IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 December 2024, accepted 15 January 2025, date of publication 22 January 2025, date of current version 4 February 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3532611

RESEARCH ARTICLE

UEF-HOCUrdu: Unified Embeddings Ensemble Framework for Hate and Offensive Text Classification in Urdu

KIFAYAT ULLAH[1], MUHAMMAD ASLAM[1], MUHAMMAD USMAN GHANI KHAN[2,4], FATEN S. ALAMRI[3], AND AMJAD REHMAN KHAN[4], (Senior Member, IEEE)

[1]Department of Computer Science, University of Engineering and Technology Lahore, Lahore 54890, Pakistan
[2]National Center of Artificial Intelligence, Al-Khawarizmi Institute of Computer Science, UET Lahore, Lahore 54890, Pakistan
[3]Department of Mathematical Sciences, College of Science, Princess Nourah Bint Abdulrahman University, Riyadh 11671, Saudi Arabia
[4]Artificial Intelligence & Data Analytics Laboratory (AIDA), CCIS, Prince Sultan University, Riyadh 11586, Saudi Arabia

Corresponding author: Faten S. Alamri (fsalamri@pnu.edu.sa)

This research was funded by Princess Nourah bint Abdulrahman University and Researchers Supporting Project number (PNURSP2025R346), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

ABSTRACT Hate speech and other forms of hostile communication on social media have several implications such as; fostering violence, promoting social divide, and negative psychological effects. Since such toxic language is becoming more and more common, it is imperative to have a proper way of identifying it, especially in low resource language like Urdu. To meet this challenge, this research proposed a new ensemble based multi-classification model and generated new dataset of 36,000 Urdu tweets categorized as 'Hate', 'Offensive' and 'Neither'. This study sought to create a model that not only achieves a high classification accuracy but also overcome key challenges inherent in natural language processing, namely, high dimensionality, sparsity, overfitting, OOV words and dialectal variations. For this purpose, an extensive comparison of different learning algorithms were conducted. As a result, the most efficient models, namely FastText, XLM-RoBERTa, ULMFiT, and XGBoost were incorporated in the proposed ensemble approach to achieve the best results in both classification and mitigation of NLP issues. To further enhance the confidence in proposed model, a stratified 5-fold cross-validation technique has been utilized. The ensemble model performed the best and achieved macro F1 score of 0.94, complemented by comprehensive labeled dataset focusing on hate and offensive speech in Urdu. By addressing key research gaps, this research provides a valuable foundation for future work and benchmarking in Urdu hate speech multi-classification tasks.

INDEX TERMS Urdu hate speech detection, Urdu multi-class classification, machine learning, deep learning, transfer learning, ensemble learning model, natural language processing (NLP).

I. INTRODUCTION

Hate speech have been on the rise on social media platforms, especially in languages with limited data such as Urdu. The increase in negative content on the internet requires efficient detection tools for the protection of users and their rights [13], [14]. Hate speech is a type of communication that is inflammatory and promotes hatred against people or groups on the basis of race, colour, ethnic origin, sexual orientation, disability, gender or religion. Offensive speech, on the other hand, are statements that are insulting and hurtful, although not necessarily leading to violence, but with the aim of causing harm or discomfort to others [16]. It is therefore important to differentiate between these two forms of speech so as to create better monitoring and classification systems. Sentiment analysis has been commonly used to classify the opinions or sentiments in text [16]. Nonetheless, identifying hate and obscene speech is a more intricate task due to the need to account for the semantics, cultural background, and dialects [12], [20].

Urdu is a language that is used by millions of people and brings its own set of challenges to the field of natural language

The associate editor coordinating the review of this manuscript and approving it for publication was Abdel-Hamid Soliman.

processing. Some of these challenges include lack of labeled data, high dimensionality of data and dialectal variations, which makes it difficult to design reliable models for text classification [20]. Previous approaches have relied on text classification techniques such as SVM and Naive Bayes but the problem with these approaches is that they don't capture the sentiment of the context as effectively as they should, especially, when dealing with languages like Urdu which has a rich cultural background [8], [36]. In the recent years, the application of machine learning (ML) and deep learning (DL) have been used to address these issues. Transfer Learning with the use of pre-trained models like BERT, RoBERTa and their multilingual versions have been found to be very useful in the identification of hate and offensive speech [7], [15]. These models are based on huge prior knowledge and are quite effective at handling the complexities of under-resourced languages.

Urdu has recently been studied mainly in the context of Roman Urdu script where Urdu is written in Roman alphabet mostly by the younger generation on social media platforms [15], [37], [38]. In Roman Urdu, there has been a breakthrough in the development, however, there is a lack of studies on the identification of hate and offensive text written in Nastaliq script. This research gap is significant due to the fact that Nastaliq-scripted Urdu is common and has unique features that make it challenging to identify hate and offensive language. However, there are several

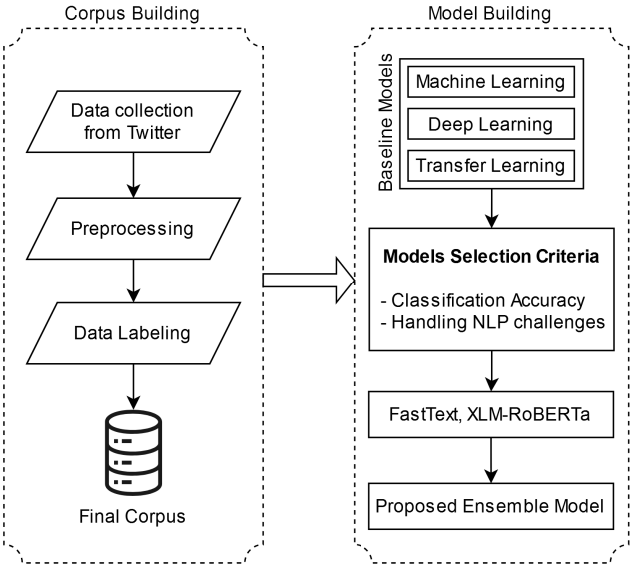


FIGURE 1. Overview of research work.

significant research gaps that are still to be explored in the context of hate speech detection for Urdu language. A major challenge is the lack of labeled data for Nastaliq-scripted Urdu especially due to the lack of publicly available data. Furthermore, the emphasis on hate and offensive speech has been rather narrow with many studies focused on other text categorization tasks. Most of the current datasets are binary, where they classify content as either positive or negative while there are limited multi-class datasets that could classify content into hate, offensive and neither classes. Moreover, while transformer-based architectures and ensemble learning techniques have shown promising results in other tasks and their applicability to the Urdu hate speech detection is still relatively unexplored. Other common NLP problems, including high dimensionality, sparsity, and OOV words, still exist and further complicate this area of research.

In order to overcome the challenges and fulfill the research gaps mentioned above, we started off by collecting a large dataset of Urdu tweets from Twitter using it's API, as depicted in Figure 1. The collected data was carefully preprocessed to ensure quality and consistency, including the removal of noise such as irrelevant symbols, hashtags, and duplicate entries. This clean dataset was further categorized into three distinct classes: Hate, Offensive, and Neither. This final labeled dataset of 36,000 tweets was utilized to train our proposed ensemble model, as well as baseline models from traditional Machine Learning, advance Deep Learning, and Transfer Learning approaches. When selecting a model to incorporate into the proposed ensemble framework there were two main considerations; high classification accuracy and effectively addressing NLP problems. Only models that met these criteria were incorporated into the proposed model. To address these issues and achieve better classification results, this study combined the unique strengths of multiple models within an ensemble learning framework, specifically integrated FastText, XLM-RoBERTa, ULMFiT and XGBoost.

The contributions of this research work can be summarized as: (1) A new dataset of 36,000 annotated Urdu tweets is developed to train and validate the model. It is a useful resource for future research in the context of hate and offensive speech classification in Urdu language. (2) The study proposed an ensemble model to fill the gap associated with efficient detection models for hate speech in Urdu language and achieved a macro F1-score of 0.94. (3) To effectively mitigate the challenges associated with NLP, this model combined FastText, XLM-RoBERTa, ULMFiT, and XGBoost within an ensemble framework.

