VFAST Transactions on Software Engineering
https://vfast.org/journals/index.php/VTSE@ 2025, ISSN(e): 2309-3978, ISSN(p): 2411-6246
Volume 13, Number 4, October-December 2025 *pp:200-212*

Multi-class Offensive Language Detection in Roman Urdu

Rida Ayesha [1], **Sarah Ali** [1,2†§], **Usman Inayat** [2*†¶], **Sajid Mahmood** [2*]

[1]Department of Computer Science, University of Management & Technology, Lahore, Pakistan;
[2]Department of Informatics & Systems, University of Management & Technology, Lahore, Pakistan

Keywords: *Hate Speech Detection Cyber Bullying Machine Learning Sentiment Classification Roman Urdu*
Journal Info: *Submitted: October 15, 2025 Accepted: December 20, 2025 Published: December 31, 2025*

Abstract

Automated systems for detecting hate speech play a crucial role in combating the proliferation of hateful content, especially as social media user bases continue to grow. Recent research efforts have focused on creating datasets for this purpose, but the majority have been designed for English, leaving low-resource languages like Roman Urdu with limited resources. To enhance hate speech identification in Roman Urdu, various machine and deep learning models were trained on a publicly available dataset of Roman Urdu tweets (RUSHOLD). For multi-class classification, both machine and deep learning techniques were employed, while restricting binary classification to deep learning methods. Given the dataset's class imbalances, particularly with some classes having fewer instances, SMOTE was employed to address this disparity. The findings indicated that the developed machine learning model outperforms the deep learning model in terms of recall, as well as key metrics such as F1 and Macro F1.

***Correspondence author email address:** usman.inayat@umt.edu.pk
DOI: 10.21015/vtse.v13i4.2251

1 Introduction

One of the key and essential benefits of social media is that it keeps people connected to their loved ones. However, it also has brought along few nuisances, the most thumping of which is the use of offensive and unfriendly language towards a person, community, society or religion, which has a negative impact on social media users.

In order to prevent social media users from hurting the sentiments/feelings of someone else, offensive speech, which is quite prevalent in all social media environments and is topped by cyberbullying, needs to be eliminated or at least minimized. The worst part of this issue is that generally people start bullying an individual or community with whom they do not even have any direct business. It may happen in a number of ways, such as by making hateful comments, using insulting language, or making jokes about the race, colour, or faith of an individual or community.

Although it is against the law to make offensive comments on another person's post, this practise is increasing rapidly. Inappropriate language is being used

in some of the comments, which generally encourages cyberbullying against specific people including politicians, celebrities, and brands etc. Bullying of a certain group, such as those from a particular age group or country, is included in this offence.

The related studies of comparable research to cyber bullying and hate speech are discussed in Section II. Experimental analysis and a brief summary of the suggested system and architecture design are included in Section III. The results of the findings are presented in Section IV. Section V contains the conclusion, discussion, and recommendations for further work.

2 Related Work

Recently, the growing existence of hate speech on social media has become a noteworthy concern and cause of distress. While many researchers have focused on European languages, few have worked on South Asian languages, such as Roman Urdu, which is frequently used in the subcontinent. The extensive review of automatic detection techniques is presented in [1], from classical ML to transformers — including challenges and datasets. Authors in [2] focuses on making hate speech detection models more interpretable, bridging the gap between high accuracy and transparency in predictions — a growing concern in NLP ethics and deployment. In their study, Nasir et al. [3] contribute by developing a methodology to detect hate speech in Roman Urdu at two levels. First, authors classified the social media content into neutral and hostile categories. Secondly, authors classified the hostile content as offensive and hate speech. Authors used a benchmark corpus (HS-RU-20) to evaluate the proposed methodology and presented a two-level classification. Furthermore, authors analyzed the word and character level features along with six supervised learning models (LR, RF, KNN, multinomial bias, SVM, and CNN). In their results, Logistic Regression performed better in terms of accuracy, at 81% on neutral-hostile.

Bilal et al. [4] present a study on detecting hate speech in Roman Urdu on social media platforms. Authors have employed a transformer-based model and introduce a Roman Urdu pre-trained BERT model named as BERT-RU (which was claimed to be the first for Roman Urdu), it is trained on a large dataset. Authors use traditional and deep learning models including LSTM, BiLSTM, BiLSTM + Attention Layer, and CNN for baseline comparisons. Transfer learning is explored by combining pre-trained BERT embeddings with deep learning models. Performance evaluation metrics include accuracy, precision, recall, and F-measure. The transformer-based model outperformed all other models, attaining remarkable results with an accuracy of 96.70%, precision of 97.25%, recall of 96.74%, and an F-measure of 97.89%. Additionally, the transformer based model demonstrates superior generalization on a cross-domain dataset.

Shahid et al. [5] discover the usage of semantic features, word embeddings, and language models to effectively capture contextual representations of violence-related content in Urdu tweets. Their work employs 1-Dimensional Convolutional Neural Network (1D-CNN) to optimize its parameters on a newly proposed annotated Urdu corpus comprising 4808 tweets collected manually from Pakistani Twitter accounts. The 1D-CNN merged with a word uni-gram model was assessed together with Urdu-BERT, Urdu-RoBERTa, fine-tuned Urdu-RoBERTa, BiLSTM, CBi-LSTM, and six other state-of-the-art machine learning models. Amongst these, the 1D-CNN model performed best, attaining 89.84% accuracy and a macro F1-score of 89.80%. It outperformed all other models, with F1-scores of 89.76% for the violence class and 89.84% for the non-violence class.

Anas Ali et al. [6] introduce a model designed for detecting offensive language in Pashto, a low-resource language. A Roman Pashto dataset was created by manually annotating 60,000 comments gathered from several social media sites. The proposed model was then trained and assessed using three feature extraction methods: BoW, TF-IDF, and sequence integer encoding. The training was conducted using four traditional classifiers along with a deep sequence model. Experimental results reveal that the random forest classifier attained the top performance amongst traditional models, attaining a testing accuracy of 94.07% by combining unigrams, bigrams, and trigrams. With

TF-IDF features, the same classifier reached a somewhat lower maximum accuracy of 93.90%. Though, the maximum overall testing accuracy of 97.21% was attained with a BLSTM model. Moreover, the dataset created in this study has been made publicly available to support more study in the field.

