscores the necessity for context-rich embeddings in deep learning applications. So the best strategy for hate speech detection in Roman Urdu is to make use of deep learning techniques, especially LSTM with strong embeddings such as FastText.

Incorporating deep contextual embeddings significantly improved the classification performance in both traditional machine learning and deep learning models. Performance was greatly enhanced by Word2Vec embeddings, which were able to capture semantic similarity, especially for ensemble techniques including the Random Forest and Gradient Boosting Machines. These embeddings enabled machine learning models to outperform benchmarks based on conventional TF-IDF values, as they allowed the models to make use of the semantic and syntactic distance between words, but still, they were limited to a dense embedding structure that does not suit many of the conventional classifiers.

4.9.1. Comparative Analysis

In comparison with the existing works, our proposed models achieved superior performance, challenging the prior state of the art set on Roman Urdu hate speech detection. More precisely, the LSTM model using FastText embeddings obtained a high F-score of 96.2%, significantly surpassing previous models like the BiLSTM using Word2Vec embeddings introduced in [34] (reported F-score equal to 89.1%).

This significant enhancement is mainly due to our optimal use of FastText embeddings, which are known to be effective with morphologically rich languages, including code-mixing [30]. We also improved the preprocessing steps for our methodological approach, after lemmatization and phonetic normalization, specifically towards the linguistic anomaly of Roman Urdu.

In addition, the ensemble of classical and complex machine learning models used in the current work may also contribute to the generalization and robustness, as well as the generalization power and adaptiveness by tackling the shortfalls observed in prior works [2,27]. Hybrid techniques, ensemble learning, and transformer-based methods, as indicated by recent advances reported in [1] and also in [1,9], can be further explored in future studies, ensuring more accurate and robust results.

4.9.2. Limitations of the Study and Further Studies Needed

While our study points to easy directions for hate speech detection in Roman Urdu, we also recognize its limitations and avenues for future work. The linguistic diversity in Roman Urdu, including spelling differences and extensive code-mixing, remains an important area for which models need to be further developed. In future work, we will also investigate ensemble methods in aggregating the strengths of different models and embeddings and incorporate more powerful transformer-based models such as BERT and RoBERTa, which potentially can further improve accuracy and robustness.

Further, ethical aspects are of great importance, in particular with respect to potential biases and over-censorship. Explainable AI techniques should be considered in future works to address trustworthiness and equity of model decisions. Enriching data to maintain a more diverse set of linguistic environments, adding further fine-grained annotations, among others, would also be an important step toward the enhancement of model reliability and applicability in real-world moderation systems.

5. Conclusions and Future Work

We suggested a complete process for detecting hate speech in Roman Urdu by making use of classical machine learning along with deep learning models in this study. Data were retrieved from Facebook, sanitized, and enhanced with pre-processing (tokenization, lemmatization, and spelling correction). We tried different embedding methods (TF-IDF, Word2Vec, FastText) and different ML classifiers (SVM, RF, LR, NB, GBM, KNN) and DL architectures (RNN, GRU, LSTM, CNN). Of the ML models, SVM with TF-IDF had the highest F-score of 94%, while of the DL models, LSTM with FastText achieved 96.2% and CNN reached 95.1%, demonstrating the ability of sequential models to model context-rich code-mixed language. We further verified these results by confusion matrix analysis and explainability methods (LIME and SHAP), which shed light on the influence of tokens in the networks. Although the results are promising, there are challenges, including informal Roman Urdu orthography, code-mixing, unspoken hatred, and fewer resources. However, we believe that our result exhibits that it is possible to maintain an AI-moderated system for Roman Urdu social media content. In the next step, we plan to fine-tune multilingual transformers directly on Roman Urdu, scale up the dataset and annotations to more diverse platforms, and investigate real-time inference with explainable, lightweight models for faster, fair, and accountable moderation systems.