The rest of this paper is structured as follows: Section II explores previous work in context of hate and offensive text speech detection and classification. Methodology and proposed model architecture are presented in section III. Section IV is dedicated to experimentation. Section V presents the results for comparative analysis of baseline and the proposed model. Finally, conclusion of the paper followed by the possible directions for further research is presented in Section VI.

II. LITERATURE REVIEW

The challenge of detecting hate and offensive language has become a critical area of interest in the NLP domain because of its social importance. Numerous research has been conducted in identifying and categorizing such content across different languages. Nevertheless, the particular problems of

Urdu text have only begun to be discussed in detail in recent years.

This section begins by reviewing existing studies conducted in various languages and concludes with a critical analysis.

A. EXISTING WORK IN DIFERENT LANGUAGES

This section provides an overview of previous studies conducted in various languages, including English, Roman Urdu, and Nastaliq Urdu.

1) ENGLISH LANGUAGE

The area of hate speech classification has been found to have developed especially where data sets are in English language. There are several approaches that have been suggested to enhance the efficacy and reliability of hate speech classifiers. In this section we have presented a brief discussion of some of the relevant work that has been done in the literature, focusing on the techniques and dataset that has been used in the development of underlying systems for English Language which could be helpful in similar work in Urdu.

Reference [1] presented a noval method for the detection of hate speech in English language which is based on a dual contrastive learning framework. The research also shows the difficulty in defining hate speech and the analysis of the content where the examples are rather ambiguous and often depend on context. To tackle such problems, the authors presented a dual contrastive learning approach that incorporates self-supervised and supervised learning. The unsupervised part of the model is intended to learn features from the large amount of data without label, whereas the supervised part refines the learned features with the labeled data. To evaluate the model, various English language datasets were used. The comparative analysis of the results showed that the proposed model was more efficient than the latest approaches in accuracy and F1-score.

In their work of [3], the authors concentrated on the transfer learning methodologies in the identification of hate speech on social media. The work revolves around one particular problem of hate speech detection and how pre-trained language models such as BERT can be employed for the same. The authors further state that transfer learning is most efficient when the amount of labeled data is limited or when the hate speech content is specific to a particular domain. They also conducted experiments on English language datasets and found out that transfer learning enhances the classification performance especially when it comes to handling the Minority Classes or the Subtle Hate Speech. This is especially useful in a multilingual context like Urdu, where there may be a lack of annotated data, and transfer learning can be used to overcome this by utilising knowledge from other languages.

In [4], the authors introduced a novel method for the extraction of features for English text classification through the use of Ant Colony Optimization (ACO). This research is based on the problem of identification of the best features which are useful for hate speech classification in high-dimensional data sets. The ACO algorithm, which mimics the foraging of ants, enhancing the classification performance. The authors tested the method on several English language hate speech datasets and showed that it is beneficial in dimensionality reduction without compromising the efficiency of the model. This approach is more suitable for dealing with high-dimensional text data, which is a challenge for most feature selection algorithms.

Reference [35] present a new hate speech detection dataset named 'ETHOS' which comes in two versions: The classification of the dataset is binary and multi-label. All the data is collected from YouTube and Reddit comments and it is moderated through Figure-Eight crowdsourcing platform. The annotation process also uses an active sampling technique to deal with the class imbalance problem and to ensure that no case of hate speech is missed.

In [6], a multimodal hate speech detection model is proposed using both text and image inputs with the help of a fusion network. It leverages the latest pre-trained models including DistilBERT, MPNet, ResNet, and EfficientNetV2 to capture content dynamics across different modalities. The methodology also uses the results of context-free word embeddings (GloVe, fastText) with contextual language models and achieves comparable results on the MMHS150K dataset.

2) ROMAN URDU

Roman Urdu is a script (Urdu written with English alphabets) that is widely used by Urdu speaking people on the social media platforms. Following are some of the various approaches that have been used in previous studies.

Reference [9] used classical machine learning technique for identifying hate text content in Roman Urdu language. In their research, they employed a corpus of Roman Urdu text which was cleaned from all forms of noise such as special characters and repetitions of the same characters that are common in digital media. To classify the text as hate and non-hate the authors employed some machine learning models, i.e., SVM, Naive Bayes and Random Forests. The study also revealed that the traditional models especially SVM gave a fair performance in detecting hate speech.

Reference [13] contributed to the field by proposing the use of transformer-based models with a focus on the usage of transformers in cyber security in Roman Urdu. In their work they also demonstrated how deep learning outperforms other machine learning algorithms in capturing context information which is important in the detection of hate speech. To overcome the challenges posed by Roman Urdu the study had employed a transformer model that apply self-attention mechanism. The model was trained on a large dataset of Roman Urdu text and the results were quite good in identifying hatred content and other forms of offensive language.

In their research, [12] presented a new work entitled Passion-Net which is the integration of hate speech detection in Roman Urdu and Explainable AI (XAI). The Passion-Net model which is proposed in this paper is expected to deliver high precision and simplicity of interpretation to effectively address the need for explain ability in hate speech detection systems that are based on AI. The study employed a big corpus of Roman Urdu text and employed an XAI model which can give reasons for its decisions. This helped in increasing the effectiveness of the model in classification of hate speech while at the same time allowing for an understanding of the model as well as the reason as to why it was classifying certain texts as hate speech. This is especially true in areas where accountability matters a lot, as it is in the health sector where the cause of actions made by the AI need to be known.

Reference [16] proposed a solution that comprises of Roman Urdu Spelling Checker (RUSC) and Bilingual Roman Urdu Language Detector (BRULD) to achieve improved results in sentiment analysis for Roman Urdu. RU-PHS, the dataset proposed by [14] for the task of political hate speech detection in Roman Urdu, contains 5,002 annotations. It employs lexical unification algorithm and uses three vectorization techniques, including TF-IDF, word2vec, and fastText in an attempt to compare the performance of the traditional machine learning techniques and the fine-tuned neural networks for the identification of the political hate speech.

3) NASTALIQ URDU

The research on hate and offensive Urdu language detection has received much attention in the recent years with the use of advance deep learning and transfer learning approaches. This section presents the literature review that has been gathered from the available research conducted in Nastaliq-scripted Urdu language.

Reference [21] suggested the identification of hate speech in Urdu. They also used Transfer learning algorithms by tuning the pre-existing models on the Urdu data. They focused on the problem of the Nastaliq script and designed a system that can identify hate speech and also the communities that are being targeted in the text. The work demonstrates that transfer learning helps in overcoming the challenges posed by the Urdu language that is sparsely covered in NLP datasets.

Reference [22] employed a deep learning technique to identify offensive language in Urdu text. They suggested using attention mechanisms that would enable the model to pay attention to specific sections of the text and, consequently, improve the model's performance. The study used a large amount of Urdu text data and showed that deep learning with attention-based mechanisms can be used to better capture the context and intricacies in Urdu hate and offensive language.

Reference [19] presented a comprehensive dataset which is designed to identify cyberbullying in Urdu tweets. This work offers an effective detection method which trained different classical learning models to determine cyberbullying trends across social media posts. This research is important as it presents one of the first large scale datasets in Urdu which accurately captures language and cultural factors that are crucial for detecting cyberbullying. This research outlines the difficulties in handling Urdu text resulted from its rich morphology and syntax, provided a starting point for future work in this domain.

Reference [23] highlighted the identification of threatening language in Urdu language through deep sequential models. Their work entailed creating dataset and applying a comprehensive deep learning approach to identify whether the content is threatening or not. The authors also employed models such as LSTM and BiLSTM which implies that deep learning algorithms can recognize context and meaning of the Urdu text data. The present work can be considered as a part of the ongoing research of text classification in Urdu and underlines the necessity to develop a model which implement the specific features of the Urdu language.

Reference [24] discusses the detection of emotions in Urdu using a deep contextual neural network. It accentuates the importance of context in understanding the application in low-resource languages. It tackles issues such as sparsity and dialectal differences and gives a contribution to the understanding of text classification in Urdu language.

Reference [7] worked on a novel hate text detection system in Urdu called UHated. They employed a transfer learning approach, pre-trained on Urdu hate speech dataset, demonstrating how leveraging knowledge from high-resource languages can improve classification performance in low-resource languages. UHated successfully responded to these challenges and has managed to achieve great results in detecting hate speech and establishing further research.