[7] et al. present a cyberbullying detection approach precisely designed for examining textual content in Roman Urdu. The technique includes cutting-edge preprocessing practices, ensemble approaches, and machine learning algorithms. A range of features—including statistical features, word N-grams, combined N-grams, and a bag-of-words (BoW) model with TF-IDF weighting—are drawn out by means of GridSearchCV and cross-validation across various experimental settings. The detection approach addresses users' colloquial and non-standard writing styles on social media. Emperical results display that the SVM model, used with embedded hybrid N-gram features, attains the highest average accuracy of about 83%. Amongst the voting-based ensemble approaches, XGBoost achieves best, attaining an accuracy of 79%.

Khan, M.M. et al. [8] attempted to employ six deep learning and machine learning-based methodologies. The probabilistic classification approach Naive Bayes, SVM, the Linear Classifier (Logistic Regression), the Bagging Model (Bootstrap Aggregating), the Boosting Model, and the Deep Neural Network Model (CNN is used). Authors have created a dataset of 100,000 remarks and phrases that are being used frequently on social media platforms in daily life. Authors discovered that the corpus can identify between hostile and neutral data by applying the supervised machine learning approaches previously discussed. This shows that the distribution of data for both classes in the training and testing folds was correct. The best method can be viewed as Logistic Regression using Count Vectors because it has outperformed CNN. Word and character n-grams did not do very well on the produced corpus. Additionally, the first fold had the greatest F1- score overall (0.932), while the ninth fold had the lowest (0.883). The average F1-score of 0.906 demonstrates the potency of this outcome. In order to aid with other languages as well, the authors plan to make the corpus more general than before. For improved performance and learning, the dataset utilised in this corpus needs to be expanded more. To deal with the complex morphology of Urdu, more in-depth analysis can include morphological and syntactic rules, combined with features such as character N-grams, unigrams, and bigrams. In order for the model to better grasp the emotions, it will be important to comprehend and manage smileys in the future. The sole flaw in the work is that the authors ought to have produced more general conclusions using a larger dataset.

Usman et al. [9] presents a trilingual dataset (English, Urdu, Spanish) and evaluates transformer + large language models for multilingual detection. While Basel et. al [10] evaluates LLMs' performance comparative to traditional classifiers, presenting LLMs' enhanced contextual understanding. In addition, Ali, M.Z. [11] employed the Rule-Based Linguistic Approach, which establishes the logistics, syntax, and semantics of a certain language before adding additional rules to more clearly define the meaning of the sentence. Suppose, the entered keyword is "bad." The words "terrible / awful / unsatisfactory" will also be automatically searched for by the language processing system. The machine learning approach creates a mathematical model based on training data, enables the computer to learn, and then, using test data, determines whether the model is a classifier or a regression model. Deep Learning Approach uses a sequence of instances, such as images of dogs or cats, and applies them to a more complex and expansive dataset that enables us to employ neural networks. It recognises and learns all the features from the data that is provided to it. Once it has received enough training, it can label unknown data to test the model. Another successful strategy is the hybrid approach, in which two or more strategies are used to improve the effectiveness and accuracy of the model. To create a good outcome model, the rule-based, machine learning, and deep learning methodologies were integrated. The results show that the Rule-Based Linguistic Approach provides us with the highest accuracy of 96.6%,

while the Supervised Learning Approach provides us with the highest accuracy of 97.19%. Therefore, these methods are the best for detecting hate speech in text. If larger real-time dataset was acquired from several social media platforms, these results may be more effective. Additionally, hate speech is not just found in texts; it may also be found in other forms of contact, such the detection of images and videos.

Researchers Mohiyaddeen, et al.[12] have identified gaps in the literature, such as a lack of properly annotated datasets, a lack of studies comparing the effectiveness of character and word n-gram feature selection approaches, and the sparse use of machine learning techniques for classification in the past. The Bayesian Model, Nearest Neighbor, Tree, Random Tree, Random Forest, Linear Regression, Logistic Regression, Logit boost, and Support Vector Machine are the seven machine learning approaches that the author employed to create a classifier. The authors automatically screened YouTube comments for foul language in both the original Urdu and romanized Urdu. The initial Urdu dataset produced by the authors is by far their biggest contribution. Additionally, authors looked into the usefulness of various n-grams and discovered that character n-grams perform better than word n-grams. Regression was the most effective and efficient of the seven key techniques that was used to examine the outcomes of a total of 17 classifiers. By utilising character trigrams, Logit Boost exhibits greater performance on Roman Urdu and achieves a 99.2% F-measure score. To recognize offensive language in Urdu and Roman Urdu, the author proposed using neural network-based models, like character-level and fully convolutional neural networks. Authors also wanted to use other social media platforms to produce a stronger dataset. The absence of data and study is the main flaw in this work.

To categorise the abusive language in the romanized Urdu dataset, Dewani, A.,et al. [13] The authors used a variety of preprocessing methods to clean up the text, tokenize, remove unnecessary words, map slang and contractions, and classify cyberbullying using RNNs, LSTMs, Bi-LSTMs, and CNNs. Each method produced a different set of results, demonstrating that precision, recall, and f1-score are the metrics that are used to assess these models. When using RNN-LSTM, authors achieve accuracy, recall, and f1-score of 85%, while RNN-Bi-LSTM yields results of 84% for all the chosen measures. CNN's performance was the least successful, yielding precision, recall, and f1-score of 78%, 79%, and 78%, respectively. The authors in [14] emphases on hate detection and target identification in Hindi/Nepali by means of an Attention BiLSTM + XLM-R model.