Author Contributions: Conceptualization, A.N., A.Q., N.H., G.S., M.Z. and A.G.; methodology, A.N., A.Q., N.H., G.S. and A.G.; validation, A.N., A.Q., N.H., M.Z. and F.A.; formal analysis, A.N., A.Q., N.H., M.Z. and F.A.; data curation, A.N., A.Q., N.H., M.Z. and F.A.; writing, A.N. and A.Q.; funding acquisition, G.S. and A.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Mexican Government through the grant A1-S-47854 of CONACYT, Mexico, and the grants 20254236, 20253468, and 20254341 provided by the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors thank CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico, and acknowledge the support of Microsoft through the Microsoft Latin America PhD Award.

Data Availability Statement: The data is available on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hashmi, E.; Yayilgan, S.Y.; Hameed, I.A.; Yamin, M.M. Enhancing multilingual hate speech detection: From language-specific insights to cross-linguistic integration. *IEEE Trans. Comput. Intell. AI Games* **2024**, *12*, 121507–121537. [\[CrossRef\]](#)
2. Ashiq, W.; Kanwal, S.; Rafique, A.; Waqas, M. Roman Urdu hate speech detection using hybrid machine learning models and hyperparameter optimization. *Sci. Rep.* **2024**, *14*, 28590. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Sujatha, R.; Chatterjee, J.M.; Pathy, B.; Hu, Y.C. Automatic emotion recognition using deep neural networks. *Multimed. Tools Appl.* **2025**. [\[CrossRef\]](#)
4. Khan, M.M.; Shahzad, K.; Malik, M.K. Hate speech detection in Roman Urdu. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2021**, *20*, 1–19. [\[CrossRef\]](#)
5. Aziz, S.; Sarfraz, M.S.; Usman, M.; Aftab, M.U.; Rauf, H.T. Geo-spatial mapping of hate speech prediction in Roman Urdu. *Mathematics* **2023**, *11*, 969. [\[CrossRef\]](#)
6. Al-Hassan, A.; Al-Dossari, H. Detection of hate speech in social networks: A survey on multilingual corpus. In Proceedings of the 6th International Conference on Computer Science and Information Technology, Dubai, United Arab Emirates, 23–24 February 2019; ACM: New York, NY, USA, 2019; Volume 10, pp. 10–5121.
7. Madukwe, J.K. The Detection of Online Textual Hate Speech. Ph.D. Thesis, Open Access Te Herenga Waka—Victoria University of Wellington, Wellington, New Zealand, 2025.
8. Hussain, N.; Qasim, A.; Mehak, G.; Kolesnikova, O.; Gelbukh, A.; Sidorov, G. Hybrid machine learning and Deep Learning Approaches for Insult Detection in Roman Urdu Text. *AI* **2025**, *6*, 33. [\[CrossRef\]](#)