Reference [20] proposed annotation guidelines for Urdu text and then collected a dataset of 21,759 tweets belonging to multiple domains i.e. Interfaith, Sectarian and Ethnic hatred. They compared eight machine learning and deep learning approaches and found BERT the most suitable for detecting hate speech.

Reference [18] in their latest research proposed a framework to detect the violence incitation in Urdu tweets based on the semantic embedding and language modalities. It investigates the use of 1-Dimensional Convolutional Neural Networks (1D-CNN) with parameter tuning on a manually labeled dataset of 4808 tweets. The classification results of the proposed 1D-CNN with word uni-gram embeddings are presented and compared with other models, which show better performance in terms of violence and non-violence classes accuracy and F1 scores.

Reference [11] proposed a multi-class Urdu dataset of 9,312 labelled reviews spanning across different domains for sentiment analysis. It compared various classifiers, such as rule-based and machine learning, as well as deep learning

approaches, and fine-tuned Multilingual BERT (mBERT) with different text encoding to set the baseline performance.

In their work, [10] collected a dataset of 7500 posts from Pakistani Facebook pages for the purpose of Urdu offensive language detection. It employed four feature engineering models, which include TF-IDF, BoW, n-Grams, and Word2Vec embeddings.

To overcome the problems of data sparsity, high dimensionality and class imbalance in the sentiment analysis of Urdu tweets, [8] present a detailed dataset and solutions. The methods used are the Dynamic Stop Words Filtering, VGFSS, and SMOTE. For the purpose of classification SVM and MNB were employed.

4) OTHER LANGUAGES

The use of multilingual approach for text categorization and particularly in the detection of hate speech has been on the rise lately. More attention have been paid to the issues related to multilingual contexts, particularly code-switching which is the use of two or more languages in one sentence or phrase. For instance, [15] created a large-scale dataset comprising on multiple languages, i.e., English, Roman Urdu, Urdu, and English, for the classification of cyberbullying based on aggression, repetition, and intention to harm. To develop a framework for classification of the text messages and to assess these features, fine-tuned MuRIL was used which resulted in good performance metrics.

Similarly, [17] in their work applied a lexicon-based method to identify toxic language at the sentence level in bilingual text written in formal English and informal Roman Urdu. The focus was on three domains: including race, religion and nationality. In another work, [11] employed the Roman Punjabi text and afterwards employed the BERT based models for classification of hate speech in code-switched English and Punjabi content. In this paper, a new approach that has new hate speech detection techniques to address the shortcomings of the current state of the art is proposed. Reference [5] introduced a transformer model for detecting hate speech in the social media context. The model is general and it was validated with data which is in English, Italian, German, Bengali languages. The performance of the proposed model was better than the baseline and state-of-art model for hate speech detection.

Reference [25] present TABHATE, a target-based hate speech dataset in Hindi language, given the scarcity of data for hate speech detection in languages other than English. They applied deep learning and transformer models. Reference [26] provide a hate speech detection framework for code-mixed Hindi-English and Hindi text in Devanagari script considering problems such as small datasets and language variations. The Tabnet classifier which they proposed can easily handle both the transliterated and the native scripts and has been trained on the features extracted by MuRIL. Reference [27] also discussed identification of hate speech in Indian languages and issues like multilingualism, code-mixing and resource constraints. They focuses on the importance of the automated approach to address abusive content and analyze the recent works concerning datasets, features, and classification approaches.

Reference [28] explored the performance of PLMs for hate speech detection in Arabic tweets and compare them with other ML and DL models. They also observed that the proposed multidialectal PLMs are more effective than monolingual and multilingual models, and fine-tuning greatly improves classification performance. Reference [29] presented hate speech detection system in Bengali; their work explored problems such as the scarcity of datasets, the complexity of the language, and variations in context. They also discuss different learning algorithms and performance measures. The work also highlights the importance of the dataset size and cross-lingual transfer learning for improving detection performance.

Reference [30] conducted an analysis of transfer learning models for hate speech classification on European Portuguese social media, employing BERTimbau, mDeBERTa, GPT, Gemini, and Mistral models. In their work they analyzed comments on YouTube and tweets, BERTimbau on European Portuguese reached 87.1% of F-score and for tweets GPT-3.5 was the best. This work also demonstrates the role of in-domain training and context in generative models. Reference [31] meet the problem of identifying target-oriented hate speech in Turkish print media by creating the TurkishHatePrintCorpus, annotated with group-targeted language. To this end, they presented HateTargetBERT, a model that incorporates target specificity into BERT, improving hate speech detection and model interpretability.

In their study, [32] assess ChatGPT and fine-tune three BERT-based styles for hate speech detection in Turkish. Reference [33] focus on the hate speech and abusive language (HSAL) detection in Indonesian social media, and they found that the traditional machine learning methods are still dominating with classic text representation features. Reference [34] have employed the analysis of online and printed hate speech to determine that hate speech increases conflict and divisions. The authors are concerned with the issue of using polarizing language that may lead to social conflicts in diverse African societies.

B. CRITICAL ANALYSIS

As the issue of online hate speech becomes increasingly prevalent, especially in low-resource languages such as Urdu, the need for effective detection mechanisms becomes crucial. Despite advancements in natural language processing (NLP), major research gaps still persist in detecting and classifying hate and offensive text in Nastaliq Urdu, as most studies primarily focused on Roman Urdu. Additionally, most of the previous research work relied on small and binary-based imbalanced datasets, which made it hard to achieve model generalization and lead to biased performance.

In light of Nastaliq Urdu, existing studies lack scalable framework for their underlying models and fail to address

common NLP challenges such as high-dimensionality, sparsity, out-of-vocabulary (OOV) words, overfitting and dialectal variations. While individual transfer learning models, in literature, have shown potential but their integration into ensemble frameworks for enhanced feature representation and classification remains unexplored for Nastaliq Urdu.

This research is motivated by the need to address the limitations of existing models and datasets, especially for Nastaliq Urdu. For this purpose, a manually annotated, balanced, multiclass dataset comprised of 36,000 Urdu tweets was carefully developed. This dataset ensures sufficient volume for model training and generalization while addressing issues of class imbalance. The contextual and semantic feature of embeddings generated from state-of-the-art transfer learning and deep learning models, i.e. XLM-RoBERTa, ULMFiT, and FastText helps overcome common NLP challenges. The proposed model is also scalable, more models can be incorporated in future to derive other embeddings that further enriches feature representation. Furthermore, the framework employs XGBoost as a classifier, which inherently supports ensemble learning, reducing variations in classification and minimizing confusion between classes. Additionally, the use of stratified k-fold validation technique ensures reliability and balanced training across all the classes.

In conclusion, this study addresses critical research gaps by presenting a scalable ensemble model that incorporates diverse embeddings for multiclass classification of hate and offensive speech in Nastaliq Urdu text. This approach provides a robust solution to existing challenges and establishes a foundation for future advancements in Nastaliq Urdu.

III. METHODOLOGY

This section discussed the corpus building process and the extensive measures taken to construct the multi-classification models for categorizing Urdu text into ''Hate'', ''Offensive'', and ''Neither'' classes. We adapted different Machine Learning, Deep Learning and Transfer Learning technqiues that were employed in previous studies and trained these models on our newly established hate and offensive Urdu dataset. We analyzed the strenghts and limitations of these baseline models and the insights gained from this analysis guided development of the proposed model, that is a hybrid of the top-performing models identified through this analysis. The baseline models served as reference points for comparison with our proposed model.

In digital media and other communication channels, the use of hate and offensive language is widespread, therefore, there is a need for better detection tools. Some common natural language processing (NLP) issues i.e. high dimensionality, overfitting, sparsity and out-of-vocabulary (OOV) words along with limited resources and dialectal variations in Urdu language further complicate the process of classification. In order to handle these challenges effectively, this research work proposed an ensemble learning model which combines the strength of multiple models to enhance the classification performance and mitigate the problems inherent in NLP. Table 1 depicts the most recent and cutting-edge baseline models that we will use in our research for analysis and comparison with our proposed method.

TABLE 1. Comparison of latest baseline work.