RUT Corpus (Roman Urdu text Corpus) is a brand-new dataset that Saeed, H.H. [15] generated by gathering harmful comments in roman urdu. To fix the issue, authors first preprocessed the text by lemmatizing, stemming, and changing all of the text to lower case. Authors have utilised cutting-edge embedding models like word2vec and glove embeddings for their embeddings. The CNN-George, BGRU-P, CNN+GRU, recurrent neural networks, and modified CNNs are some examples of the deep learning architectures that have been used to learn multiple representations and classifications of the text. Different hyperparameter tweaking provided various outcomes, such as a maximum f1-score of 83 percent for all ML models and a f1-score of 85 percent for word2vec with deep learning models, but the highest f1-score was achieved by ensemble embeddings with ML+B.Deep(MV) was 86%.

By creating the first-ever substantial labelled corpus of toxic and non-toxic comments, Rizwan, H., et al. [16] tackle the problem of Roman Urdu toxic comment detection. The author's hypothesis was that the gaps authors found could cause their model to malfunction when it comes across some other spellings. Additionally, the model is unfamiliar with the ideograms (symbol representation of words) The pre-processing step's elimination of punctuation marks is a definite factor in this kind of misdiagnosis. The author employed CNN-George, BGRU-P, CNN+GRU, BLSTM, BGRU, and CNN Tweaked approaches. As demonstrated by the results obtained using the prepared corpus, task-specific word vectors outperform embeddings learned via GloVe and Word2Vec's CBOW method. Amongst all learned embeddings, those at-

tained out of FastText's skip-gram model got the best experimental results. For the future, the author claims that this research thoroughly explores an ensemble's performance, stressing traits that it can and cannot reliably detect. The tagged RUT corpus was made accessible to the scholarly community by the author for use in later projects. The authors think that using this resource will result in more accurate hazardous comment identification algorithms for Roman Urdu. The author uses the average accuracy, precision, recall, and F1 values supplied for each model's overall folds to illustrate the model's limitations. Accuracy can deceive due to the dataset's imbalance, hence this study uses F1 as the primary metric of evaluation. F1 is the harmonic mean of precision and recall.

According to Albladi, et. al [17], the rise of social media has amplified communication but also increased the spread of hate speech. Traditional detection methods struggle with context sensitivity, leading to inaccuracies. Usman et. al [18] demonstrates performance gains in social media hate speech detection by augmenting BERT with GPT-2 based augmentation. Recent research highlights the use of LLMs like GPT-3 and BERT in improving hate speech detection, offering better contextual understanding. Ahmad et al. [19] created a multilingual tweet dataset and illustrated that pre-trained transformer models (e.g., XLM-R) perform efficiently across languages. However, challenges such as bias and fairness persist, necessitating further research to enhance accuracy and ethical implementation.

The paper [20]discusses multimodal detection, explain ability, and counternarrative generation. Nguea-jio et. al [21] states that Explainable AI (XAI) plays a crucial role in detecting hate speech and misinformation by improving model transparency and interpret ability. Recent studies highlight the relationship between these issues and explore XAI techniques to mitigate them. The survey reviews state-of-the-art XAI methods, datasets, and evaluation metrics, offering insights into their strengths and limitations. Future research should focus on enhancing model explainability for fairer and more effective detection systems.

2.1 Proposed Methodology

2.2 Dataset Description

The used dataset was created by (Hate-Speech and Offensive Language Detection in Roman Urdu) authors using Twitter API [16]. The name of the dataset is Roman Urdu Hate speech and Offensive Language Detection (RUSHOLD). The dataset in total has 2 columns of text and labels. The text column represents the tweet or text and the label column represents its respective class. Multi-class dataset has five classes which states whether the tweet is normal, abusive, religious hate, sexism and profane. Binary classification dataset has two classes which states whether the tweet is offensive or neutral.

2.2.1 Multi-class RUSHOLD Dataset

From RUSHOLD dataset, multi-class dataset is one of them and it didn't have much results to play with, with the best of the efforts the results are improved and give us decent difference for comparison.

Table 1. The number of instances in respective areas for Multi-class RUSHOLD Dataset

Labels	Training Data	Testing Data	Examples
Offensive	1768	442	2210
Normal	3937	984	4921
Sexism	617	152	771
Religious Hate	575	144	719
Profane/Untargeted	471	118	589
Total	7368	1840	9210

Table 1 displays the total number of dataset entries and records, along with their respective labels, for the Multi-class RUSHOLD Dataset. This dataset consists of 9210 instances and is further divided into 7368 and 1842 parts for training and testing the models, respectively.

2.2.2 Binary class RUSHOLD Dataset

Table 2 displays the total dataset entries with their respective labels for the binary class RUSHOLD dataset, which consists of 18450 instances and is further divided into 14766 and 3684 parts for training and testing of the models, respectively.

2.3 Workflow of the System

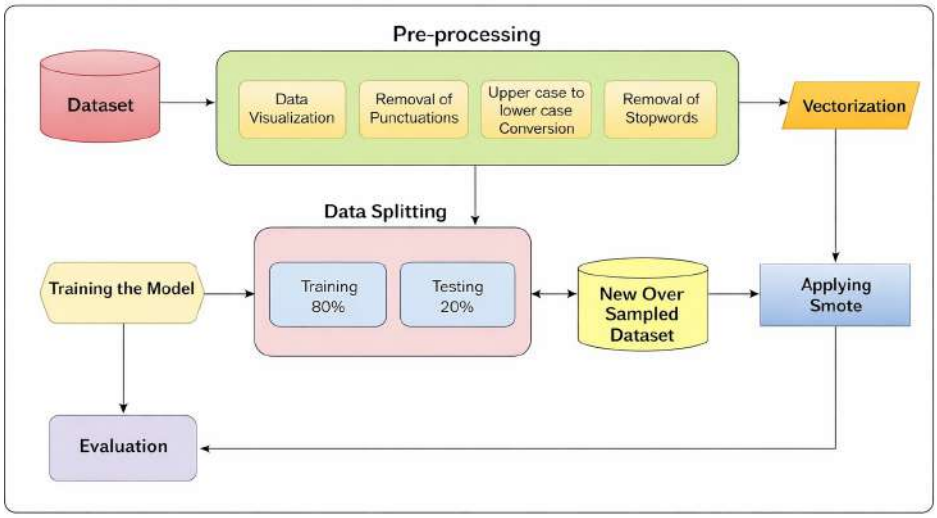

Figure 1. Workflow of the hate speech detection system using Machine Learning algorithms

Table 2. The number of instances in respective areas for Binary Class RUSHOLD Dataset

Labels	Training Data	Testing Data	Examples
Offensive	7881	1959	9840
Neutral	6885	1725	8610
Total	14766	3684	18450

2.3.1 Machine Learning Models

In the training of multi-class dataset, SMOTE is being used to balance the dataset labels. After applying smote, the labels of records became almost equal and then splitting is being done on 80% and 20% basis.