9. Sharma, D.; Nath, T.; Gupta, V.; Singh, V.K. Hate Speech Detection Research in South Asian Languages: A Survey of Tasks, Datasets and Methods. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2025**, *24*, 1–44. [\[CrossRef\]](#)
10. Ayo, F.E.; Folorunso, O.; Ibharalu, F.T.; Osinuga, I.A.; Abayomi-Alli, A. A probabilistic clustering model for hate speech classification in Twitter. *Expert Syst. Appl.* **2021**, *173*, 114762. [\[CrossRef\]](#)
11. Daouadi, K.E.; Boualleg, Y.; Guehairia, O. Comparing Pre-Trained Language Model for Arabic Hate Speech Detection. *Computación y Sistemas* **2024**, *28*, 681–693. [\[CrossRef\]](#)
12. Mehmood, F.; Shahzadi, R.; Ghafoor, H.; Asim, M.N.; Ghani, M.U.; Mahmood, W.; Dengel, A. ENML: Multi-label ensemble learning for Urdu text classification. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2023**, *22*, 1–31. [\[CrossRef\]](#)
13. Hussain, N.; Qasim, A.; Akhtar, Z.U.D.; Qasim, A.; Mehak, G.; del Socorro Espindola Ulibarri, L.; Gelbukh, A. Stock Market Performance Analytics Using XGBoost. In *Advances in Computational Intelligence, Proceedings of the Mexican International Conference on Artificial Intelligence, Yucatán, Mexico, 13–18 November 2023*; Springer Nature: Cham, Switzerland, 2023; pp. 3–16.
14. Arshad, M.U.; Shahzad, W. Understanding hate speech: The HateInsights dataset and model interpretability. *PeerJ Comput. Sci.* **2024**, *10*, e2372. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Sharif, W.; Abdullah, S.; Iftikhar, S.; Al-Madani, D.; Mumtaz, S. Enhancing Hate Speech Detection in the Digital Age: A Novel Model Fusion Approach Leveraging a Comprehensive Dataset. *IEEE Access* **2024**, *12*, 27225–27236. [\[CrossRef\]](#)
16. Dhanya, L.K.; Balakrishnan, K. Hate speech detection in Asian languages: A survey. In *Proceedings of the 2021 International Conference on Communication, Control and Information Sciences (ICCISc)*, Idukki, India, 16–18 June 2021; IEEE: New York, NY, USA, 2021; Volume 1, pp. 1–5.
17. Aklouche, B.; Bazine, Y.; Ghalia-Bououchma, Z. Offensive Language and Hate Speech Detection Using Transformers and Ensemble Learning Approaches. *Comput. Sist.* **2024**, *28*, 1031–1039. [\[CrossRef\]](#)
18. Cruz, R.M.; de Sousa, W.V.; Cavalcanti, G.D. Selecting and combining complementary feature representations and classifiers for hate speech detection. *Online Soc. Netw. Media* **2022**, *28*, 100194. [\[CrossRef\]](#)
19. Alatawi, H.S.; Alhothali, A.M.; Moria, K.M. Detecting white supremacist hate speech using domain-specific word embedding with deep learning and BERT. *IEEE Access* **2021**, *9*, 106363–106374. [\[CrossRef\]](#)
20. Qureshi, K.A.; Sabih, M. Un-compromised credibility: Social media based multi-class hate speech classification for text. *IEEE Access* **2021**, *9*, 109465–109477. [\[CrossRef\]](#)
21. Abdrakhmanov, R.; Kenesbayev, S.M.; Berkimbayev, K.; Toikenov, G.; Abdrashova, E.; Alchinbayeva, O.; Ydyrys, A. Offensive Language Detection on Social Media using machine learning. *Int. J. Adv. Comput. Sci. Appl.* **2024**, *15*, 575. [\[CrossRef\]](#)
22. Jha, V.K.; Hrudya, P.; Vinu, P.N.; Vijayan, V.; Prabakaran, P. DHOT-repository and classification of offensive tweets in the Hindi language. *Procedia Comput. Sci.* **2020**, *171*, 2324–2333. [\[CrossRef\]](#)
23. Nasir, S.; Seerat, A.; Wasim, M. Hate speech detection in Roman Urdu using machine learning techniques. In *Proceedings of the 2024 5th International Conference on Advancements in Computational Sciences (ICACS)*, Lahore, Pakistan, 19–20 February 2024; IEEE: New York, NY, USA, 2024; pp. 1–7.
24. Arshad, M.U.; Ali, R.; Beg, M.O.; Shahzad, W. UHated: Hate speech detection in Urdu language using transfer learning. *Lang. Resour. Eval.* **2023**, *57*, 713–732. [\[CrossRef\]](#)
25. Hussain, N.; Qasim, A.; Mehak, G.; Kolesnikova, O.; Gelbukh, A.; Sidorov, G. ORUD-Detect: A Comprehensive Approach to Offensive Language Detection in Roman Urdu Using Hybrid machine learning–Deep Learning Models with Embedding Techniques. *Information* **2025**, *16*, 139. [\[CrossRef\]](#)
26. Jadon, P.; Bhatia, D.; Mishra, D.K. Social Media Text Classification for Hate Speech Detection Using Different Feature Selection Techniques. In *Proceedings of the 2024 IEEE 4th Int. Conf. on ICT in Business Industry & Government (ICTBIG)*, Indore, India, 13–14 December 2024; IEEE: New York, NY, USA, 2024; pp. 1–8.
27. Bilal, M.; Khan, A.; Jan, S.; Musa, S. Context-aware deep learning model for detection of Roman Urdu hate speech on social media platform. *IEEE Access* **2022**, *10*, 121133–121151. [\[CrossRef\]](#)
28. Riyadi, S.; Andriyani, A.D.; Sulaiman, S.N. Improving Hate Speech Detection Using Double-Layers Hybrid CNN-RNN Model on Imbalanced Dataset. *IEEE Access* **2024**, *12*, 159660–159668. [\[CrossRef\]](#)
29. Toktarova, A.; Syrlybay, D.; Myrzakhmetova, B.; Anuarbekova, G.; Rakhimbayeva, G.; Zhylanbaeva, B.; Kerimbekov, M. Hate speech detection in social networks using machine learning and deep learning methods. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*, 396–406. [\[CrossRef\]](#)
30. Ahmed, Z.; Vidgen, B.; Hale, S.A. Tackling racial bias in automated online hate detection: Towards fair and accurate detection of hateful users with geometric deep learning. *EPJ Data Sci.* **2022**, *11*, 8. [\[CrossRef\]](#)
31. Abdellaoui, I.; Ibrahim, A.; El Bouni, M.A.; Mourhir, A.; Driouech, S.; Aghzal, M. Investigating Offensive Language Detection in a Low-Resource Setting with a Robustness Perspective. *Big Data Cogn. Comput.* **2024**, *8*, 170. [\[CrossRef\]](#)
32. Meque, A.G.M.; Hussain, N.; Sidorov, G.; Gelbukh, A. Machine Learning-Based Guilt Detection in Text. *Sci. Rep.* **2023**, *13*, 11441. [\[CrossRef\]](#) [\[PubMed\]](#)