Author	Methodology	Results	Dataset
[19]	ML and DL techniques	F1-score = 0.84	12,500 rows
[18]	1D-CNN, BERT, and ML models	Accuracy = 90%	4,808 rows
[7]	Transfer learning with FastText and RoBERTa	F1-score = 0.82	7,800 rows
[10]	Ensemble of BoW, TF-IDF, Word2Vec	Accuracy = 90%	7,500 rows
[11]	mBERT	F1-score = 0.82	9,312 rows

The methodology is structured to address several key objectives: (1) to build a corpus of labelled data consisting of Urdu tweets classified as: ''Hate'', ''Offensive'', and ''Neither''; (2) to use simple feature extraction methods including BoW and TF-IDF and more sophisticated text representation techniques including pre-trained word embeddings like FastText and Word2Vec to better capture the semantic and context present in Urdu text; (3) to develop, train and compare several learning models that will be used as a reference point for comparison with our proposed ensemble model; (4) to design and build a proposed model for enhanced classification accuracy and reduced impact of high dimensionality, sparsity, overfitting and OOV words; and (5) to test and compare the efficiency and effectiveness of proposed model with baseline in regards of classification metrics.

The next section will describe the process of constructing the corpus, from the collection of raw data to the preprocessing and labeling of the data, which formed the foundation of our model development.

A. CORPUS BUILDING

Here we discuss the steps which were carried out in order to create Urdu dataset for hate speech and offensive language detection. Initially, we collected 36,000 tweets from twitter consuming its API. Specific hashtags were used to capture tweets belonging to different categories such as political, sectarian, religious etc. For offensive data, a lexicon of offensive words were used to capture tweets that contain offensive language. Following the data collection, pre-processing step was employed to remove noise such as URLs and hashtags from tweets. Finally, the data was manually labelled into ''Hate'', ''Offensive'', and ''Neither'' categories. Figure 2 summarizes the overall process of corpus building. All these steps will be elaborated in the subsequent sections.

1) DATA COLLECTION

The dataset was sourced from Twitter, with raw data collected using the Tweepy client, which provides access to both real-time streaming and archived data. Algorithm 1 presents

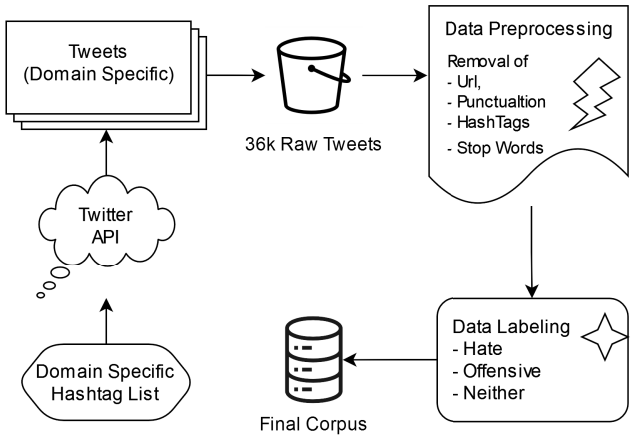


FIGURE 2. Corpus building process.

the pseudo-code of underlying python code. As our study is based on the Urdu language, the Twitter API was set to search for the Urdu language only. To collect the tweets that contains hate content, we used the API with a set of keywords and hashtags that are related to the topic. This process generated a lot of data, and the data was then cleaned and preprocessed.

Algorithm 1 Fetching Tweets Using Tweepy Client

Input:
- Twitter API bearer token for authentication
- API parameters: keyword, language, start time, result limit

Output:
- A dataset of tweets with columns: tweet id and tweet text

Initialize:
- Initiate authentication using secret bearer token
- Create a Tweepy client
- Initiate keyword, language, start time, max results, limit
- Initialize a list `hs_tweets` for column names
- Create a query string excluding retweets

for *each response in the paginated query results* **do**
  Add a delay of 0.1 seconds to prevent rate limits
  Append tweet id and tweet text to `hs_tweets`
end for

if *no exception occurred* **then**
  Convert `hs_tweets` into a DataFrame;
  Remove duplicate entries;
  Save the DataFrame as an Excel file;
end if

2) DATA PREPROCESSING

The data preprocessing is illustrated in Figure 2, which undergoes the removal of any unwanted data and organize it in such a way that is suitable for model training. Initially, the data was in raw form which consists of tweets in different languages that have similar syntax and writing style like Urdu including Pashto, Persian and Arabic. The first step in data preprocessing was the removal of all those tweets that were not in Urdu language. Next all the symbols, the punctuations, and other characters that do not help in classification were also omitted. These include the use of hashtags, handles, links, emojis and extra spaces etc. All the common words which do not add any value to the document, also known as stop words, were removed with the help of a stop word list, for instance "اور","کے","میں","کو","تھا" etc. Finally the Roman Urdu text were also translated into standard Urdu script where ever it was used in the text. The next step explains the "Data Labelling Process" which was carried out manually.

3) DATASET LABELLING

Figure 3 illustrate the complete process of data annotation. The data preprocessing was followed by manual labelling process in which each tweet is examined in order to identify the presence of hate or offensive context. If there were presence of an offensive word, the tweet was straight classified as "Offensive". If the context of a given sentence is hateful but does not contain any offensive words, then it is assigned to the Hate class. Tweets that do not belong to these two categories are classified as "Neither". Table 2

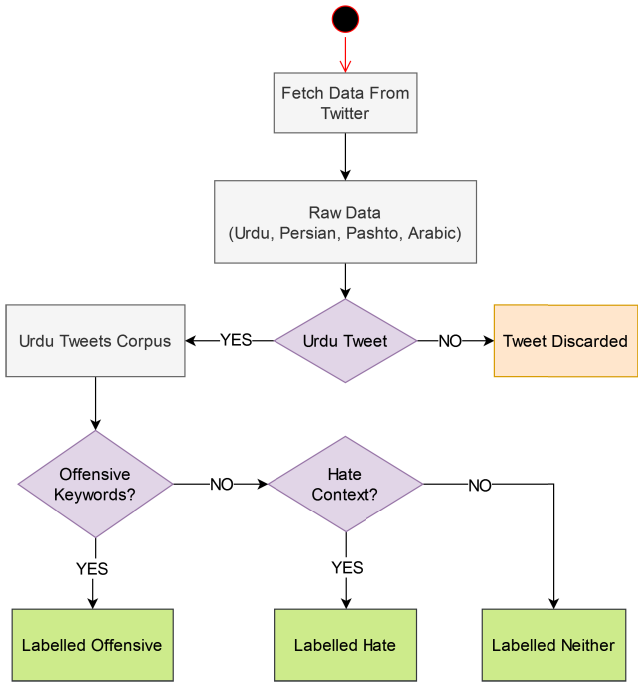


FIGURE 3. Data labelling process flow.

shows the distribution of data and sample of noise-free tweets that are ready for feature engineering. The second sample tweet is classified as "Hate" because it is promoting negative opinion against a particular group of people, implying that they are a danger to the nation. It is provocative, promotes fear and violence, which is hate speech. The last sample tweet is classified as "Offensive" because it contains offensive language to refer to a group of people as 'ignorant' and accusing them for the problems in the country. Although it is rather provoking and impolite, it does not contain such

elements as violence or threat to harm, that is why it is considered as the offense speech.

TABLE 2. Corpus distribution.

Class	Label	Quantity	Sample Tweets
Neither	0	12,000	ہفتے کے آخر میں چھٹیوں کا پلان بنا رہا ہوں۔ کوئی جگہ تجویز کریں؟
Hate	1	12,000	یہ لوگ ہمارے ملک کے لیے وبالِ جان ہیں، ان کا ہر جگہ ہونا ہمارے لیے خطرہ ہے۔
Offensive	2	12,000	تم جیسے جاہلوں کی وجہ سے ملک تباہ ہو رہا ہے

4) WORD CLOUD VISUALIZATION

Figure 4 depicts the word cloud visualization of the entire corpus, it is to visually represent the most frequent words in the dataset, which consists of three classes: ''Hate'', ''Offensive'', and ''Neither''. This graphic aid is useful for underlining the keywords related to each category and get a sense of the frequency distribution in the data set.

FIGURE 4. Word cloud representation of entire corpus.

B. FEATURE ENGINEERING

Training of learning algorithms could only perform on numerical data and cannot directly interpret textual information, which makes feature extraction a crucial step to convert text data into numerical features that are understandable by learning algorithms. For this purpose, we have employed the following feature engineering techniques to cover as many aspects of the text as possible, including context and semantic relationship.