The Figure 1 shows the workflow of the hate speech detection system using Machine learning algorithms.

Some pre-processing techniques were applied on the dataset such as removing all punctuations, converting all upper-case letters to lower case, removing noisy features, and stemming and lemmatization were applied to normalize the words. Furthermore, the removal of stop words from data and removal of special symbols was done using regular expression. The SMOTE technique was then used to balance the dataset so that the model can be prevented from over-fitting or under-fitting problem. After balancing the dataset, it was split into 80:20 ratio so that the model can be trained on enough number of instances to evaluate it.

Firstly, this process converted the uppercase corpus to the lowercase corpus. The next step after this was to remove punctuation and stop words from that and remove special words using normal expression. Concluding all this with visualizing of the dataset to check uppercases and lowercases, checking punctuation, and stop words.

TF-IDF is a text vectorization method for transforming textual data into numerical vectors that can be processed by machines. It is created on two key ideas: term frequency (TF) and document frequency (DF). Term frequency calculates how often a particular word appears in a document. TF-IDF is commonly applied to portray text data in a meaningful numerical form for more analysis.

A frequent method to manage imbalanced datasets is by oversampling the minority class. The most basic method simply duplicates existing minority samples, but this does not give additional information to the model. A more optimized technique is to create new synthetic samples based on the existing data. So, the SMOTE technique is applied to produce additional records and balance the dataset.

After balancing, the dataset was split in 80:20 ratio so that the model can be trained on enough amount of instances to evaluate it. The dataset of 18450 examples is divided into 14766 and 3684 numbers of in-

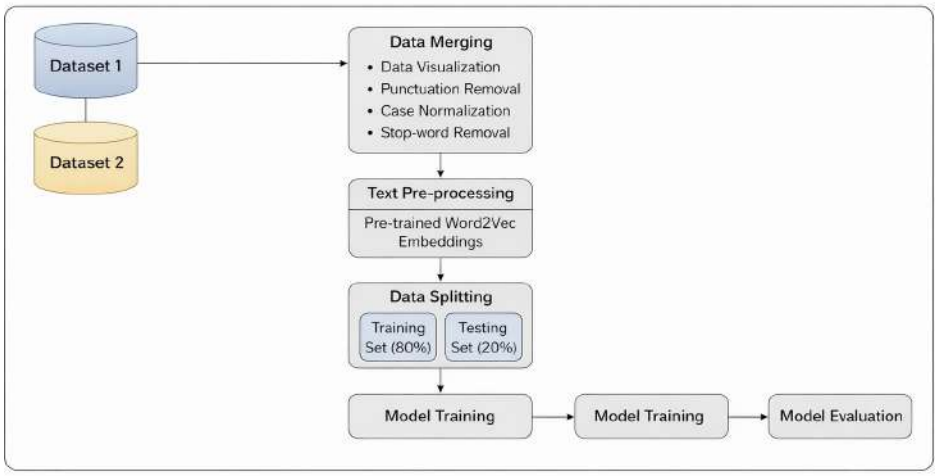

Figure 2. Workflow of the binary-class classification using Deep Learning models

stances for training and testing of the models respectively.

After splitting dataset into training and testing part, it is ready to start the training process. Now the models being fitted and train them by importing the particular function of that model. Then it gives the classification report of the model which further used in the evaluation of the model.

Evaluations of the models are being done by various evaluation techniques such as confusion matrix, by constructing comparison graphs of the different entities of the classification report. In evaluation F1-score is the major factor which is being compared in the broader perspective with the baseline paper.

In the training of Binary-Class dataset, records were not large enough so two binary class datasets were being merged that give us some good number of records (18450) to play with. Pre-trained Word2Vec embeddings were imported and then splitting is being done on 80% and 20% basis.

The Figure 2 shows the workflow of the hate speech detection system using Deep learning algorithms.

Some pre-processing techniques were applied on the dataset such as removing all punctuations, converting all upper-case letters to lower case, removing noisy features, applying stemming and lemmatization to normalize the words. Furthermore, the removal of stop words from data and removal of special symbols was done using regular expression. The SMOTE technique was then used to balance the data set so that the model can be prevented from the over-fitting or under-fitting problem. After balancing the dataset, it was split into 80:20 ratio so that the model can be trained on enough number of instances to evaluate it.

Firstly, this process converted the uppercase corpus to the lowercase corpus. The next step after this was to remove punctuation and stop words from that and remove special words using normal expression. Concluding all this with visualizing of the dataset to check uppercases and lowercases, checking punctuation, and stop words.

In deep learning, there should be a huge chunk of records. The need for large records is required because deep learning algorithms work better on larger records. This displays the total dataset entries/records with their respective labels, which gives a total number of 18450 entries.

Multiple techniques can be used to extract features from text data like Bag-of-Words, TF-IDF, etc. These techniques are great at extracting features or highlighting the words with more importance or impact but authors cannot represent the semantic relation of words or context of the words. That's why word2vec is being used to extract word embeddings here every word will be converted into a vector and it is known that every vector represents a point in the n-dimensional vector space. It has been clearly seen in [13] that word embeddings increase the predictive ability of the model.

Word embeddings are trained using neural networks either a continuous bag of words or skip-grams. Input data can also be used to train the word embeddings but because there was not an enormous amount of data so the pre-trained word embeddings trained on 4,770,0677 random and hate-speech tweets using word2vec by (Hate-Speech and Offensive Language Detection in Roman Urdu) were used.

After balancing the dataset was split in 80:20 ratio so that the model can be trained on enough amount of instances to evaluate it. The dataset of 18450 examples is divided into 14766 and 3684 numbers of instances for training and testing of the models respectively.