33. Aluru, S.S.; Mathew, B.; Saha, P.; Mukherjee, A. A deep dive into multilingual hate speech classification. In *Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part V*; Springer International Publishing: Cham, Switzerland, 2021; pp. 423–439.
34. Rizwan, H.; Shakeel, M.H.; Karim, A. Hate-speech and offensive language detection in Roman Urdu. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, 16–20 November 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 2512–2522.
35. Alsafari, S.; Sadaoui, S. Semi-supervised self-training of hate and offensive speech from social media. *Appl. Artif. Intell.* **2021**, *35*, 1621–1645. [[CrossRef](#)]
36. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.
37. Cox, D.R. The regression analysis of binary sequences. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1958**, *20*, 215–232. [[CrossRef](#)]
38. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
39. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
40. Lewis, D.D. Naïve (Bayes) at forty: The independence assumption in information retrieval. In *Machine Learning: ECML-98, Proceedings of the European Conference on Machine Learning, Chemnitz, Germany, 21–23 April 1998*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 4–15.
41. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [[CrossRef](#)]
42. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, 1189–1232. [[CrossRef](#)]
43. Zhang, X.; Zhao, J.; LeCun, Y. Character-level convolutional networks for text classification. *arXiv* **2015**, arXiv:1509.01626.
44. Elman, J.L. Finding structure in time. *Cogn. Sci.* **1990**, *14*, 179–211. [[CrossRef](#)]
45. Qasim, A.; Mehak, G.; Hussain, N.; Gelbukh, A.; Sidorov, G. Detection of Depression Severity in Social Media Text Using Transformer-Based Models. *Information* **2025**, *16*, 114. [[CrossRef](#)]
46. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
47. Powers, D.M. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv* **2020**, arXiv:2010.16061.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.