1) BAG OF WORDS (BoW)

It is one of the basic approaches to represent text in NLP. It transforms text into a matrix in which each row represent a document, and every column corresponds to word that occurs at least once in the whole corpus. The numbers in the matrix represent the count of the words occurred in a given document and it does not include word order and grammar.

$$Z = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1p} \\ z_{21} & z_{22} & \cdots & z_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ z_{q1} & z_{q2} & \cdots & z_{qp} \end{bmatrix} \tag{1}$$

where, Z is the transformed matrix, q is the total number of documents, p represents the total number of unique terms in the collection, z_{ij} represents the frequency of term j in document i.

2) TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) is a method for determining significance of a term in a document compared to a whole corpus. It is most frequently applied in text analysis. The first part is the TF, calculates how often a word is used in a specific document and the second part is the IDF which scales down the frequency of words that are used frequently in all documents. This way, TF-IDF makes focus on the keywords in a particular text and reduces the frequency of the stop words that are used in many documents. Equation for TF-IDF is as under:

$$\text{TF-IDF}(w, \text{doc}, C) = \text{TF}(w, \text{doc}) \times \text{IDF}(w, C) \tag{2}$$

where:

$$\text{TF}(w, \text{doc}) = \frac{n_{w,\text{doc}}}{\sum_j n_{j,\text{doc}}} \tag{3}$$

$$\text{IDF}(w, C) = \log\left(\frac{|C|}{|\{\text{doc} \in C : w \in \text{doc}\}|}\right) \tag{4}$$

where, w represent word or term, doc is used for document, C is the corpus of documents.

3) WORD2VEC EMBEDDINGS

Word2Vec is a neural network model which produce embeddings that are numerical interpretation of vocabulary words that retain the meaning and inter-relationship. It maps words to high dimensional vectors of a fixed size and bring semantically related words close in the vector space. There are two architectures in Word2Vec, i.e. CBOW that predicts a word given the context and Skip-Gram predicts the context for a given word. Equation of Word2Vec for the Skip-Gram is:

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{-m\le k\le m, k\neq 0} \log Q(v_{i+k}|v_i) \tag{5}$$

where, N are the total number of terms, m represent context window size, v_i is the target term, v_{i+k} are context terms, $Q(v_{i+k}|v_i)$ is the probability of context term given target term.

4) FASTTEXT EMBEDDINGS

It is a word embedding technique developed at Facebook AI Research (FAIR) as an enhancement to Word2Vec where the model takes into consideration word parts instead of the entire word especially for languages that have many words that are derived from other words. In contrast to learning vectors for whole words, FastText uses character n-grams in order to come up with embeddings for out-of-vocabulary words. The FastText embedding for a term t is calculated as the aggregate of the embeddings of its character n-grams:

$$\mathbf{e}(t) = \sum_{n\in N(t)} \mathbf{e}(n) \tag{6}$$

where $N(t)$ denotes the collection of all character n-grams associated with the term t, and $\mathbf{e}(n)$ represents the vector corresponding to the n-gram n. This approach helps in increasing the generalization capability for the out-of-vocabulary words and provides enhanced representations for the languages that have a lot of inflections.

5) XLM-RoBERTa EMBEDDINGS

XLM-RoBERTa embeddings are contextualized word vectors derived from a transformer-based architecture which is pretrained on multilingual corpus of 100 languages including low resource language like Urdu. These embeddings represent meanings of words within context and generate dense, high rank vectors which perform well in model architectures that involve word meaning understanding and transfer learning to different languages. Hate speech detection is one of the areas where they shine, since they are capable of dealing even with slang and idioms, and even in different languages.

6) ULMFIT EMBEDDINGS

ULMFiT embeddings are contextualized word vectors that are specifically trained to be further fine-tuned for downstream objectives. ULMFiT uses a two stage training procedure where it first learns a set of general language representations from a very large corpus of text and then fine-tunes these embeddings on text relevant to the task at hand. This makes it easy to do transfer learning, which is quite helpful when working with text classification tasks that have little labelled data.

Table 3 presents the comparison of various embeddings used in this research work and Table 4 illustrates the feature type utilized with each approach.

TABLE 3. Comparison of embedding techniques.

Embedding	Context	Dimensionality	Pre-training
Bag of Words (BoW)	No	Vocabulary size	No
TF-IDF	No	Vocabulary size	No
Word2Vec	No	100-300	Yes
FastText	No	100-300	Yes
XLM-RoBERTa	Yes	768+	Yes
ULMFiT	Yes	400-1150	Yes

TABLE 4. Methodologies and feature engineering techniques.

Methodology	Features
Machine Learning	BoW & TF-IDF
Deep Learning	Word2Vec & FastText
Transfer Learning	Dynamic contextual embeddings
Proposed Ensemble Method	FastText, XLM-RoBERTa, ULMFit

In this research work, different pre-trained embeddings were incorporated in deep learning models. Table 5 lists their filename and size.

TABLE 5. Source files of pre-trained embedding.

File Name	Embedding Type	File Size
urduvec_140M_100K_300d.bin	Word2Vec	118.1 MB
cc.ur.300.vec	FastText	2.43 GB

C. EVALUATION METRICES

To evaluate the efficiency of the developed models and perform a detailed comparison, a set of different evaluation metrics were used. These metrics give a detailed evaluation of two things, the performance in general and the accuracy in classifying instances into different classes. Below are the key metrics used in this study:

1) ACCURACY

Accuracy is total number of instances classified correctly divided by the total number of instances, including both positive and negative. It gives an overall measure of the effectiveness of the classifier. Equation for Accuracy is as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{7}$$

where TP stands for True Positives, TN refers to True Negatives, FP denotes False Positives, and FN corresponds to False Negatives.

2) PRECISION

It is defined as the true positive divided by the sum of true and false positive. It depicts the level of the actual positive predictions. Equation for Precision is as under:

$$\text{Precision} = \frac{TP}{FP + TP} \tag{8}$$

where FP represents False Positives, and TP represents True Positives.

3) RECALL

Recall (Sensitivity) calculates the ratio of the actual positive instances that were predicted correctly. It shows how closely the model fits all the instances of the particular class.

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{9}$$

where TP represents the True Positives, and FN represents the False Negatives.

4) MACRO F1-SCORE

Macro F1 Score is the average of the F1 Score achieved for each class and F1 Score is the harmonic mean of precision and recall, which is a common measure to balance the two. The Average F1-Measure is calculated as:

$$\text{Average F1-Measure} = \frac{1}{N}\sum_{i=1}^{N} \text{F1-Measure}_i \tag{10}$$

where N represents the total number of categories, and the F1-Measure for each category is defined by the following equation:

$$\text{F1-Measure}_i = 2 \times \frac{\text{Exactness}_i \times \text{Sensitivity}_i}{\text{Exactness}_i + \text{Sensitivity}_i} \tag{11}$$

5) COHEN'S KAPPA

Using Cohen's Kappa, it is possible to find out a measure (statistical) that shows the extent of agreement between two raters or models beyond chance. Unlike accuracy, it takes into consideration the random occurrence of the agreement and is therefore a better measure. The value of Cohen's Kappa is between -1 and 1, where -1 indicate full disagreement and 1 shows perfect agreement, and 0 means agreement can be done by chance.

$$\kappa = \frac{A_o - A_e}{1 - A_e} \tag{12}$$

where A_o is the actual agreement (accuracy), and A_e is the expected agreement by chance, which is calculated based on the distribution of class frequencies in the data.

The interpretation of κ is as follows: When $\kappa = 1$, it indicates complete agreement between the raters or models. If $\kappa = 0$, the agreement is similar to what would be expected by random chance. A value of $\kappa < 0$ suggests that the agreement is worse than random chance.

Cohen's Kappa is especially important when comparing two models' performance across two different datasets with unequal class sizes or in the case of imbalanced classes since it is a better measure than the accuracy score alone.

6) ROC CURVE

The ROC (Receiver Operating Characteristic) curve is a graphical representation of a model's performance on the entire range of classification thresholds. It depicts the relationship between True Positive Rate (Recall) and False Positive Rate (1- Specificity). Area under the curve (AUC) is used to measure the performance of the model, where AUC of 1 is perfect model and 0.5 is a random classifier. As defined in Equation (9), the True Positive Rate (TPR) is equivalent to Recall. Equation for False Positive Rate is as under.