After splitting dataset into training and testing part, it is ready to start the training process. Now the models being fitted and train them by importing the particular function of that model. Then it gives the classification report of the model which further used in the evaluation of the model.

Evaluations of the models are being done by various evaluation techniques such as confusion matrix, by constructing comparison graphs of the different entities of the classification report. In evaluation F1-score is the major factor which is being compared in the broader perspective with the baseline paper.

3 Results & Discussion

3.0.1 Results for Multi-class Classification

The results of Multi-class classification have shown adequate improvements. Although, the F1-score is being compared in broader perspective with the baseline paper. The proposed methodology achieved the maximum F1-score of 93.69% on Random Forest whereas the maximum F1-score achieved by [16] was 75.0%.

Table 3 discusses the findings of experiments where the results of embeddings such as accuracy, precision, recall and F1-score were summarized accordingly to show the difference in two different models. The Random Forest outperforms all embeddings with an F1-score of 93.69% which is followed by XG Boost with an F1-score of 90.55% and Multinomial NB with an F1-score of 86.85%. Bert+ CNN-gram and XLM Roberta show weak performance with F1-scores of 75.0% and 72.0% respectively. Multilingual Bert embeddings yield poorest performance among all the embeddings with an F1-score of 67.0%. MNB (Multinomial Naïve Bayes), RF (Random Forest) and XGB are the proposed algorithms and the rest are from the baseline paper Algorithms.

The comparison of the results was performed with the baseline paper [16] of different algorithms to determine which model gives the results better but in the bigger perspective the results were compared on the basis of F1-score. The smote technique was applied on the Multi-class dataset because of the imbalanced dataset. Smote is a powerful solution for imbalanced dataset. SMOTE is a procedure that creates data augmentation by generating synthetic data points on the basis of original data points. SMOTE can be considered as an advanced form of oversampling, or as a definite procedure for data augmentation. The main benefit of SMOTE is that it does not just duplicate current data points; instead, it creates synthetic samples that are somewhat different from the originals. No embeddings were used but for vectorization, the Tf-idf was used. Multinomial NB was applied which gave an accuracy of 87.31% whereas azam et. al implemented XLM Roberta whose accuracy was 79.0%. Random Forest and XG boost achieved an accuracy of 93.72% and 90.67% respectively. Meanwhile M-BERT and BERT+CNN gave accuracy of 77.0% and 82.0% which shows how superior RF and XGB is. Talking about precession, the proposed Multinomial NB achieved the precession of 87.64% whereas the baseline [16] implemented XLM Roberta whose accuracy was 70.0%. Proposed Random Forest and XG boost that gave accuracy of 93.75% and 90.80% respectively. Meanwhile M-BERT and BERT+CNN gave accuracy of 72.0% and 75.0%. Talking about recall, the proposed Multinomial NB which gave the precession of 87.31% whereas the basline [16] implemented XLM Roberta whose accuracy was 75.0%. The proposed Random Forest and XG boost gave an accuracy of 93.72% and 90.67% respectively. Meanwhile M-BERT and BERT+CNN gave accuracy of 65.0% and 74.0%. Talking about F1-score, the proposed Multinomial NB which gave the precession of 86.85% whereas

Table 3. Final results of Multi-class model compared with baseline [16]

Evaluation Metric	Multinomial NB	Random Forest	XGBoost	XLM-Roberta	Multilingual BERT	BERT + CNN-gram
		Proposed			[16]	
Accuracy	87.31%	93.72%	90.67%	79.0%	77.0%	82.0%
Precision	87.64%	93.75%	90.80%	70.0%	72.0%	75.0%
Recall	87.31%	93.72%	90.67%	75.0%	65.0%	74.0%
F1-score	86.85%	93.69%	90.55%	72.0%	67.0%	75.0%

[16] implemented XLM Roberta whose accuracy was 72.0%. The proposed Random Forest and XG boost that gave accuracy of 93.69% and 90.55% respectively. Meanwhile M-BERT and BERT+CNN gave accuracy of 67.0% and 75.0%.

Table 4 shows the testing result of proposed Multi-class Machine Learning models for each individual class.

Figure 3, 4, 5 and 6 shows the comparison of normalized confusion matrices for the proposed Multi-class Machine Learning models with baseline [16] on the RUSHOLD dataset.

The confusion matrix for the baseline is reproduced from the normalized results reported in the reference study [16] for visual comparison. The heatmaps give a class-wise visualization of prediction performance, showing the models' capability to differentiate between closely related hate categories such as Offensive and Sexism. Compared to the baseline, the proposed classifiers prove enhanced diagonal dominance, representing better class separability and lower misclassification across minority classes.

Table 5 represents the testing results of proposed Multi-class Deep Learning models.

The results of Multi-class classification on Deep Learning Algorithms have not shown much good results. Although, the F1-score is being compared in broader perspective with the baseline paper. So, Bi-LSTM showed 75.00% which is maximum in results.

3.0.2 Results for Binary Class Classification

Three major algorithms RNN, Bi-LSTM, Bi-GRU have been applied to the Binary Class Dataset that gave some good results and maximum F1-score of 92.0% which implies that how accurate the model is. Furthermore, results could be more improved if the dataset was more larger.