Fall-out, or False Positive Rate (FPR), is the proportion of actual negatives that are incorrectly identified as positives by the model. The formula for Fall-out is given by:

$$\text{Fall-out (FPR)} = \frac{FP}{FP + TN} \tag{13}$$

where FP represents false positives, and TN denotes true negatives.

A model with an ROC curve that is closer to the top left corner of the graph is considered better. AUC which is the Area Under the Curve, is a value between 0.5 and 1, where 0.5 means random and 1 perfect separation.

D. BASELINE ARCHITECTURE

In order to design a strong model for multi-class classification of hate and offensive speech in Urdu language, a step by step and hierarchical approach was used. The process began with the identification and evaluation of several learning models based on the training of each model separately. These initial experiments were useful for establishing the baseline performance and to understand the relative advantages and disadvantages of each model type. The lessons learnt from these assessments helped in the development of a complex ensemble model. Thus, taking the advantages of multiple transfer learning models, the ensemble model proposed in this paper reached a higher classification accuracy. The subsequent sections describe the process of building the baseline model and the proposed architecture of the ensemble model.

1) APPROACHES

The first stage of model selection was to test different standalone models to set some expectant criteria for the performance. To maintain the validity of the experimentation, 70% of the data is employed for training and 30% for testing purposes. The models used in this phase are the conventional Machine Learning approaches, cutting-edge Deep Learning models and latest Transfer Learning models. We evaluated the performance of classic machine learning models, i.e., Multinomial Naïve Bayes (MNB), Support Vector Machines (SVM), and Random Forest (RF) utilizing feature representation techniques including, Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF). Well-known neural network architectures such as Bi-directional Long Short Term Memory (BiLSTM) and Convolutional Neural Networks (CNNs) were employed using Word2Vec and FastText as the feature extraction methods. Transfer learning approaches were utilized to build upon the existing knowledge from large datasets. These include mBERT, XLM-RoBERTa, and ULMFiT which were fine-tuned for the respective classification tasks. These models used transfer learning and got knowledge from large diverse corpus and performed well in the task of identifying hate and offensive language in Urdu.

E. PROPOSED MODEL RATIONALE

Hate speech detection in a language like Urdu which is less explored in the field of research comes with several challenges that are known in the literature and must be addressed to deliver a good performance by the model. Some of these challenges include: High dimensionality, sparse data, overfitting, OOV words, and variation in dialects.

To address these issues, a transfer learning strategy based on ensemble methods that includes FastText, XLM-RoBERTa, ULMFiT and XGBoost was proposed. The choice of the models has been made based on the findings in literature about their effectiveness and potential to address the challenges of Urdu text classification. For ''High Dimensionality & Sparsity'', pre-trained word embeddings and XGBoost were utilized. XGBoost uses tree-based algorithm, which can effectively handle the high dimensionality and sparsity by selecting features that are most relevant to the model. For Out-of-Vocabulary words FastText was employed. Through the use of character n-grams, FastText has the ability to generate embeddings for words which are not seen by the model, thus improving the model's capacity to work with a wide range of textual data. XGBoost is also suitable at dealing

with overfitting as it uses regularization when the model is high capacity and the data set is small. Despite the fact that our methodology was developed with these challenges in mind, we would like to note that the current research does not contain direct evidence of the effectiveness of these particular solutions. Nonetheless, the gains in performance that are seen here indicate that these techniques are probably aiding in making the model more reliable.

F. SYSTEM ARCHITECTURE

A detailed architectural diagram of the proposed method is illustrated in Figure 5 which shows how the proposed model harness various transfer learning techniques through a series of layers that starts with feature extraction and ends at classification. The architecture of the proposed model is classified into three modules, namely, ''Feature Building'', ''Feature Stacking'' and ''Meta Classifier''. Feature Building layer produces dense and contextualized vectors from three models, i.e., FastText, XLM-RoBERTa, and ULMFiT. FastText utilizes subword information, XLM-RoBERTa serves dynamic contextual embeddings and ULMFiT offers task fine-tuned embeddings for Urdu. At Feature Stacking layer, the embeddings from previous layer are combined to generate a single feature vector that encompasses both low level and high level characteristics of the text. The concatenated feature vector is then fed into an XGBoost classifier for its effectiveness to handle high-dimensional data and avoiding overfitting. Finally, the classifier categorizes the text as ''Hate'', ''Offensive'', and ''Neither''.

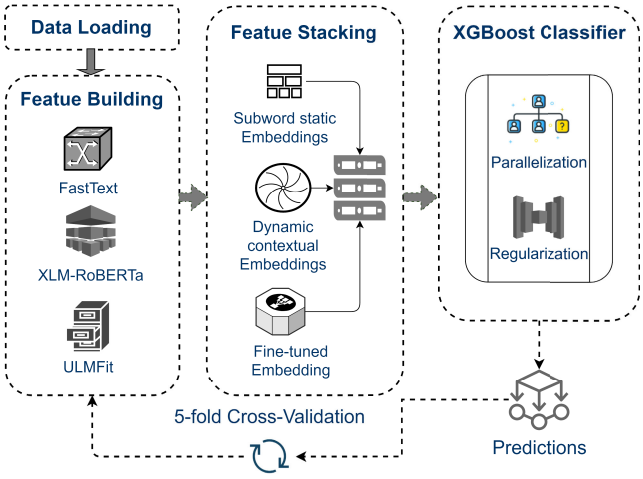


FIGURE 5. Proposed model architectural diagram.

The subsequent sections provide explanations of each module involved in the system architecture, their functions, and how they affect the overall system performance. Algorithm 2 illustrates the pseudo-code of the proposed ensemble method.

G. SYSTEM MODULES

The proposed method is characterized by several main components i.e. Feature building, Embedding stacking and Meta classifier, which jointly ensure the efficient classification of text data. The following sections present the description of each module in detail.

Algorithm 2 Pseudocode for the Proposed Diverse Embeddings Ensemble Framework

Input:
- Labelled dataset: 36k tweets labelled as ''Hate'', ''Offensive'', and ''Neither''
- FastText embeddings: cc.ur.300.vec

Output:
- Precision
- Recall
- F1-Score
- Accuracy
- ROC Curve
- Confusion Matrix
- Cohen's Kappa

Pre-processing:
for *each tweet in the dataset* **do**
  Tokenize the text data;
  Extract FastText embeddings;
  Extract XLM-RoBERTa embeddings using the pre-trained transformer model;
  Fine-tune ULMFiT on labelled data;
  Extract ULMFiT embeddings;
end for

Concatenate Embeddings:
Combine FastText, XLM-RoBERTa, and ULMFiT embeddings;

Stratified 5-Fold Cross-Validation:
for *each fold in 5-fold stratified cross-validation* **do**
  Train XGBoost classifier on training data;
  Predict on validation data;
  Produce Accuracy, Precision, Recall, F1-Score;
  Generate confusion matrix, ROC curve and Cohen's Kappa;
end for

Average Metrics Across Folds:
Compute mean accuracy, precision, recall, and F1-score;

Visualization:
Plot normalized confusion matrix;
Plot ROC curve for each class;
Draw stacked bar chart to illustrate Cohen's Kappa statistics;

1) FEATURE ENGINEERING

This module incorporate FastText, XLM-RoBERTa, and ULMFiT to generate an extensive set of embeddings described in Section III-B. These embeddings play a very important role in improving the model's performance and encompasses both the word level and contextual information. FastText Embeddings are subword-level embeddings that are effective in addressing the issue of Out-of-Vocabulary words. (see Section III-B4). XLM-RoBERTa Embeddings are contextual word embeddings that has the capability to capture context-specific semantics. (see Section III-B5). ULMFiT Embeddings are fine-tuned embeddings that can better capture the domain-specific patterns (see Section III-B6).

This next module combines these embeddings to create a stacked feature representation that will be used in subsequent phases.

2) FEATURE STACKING

At this layer the word embeddings from XLM-RoBERTa, ULMFiT and FastText are combined in a single feature vector for each of the input text. This approach leverages the distinct strengths of each model to get a rich and detailed representation of the text, including both the overall context and the semantics that is specific to the context. This enhanced and hybrid feature vector is then passed on to the meta-classifier (XGBoost) which in turn provides enhanced classification accuracy in terms of all the metrics utilized and particularly in identifying the grey areas of hate and offense in the Urdu language.

3) META CLASSIFIER

For the final layer, we have decided to use XGBoost as our meta-classifier in order to combine the embeddings coming from the previous layer. XGBoost is one of the most effective algorithms in handling complex data due to its ability to control high-dimensional data. It uses the boosting method where a series of decision trees is built, and each tree is designed to learn from the mistakes of the previous trees. This way the model refines its predictions in iterative manner on the basis of previous results to improve the predictive power. In addition, XGBoost uses regularization techniques which makes the algorithm less prone to overfitting and hence improve the performance when dealing with new data. Finally, the meta-classifier leverages these learned patterns from the combined embeddings to effectively categorize the input text into ''Hate'', ''Offensive'', and ''Neither'' categories.

4) TRAINING

The proposed model was validated on a stratified 5-fold cross-validation method to ensure that the results are statistically applicable to any subset of the dataset. In this approach the dataset was divided into five equal sets termed as folds. In each round, the model was trained on 4 fold of the data and validated on the last fold. This process was done five times where each fold was used in turn as the validation data set. The training and validation process was performed in a way that included cross-validation and hence helped in reducing the chances of over-fitting and also gave a view of how the model will perform on all the divisions of the data.

XGBoost classifier was tuned with the best set of hyperparameters in order to improve the performance. Different sets of hyperparameters were adjusted in order to determine the best overall configuration.

IV. EXPERIMENTATION

This section highlights the dataset upon which the experiments were conducted, the models, metrics used to assess the results, hyper-parameters, and the environment in which the experiments took place.

A. DATASET INFORMATION

In this research the data was gathered from Twitter platform using their official API. Tweets were classified into ''Hate'', ''Offensive'' and ''Neither''. The entire corpus contains 36,000 labelled tweets. For details see Section III-A.

B. MODELS

We compared several baseline models with our proposed ensemble model for the task of classification including, SVM, Random Forest, Naive Bayes, BiLSTM, CNN, mBERT and XLM-RoBERTa. Various types of embedding were utilized, including BoW, TF-IDF, FastText, Word2Vec, and self-generated embeddings from transformer-based models. ULMFiT was utilized for fine-tuning during experimentation of the ensemble learning approach.

C. EVALUATION METRICS

Various metrics were used to evaluate the performance of each model, i.e., Precision, Recall, Accuracy, Macro F1-score, Cohen's Kappa and ROC. Stratified 5-fold cross-validation technique was incorporated to validate each model and reduce the possibility of overfitting. For detail description of each metric see Section III-C

D. TRAINING SETUP

The experiments were performed on Google Colab, an onilne platform backed by Google, which provided access to free and paid GPU. The dataset was stored in Google Drive and the main programming language adopted for the experiments was Python. Some of the libraries and tools utilized in this research work are: numpy, sklearn, pandas, pytorch, transformers, fastai, scikit-learn, matplotlib, seaborn, tensorflow.

E. HYPER-PARAMETERS

Table 6 lists the hyper-parameters used in respective models and were set after performing experimentation and tuning the model to get the best results.

V. RESULTS

This section presents the performance of baseline methods and proposed model, using different performance metrics i. e. precision, accuracy,recall, and macro f1-score. Only the best performing models which are highlighted bold in Table 7 were considered as a reference point to evaluate the classification accuracy of the proposed method. Different feature representations techniques were employed such as BoW, TF-IDF, pre-trained Word2Vec & FastText embeddings. For better visualization of results, confusion matrices, ROC curves and grouped bar charts for Cohen's Kappa are employed. Comparative analysis and summary of results are presented in a tabular format. Lastly, discussion of results is

TABLE 6. Hyper-parameters for key models.

Model	Hyper-parameter	Value
SVM	Kernel	RBF
	C (Penalty parameter)	1.0
	Gamma	scale
	Tolerance	1e-3
Random Forest	Number of Estimators	100
	Max Depth	None
	Min Samples Split	2
Naive Bayes	Distribution	Multinomial
	Smoothing (Alpha)	1.0
BiLSTM	Learning Rate	1e-3
	Epochs	10
	Batch Size	32
	Dropout Rate	0.5
CNN	Learning Rate	1e-4
	Epochs	10
	Batch Size	64
	Dropout Rate	0.25
mBERT	Learning Rate	2e-5
	Epochs	4
	Optimizer	AdamW
	Regularization	L2 Regularization
XLM-RoBERTa	Learning Rate	2e-5
	Epochs	20
	Optimizer	AdamW
	Regularization	L2 Regularization
ULMFiT	Learning Rate	1e-3
	Epochs	20
	Optimizer	Adam
	Regularization	Dropout (0.5)
XGBoost	Learning Rate	0.3
	Number of Estimators	10
	Regularization	L2 Regularization

presented that provides an overview of the findings drawn from the analysis of results.

A. CONFUSION MATRIX ANALYSIS

Confusion matrix helps better visualize the details of classification outcomes, pinpointing areas where models may struggle, particularly with misclassifications between related categories e.g. hate and offensive. Due to the use of 5-fold cross-validation technique, normalized confusion matrix are used. The confusion matrices for top performing models are provided in Figure 6. The values are calculated in percentage to represent a normalize confusion matrix and provide the distribution of true labels predicted correctly and incorrectly. The following insights are reflected from the comparison of the baseline and proposed model confusion matrices. **Random Forest with TF-IDF** achieved fair performance, but had more difficulties in identifying Hate content from Offense (with 24.91% misclassification). The overall accuracy for Neither was 87.92%, 62.80% for Hate and Offensive was 74.22%. **BiLSTM with FastText** was seen to perform better compared to Random Forest particularly in the Offensive content detection, with an accuracy of 92.59%. Though, in case of Neither tweets its accuracy declined to 81.16% where there was some confusion between Neither and Hate tweets. On the other hand, it was much accurate for the Hate category with a score of 79.44%. **XLM-RoBERTa** greatly enhanced classification for all classes and greatly reduced the misclassifications. For the class Neither, it was able to reach an accuracy of 90.25%, for class Hate it was at 85.22% and for the class Offensive it had a very good accuracy of 95.63%. This performance gain can be associated with the enhanced capability of the transformer model for capturing the contextual information within Urdu text. **Proposed ensemble model** outperformed all the baseline models by achieving the highest accuracy for all the three categories: Highest having 96.53% for Offensive then 94.62% for Neither and 91.13% for Hate. This enhancement stems from the strengths of the integrated models.

B. ROC CURVE ANALYSIS

In addition to confusion matrices, ROC curve helps in visualizing the model's ability to differentiate a given class from others in terms of better classification accuracy. The ROC curves in Figure 6 give further understanding of how the models are able to classify the data. **Random with TF-IDF** had an average AUC of 0.91, Neither had the best performance with a classification AUC of 0.94 while there was slight difficulty in discriminating between Hate content which had a classification AUC of 0.85. **BiLSTM with FastText** can be seen to have increased average AUC of 0.96 and specific emphasis in the detection of Offensive content with an AUC of 0.98. **XLM-RoBERTa** perform well across the board with an average AUC of 0.98. Offensive content was classified exceptionally well with an AUC of 0.99. **Proposed Model** achieved near perfect results with the overall mean AUC of 0.99 and the AUC of 1.00 for the Offensive content category.

C. COHEN'S KAPPA ANALYSIS

Cohen's Kappa, assesses the agreement between the predicted and actual label across all the classes. In Figure 7, a bar chart is provided to compare Cohen's Kappa scores for both the baseline models and the ensemble model proposed. **Random Forest with TF-IDF** shows 62.49% agreement and a 37.51% disagreement, which can be regarded as moderate results. **BiLSTM with FastText** has better agreement at 77.71% showing an increase in the level of classification accuracy. **XLM-RoBERTa** has a substantial increase in the level of agreement of 85.58%, which points to high model validity. **Proposed Model** has highest agreement of 91.28%, which proves great performance of the model and its stable and precise classification.