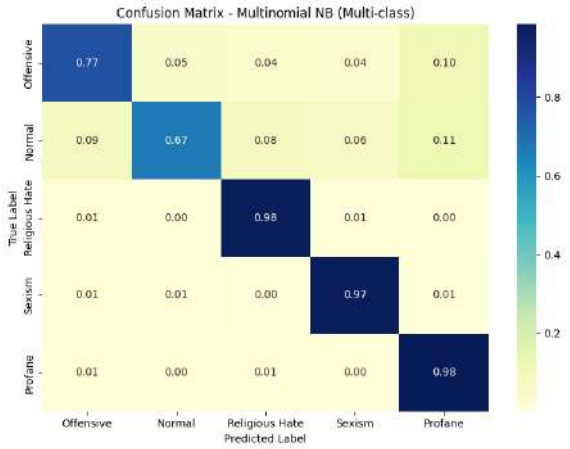

Figure 3. Confusion matrix for NB

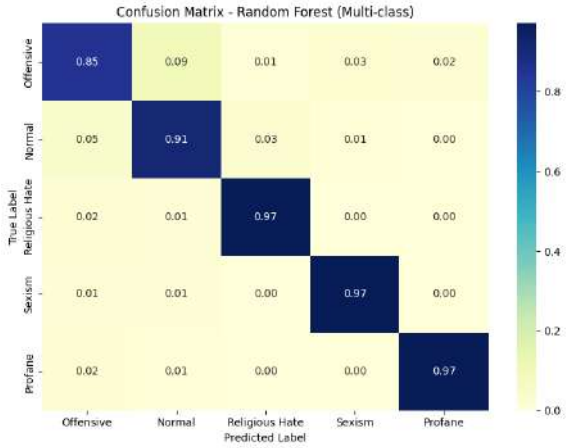

Figure 4. Confusion matrix for RF

Table 6 shows very astonishing results on Binary classification and provides us with up to 92% of accuracy and F1-Score on Bi-LSTM. Bi-LSTM outperformed all the other algorithms and the best model in terms of results and the simple RNN performed the worst, attaining an accuracy of 63% and an F1-score of 61%.

Table 4. Class-wise testing results of proposed Multi-class Machine Learning models on RUSHOLD dataset

ML Algorithm	Metric	0 (Offensive)	1 (Normal)	2 (Religious Hate)	3 (Sexism)	4 (Profane)
Multinomial NB	Precision	87.0%	91.0%	88.0%	90.0%	83.0%
	Recall	77.0%	67.0%	98.0%	97.0%	97.0%
	F1-Score	82.0%	77.0%	92.0%	93.0%	89.0%
Random Forest	Precision	92.0%	87.0%	96.0%	97.0%	96.0%
	Recall	85.0%	91.0%	97.0%	98.0%	98.0%
	F1-Score	89.0%	89.0%	96.0%	98.0%	97.0%
XGBoost	Precision	91.0%	84.0%	94.0%	94.0%	90.0%
	Recall	77.0%	92.0%	97.0%	95.0%	96.0%
	F1-Score	83.0%	88.0%	96.0%	94.0%	93.0%

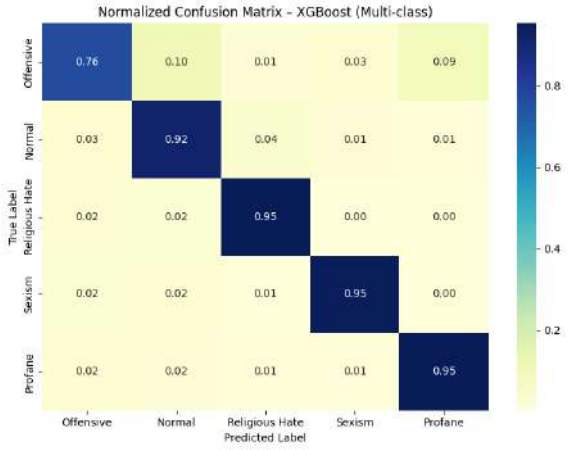

Figure 5. Confusion matrix for XGBoost

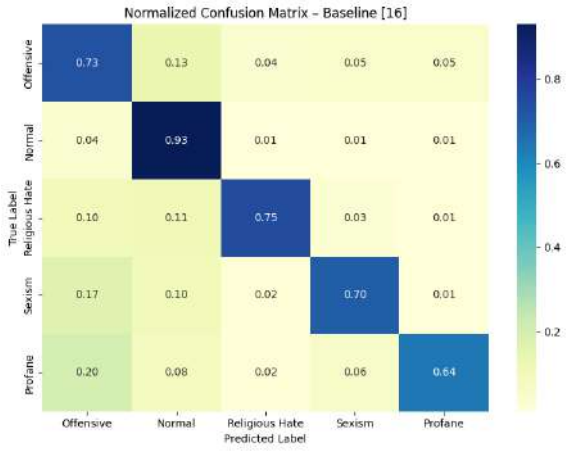

Figure 6. Confusion matrix for Baseline

The ROC curves for the binary and multi-class classification are demonstrated in Figure 7

Figure 7(a) illustrates the ROC curves for binary

Table 5. Testing results of proposed Multi-class Deep Learning models on RUSHOLD dataset

DL Algorithm	Accuracy	Precision	Recall	F1-score
Simple RNN	59.00%	51.00%	59.00%	53.00%
1D CNN	61.00%	56.00%	61.00%	54.00%
Bi-LSTM	79.00%	72.00%	79.00%	75.00%

Table 6. Testing results of proposed Binary-class Deep Learning models on RUSHOLD dataset

DL Algorithm	Accuracy	Precision	Recall	F1-score
Simple RNN	63.00%	63.00%	63.00%	61.00%
Bi-LSTM	92.00%	92.00%	92.00%	92.00%
Bi-GRU	80.00%	80.00%	80.00%	80.00%

hate speech classification using deep learning models with pre-trained Word2Vec embeddings.The curves shows robust discriminative competence across all models, with Bi-LSTM and Bi-GRU attaining higher AUC values, representing enhanced sensitivity and robustness compared to the Simple RNN architecture.

Figure 7(b) and 7(c) shows the micro-averaged one-vs-rest ROC curves for both machine learning and deep learning models for the multi-class RUSHOLD dataset. The ROC-AUC study gives a threshold-independent evaluation of class separability across all five categories. Despite the fact that the classical machine learning classifiers shows steady discriminative performance, deep learning models with pre-trained Word2Vec embeddings attain relatively higher AUC values, mainly for the 1D-CNN and Bi-LSTM architectures. This underline the usefulness of contextual sequence modeling and distributed word representations in

recording complex linguistic patterns in Roman Urdu hate speech.

4 Conclusions and Future Work

In conclusion, the discussed issue is very real and legitimate in the present day since bullying on social and digital media platforms has grown as those platforms have developed. The implementation of such hate speech filtering technologies is essential. The major goal of this study is to lessen or identify the hate speech. Various machine learning and deep learning techniques are applied in this study. Bi-LSTM, a proposed deep learning model, performed better than all other deep learning algorithms and was by far our top model in terms of outcomes. With an F1-score of 93.69 %, the Random Forest model surpasses all embeddings while utilising machine learning algorithms. This demonstrates how effective these algorithms were at identifying hateful terms. The dataset's size was the disadvantage. Our dataset was quite small, and while Deep Learning models perform better on larger datasets, simple RNNs (Recurrent Neural Networks) performed poorly, yielding an accuracy of only 63%. After considering everything, the work will be improved by using a better dataset and advanced machine learning techniques such as boosting, bagging, and stacking along with using specialized hardware for training. Also, the model deployment will be a focus for the future work.

Author Contributions

Rida Ayesha conceived the research idea, designed the methodology, and drafted the initial manuscript. **Sarah Ali** contributed to data curation, experimental analysis, and result interpretation. **Usman Inayat** assisted in model development, validation, and technical review of the manuscript. **Sajid Mahmood** supervised the study, provided critical revisions, and finalized the manuscript. All authors have read and approved the final version of the manuscript.

Compliance with Ethical Standards

It is declare that all authors don't have any conflict of interest. It is also declare that this article does not contain any studies with human participants or animals performed by any of the authors. Furthermore, informed consent was obtained from all individual participants included in the study.

5 AI Assistance Disclosure

The authors declare that the artificial intelligence (AI) tool Chat-GPT was used only for language editing, formatting, or technical refinement. No AI tool was used for the generation of research data, analysis, results, interpretations, or cited scholarly content. All AI-assisted content was reviewed and validated by the authors, who take full responsibility for the final manuscript.

6 Nomenclature

Nomenclature

Abbreviations

API	Application Programming Interface
BoW	Bag of Words
DL	Deep Learning
ML	Machine Learning
NLP	Natural Language Processing
SMOTE	Synthetic Minority Oversampling Technique
TF-IDF	Term Frequency–Inverse Document Frequency

Datasets

HS-RU-20	Hate Speech Roman Urdu Dataset
RUSHOLD	Roman Urdu Hate Speech and Offensive Language Detection Dataset

Models & Algorithms

BERT	Bidirectional Encoder Representations from Transformers
Bi-GRU	Bidirectional Gated Recurrent Unit
Bi-LSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network
GRU	Gated Recurrent Unit
LSTM	Long Short-Term Memory

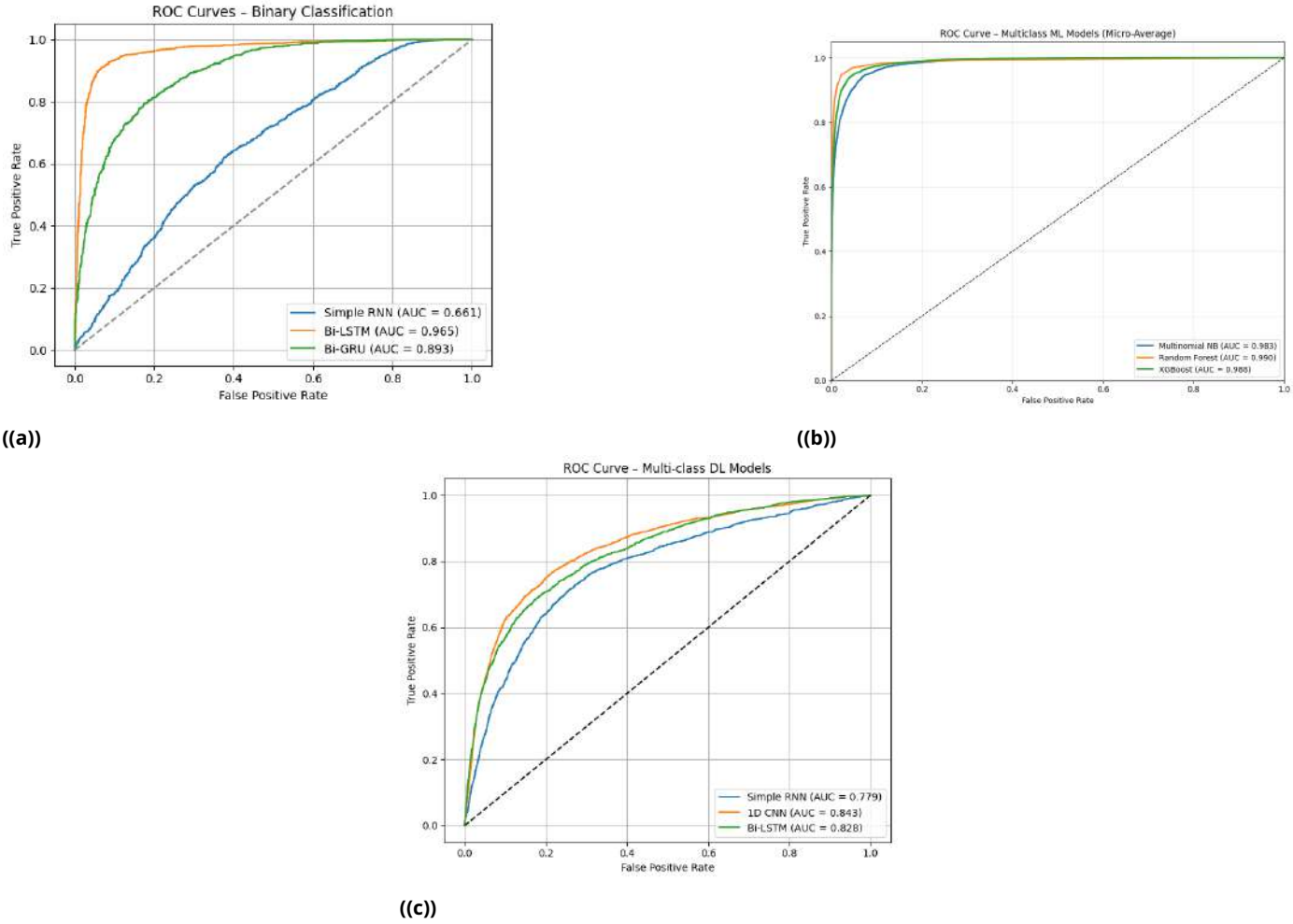

Figure 7. ROC–AUC comparison of the proposed models for both binary and multi-class classification: (a) Binary classification, (b) multi-class machine learning models, (c) multi-class deep learning models.

MNB	Multinomial Naïve Bayes
NB	Naïve Bayes
RF	Random Forest
RNN	Recurrent Neural Network
SVM	Support Vector Machine
XGB	Extreme Gradient Boosting
XLM-R	Cross-lingual Language Model RoBERTa

Symbols

F_1	F1-score (Harmonic mean of Precision and Recall)
FN	False Negatives
FP	False Positives
P	Precision
R	Recall
TN	True Negatives
TP	True Positives

References

[1] G. Ramos, F. Batista, R. Ribeiro, P. Fialho, S. Moro *et al.*, "A comprehensive review on automatic hate speech detection in the age of the transformer," *Social Network Analysis and Mining*, vol. 14, 2024.

[2] H. K. Sariyanto, D. Ulucan, O. Ulucan, and M. Ebner, "Towards explainable hate speech detection," in *Findings of the Association for Computational Linguistics: ACL*

2025. Association for Computational Linguistics, 2025, pp. 12 883–12 893.

[3] S. Nasir, A. Seerat, and M. Wasim, "Hate speech detection in roman urdu using machine learning techniques," in *2024 5th International Conference on Advancements in Computational Sciences (ICACS)*, 2024, pp. 1–7.

[4] M. Bilal, A. Khan, S. Jan, S. Musa, and S. Ali, "Roman urdu hate speech detection using transformer-based model for cyber security applications," *Sensors*, vol. 23, no. 8, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/8/3909

[5] M. S. Khan, M. S. I. Malik, and A. Nadeem, "Detection of violence incitation expressions in urdu tweets using convolutional neural network," *Expert Systems with Applications*, vol. 245, p. 123174, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417424000393

[6] A. A. Khan, M. H. Iqbal, S. Nisar, A. Ahmad, and W. Iqbal, "Offensive language detection for low resource language using deep sequence model," *IEEE Transactions on Computational Social Systems*, pp. 1–9, 2023.

[7] A. Dewani, M. A. Memon, S. Bhatti, A. Sulaiman, M. Hamdi, H. Alshahrani, A. Alghamdi, and A. Shaikh, "Detection of cyberbullying patterns in low resource colloquial roman urdu microtext using natural language processing, machine learning, and ensemble techniques," *Applied Sciences*, vol. 13, no. 4, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/4/2062

[8] M. M. Khan, K. Shahzad, and M. K. Malik, "Hate speech detection in roman urdu," vol. 20, no. 1, 2021. [Online]. Available: https://doi.org/10.1145/3414524

[9] M. Usman, M. Ahmad, M. S. Tash, I. Gelbukh, R. Q. Tellez, and G. Sidorov, "Multilingual hate speech detection in social media using translation-based approaches with large language models," 2025. [Online]. Available: https://arxiv.org/abs/2506.08147

[10] B. Barakat and S. Jaf, "Beyond traditional classifiers: Evaluating large language models for robust hate speech detection," *Computation*, vol. 13, no. 8, p. 196, 2025.

[11] M. Z. Ali, S. Rauf, K. Javed, S. Hussain *et al.*, "Improving hate speech detection of urdu tweets using sentiment analysis," *IEEE Access*, vol. 9, pp. 84 296–84 305, 2021.

[12] M. Mohiyaddeen and S. Siddiqi, "Automatic hate speech detection: A literature review," *Available at SSRN 3887383*, 2021.

[13] A. Dewani, M. A. Memon, and S. Bhatti, "Cyberbullying detection: advanced preprocessing techniques & deep learning architecture for roman urdu data," *Journal of big data*, vol. 8, no. 1, p. 160, 2021.

[14] R. G. Kodali, D. P. Manukonda, and D. Iglesias, "bytesizedllm@nlu of devanagari script languages 2025: Hate speech detection and target identification using customized attention bilstm and xlm-roberta," in *Proceedings of the First Workshop on Challenges in Processing South Asian Languages*. ACL, 2025, pp. 242–247.

[15] H. H. Saeed, M. H. Ashraf, F. Kamiran, A. Karim, and T. Calders, "Roman urdu toxic comment classification," *Language Resources and Evaluation*, pp. 1–26, 2021.

[16] U. Azam, H. Rizwan, and A. Karim, "Exploring data augmentation strategies for hate speech detection in Roman Urdu," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, Jun. 2022, pp. 4523–4531. [Online]. Available: https://aclanthology.org/2022.lrec-1.481

[17] A. Albladi, M. Islam, A. Das, M. Bigonah, Z. Zhang, F. Jamshidi, M. Rahgouy, N. Raychawdhary, D. Marghitu, and C. Seals, "Hate speech detection using large language models: A comprehensive review," *IEEE Access*, vol. 13, pp. 20 871–20 892, 2025.

[18] N. H. Usman and S. M. K. Quadri, "Scalable and advanced framework for hate speech detection on social media using bert and gpt-2," *Journal of Computer Science*, vol. 21, no. 3, pp. 584–594, 2025.

[19] M. Ahmad, M. Waqas, A. Hamza, S. Usman, I. Batyrshin, and G. Sidorov, "Ua-hsd-2025: Multi-lingual hate speech detection from tweets using pre-trained transformers," *Computers*, vol. 14, no. 6, p. 239, 2025.

[20] P. Kapil and A. Ekbal, "A survey on combating hate speech through detection and prevention," in *Proceedings of ICON 2024*, 2024, pDF available at ACL Anthology.

[21] M. K. Ngueajio, S. Aryal, M. Atemkeng, G. Washington, and D. Rawat, "Decoding fake news and hate speech: A survey of explainable ai techniques," *ACM Comput. Surv.*, vol. 57, no. 7, Feb. 2025. [Online]. Available: https://doi.org/10.1145/3711123