D. DISCUSSION OF RESULTS

In Table 8, we have analyzed and compared the performance of the top performing models from each learning algorithm. The discussion is based on the effectiveness of each model in different classes and assessment of the overall performance using metrics like accuracy, average AUC, and Kappa score. Random Forest with TF-IDF is fairly accurate for neither and offensive classes while being rather ineffective for hate class, which is more specific and complex than the other two. Despite this, the model got AUC (average) score of

TABLE 7. Performance comparison of different classifiers.

Approach	Classifier	Feature Type	Precision	Recall	Macro F1-score	Accuracy
ML	SVM	BoW	0.71	0.69	0.70	72.0%
	Random Forest	BoW	0.75	0.73	0.74	74.5%
	Naive Bayes	BoW	0.67	0.65	0.66	67.5%
	SVM	TF-IDF	0.74	0.71	0.72	73.0%
	Random Forest	**TF-IDF**	**0.75**	**0.75**	**0.75**	**75%**
	Naive Bayes	TF-IDF	0.70	0.68	0.69	71.0%
DL	BiLSTM	Word2Vec	0.80	0.78	0.79	79.0%
	CNN	Word2Vec	0.75	0.73	0.74	74.5%
	BiLSTM	**fastText**	**0.86**	**0.85**	**0.85**	**85%**
	CNN	fastText	0.77	0.75	0.76	76.5%
TL	mBERT	Self-generated embeddings	0.84	0.82	0.83	83.0%
	XLM-RoBERTa	**Self-generated embeddings**	**0.90**	**0.90**	**0.90**	**90%**
EL	**Proposed Model**	FastText, XLM-RoBERTa	**0.94**	**0.94**	**0.94**	**94%**

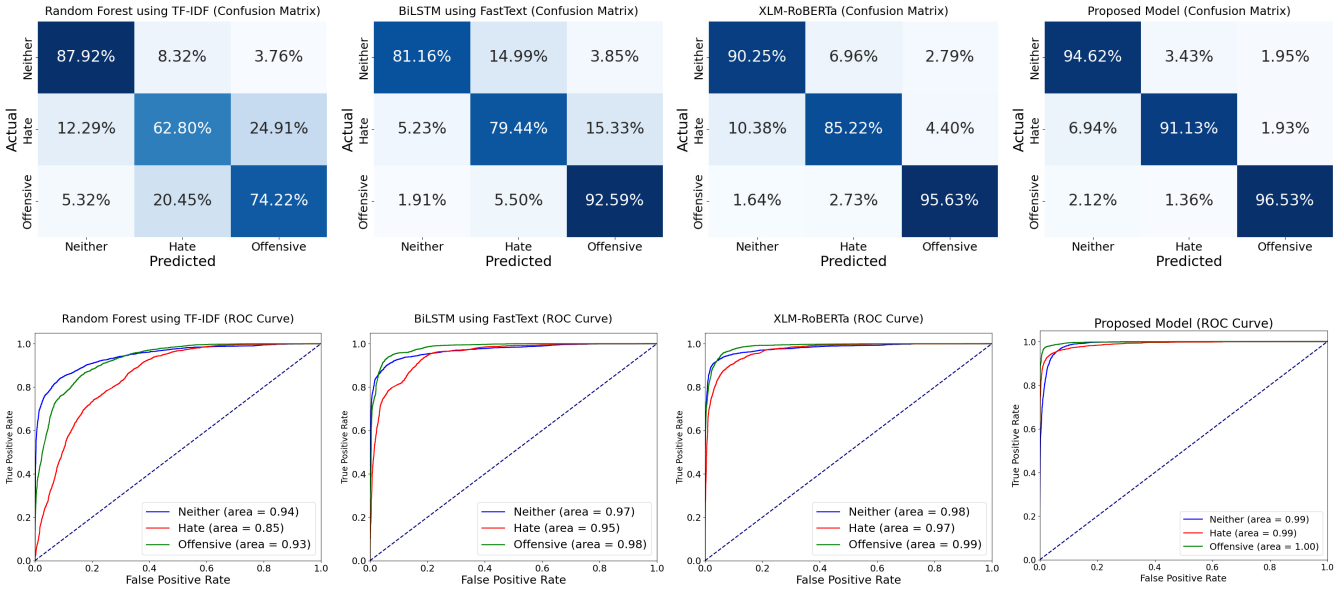


FIGURE 6. Confusion matrix and ROC (Receiver Operating Characteristic) curves of baseline & proposed ensemble model.

TABLE 8. Comparative analysis and summary of results.

Model	Accuracy				Avg AUC	Kappa
	Neither	Hate	Offensive	Overall		
Random Forest with TF-IDF	87.92%	62.80%	74.22%	75%	0.91	62.5%
BiLSTM with FastText	81.16%	79.44%	92.59%	85%	0.96	77.7%
XLM-RoBERTa	90.25%	85.22%	95.63%	90%	0.98	85.6%
Proposed Ensemble Model	**94.62%**	**91.13%**	**96.53%**	**94%**	**0.99**	**91.3%**

0.91, and its classification agreement is only moderate, as reflected by a Kappa score of 62.5%. This is not surprising, as most traditional machine learning methods solely count the frequency of words and may not be sophisticated enough to capture the underlying context for classification.

In contrast, the BiLSTM with FastText embeddings offers a considerable gain, especially in the hate and the offensive content classification. Offensive content accuracy is enhanced to 92.59% while the Kappa score rises to 77.7% to show the model has become more reliable for classification. However, it has been observed that the model is still performing slightly lower than Random Forest in the case of "Neither" class, which suggests that there might be some difficulty in distinguishing "Neither" from "Hate" and "Offensive". The XLM-RoBERTa enhanced the classification accuracy among all categories. The model particularly does well in predicting hate speech with a 85.22% accuracy and a Kappa score of 85.6% which shows great consistency. As evidenced by the AUC score of 0.98, XLM-RoBERTa can be considered to have a high ability in discriminating between classes, which would be useful to

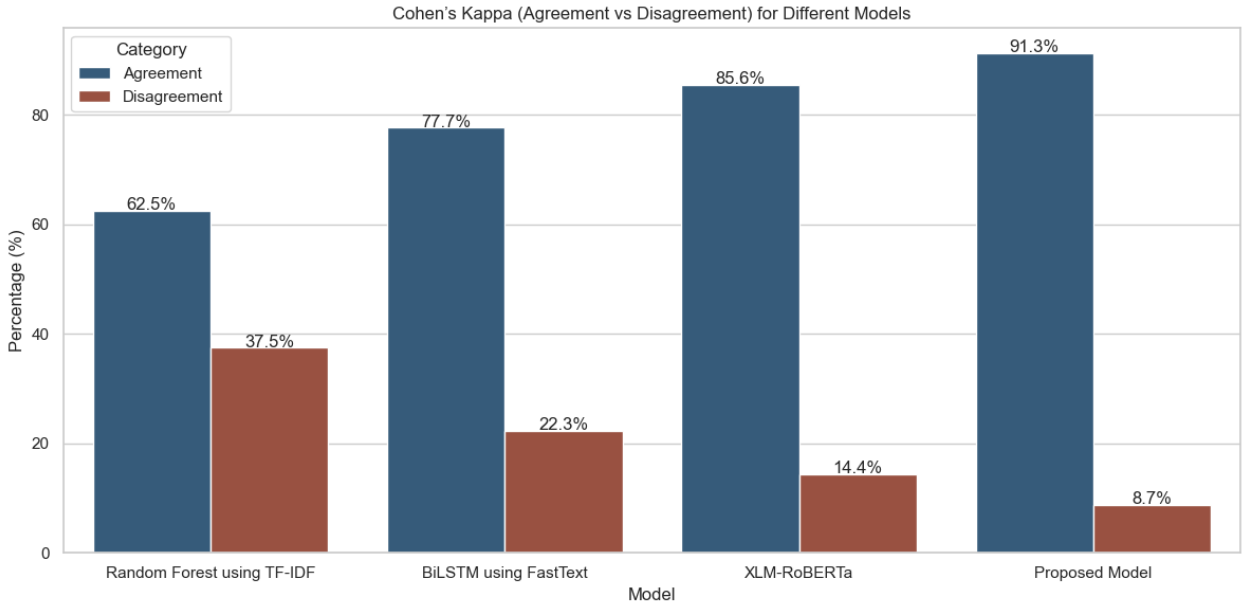


FIGURE 7. Cohen's Kappa scores for baseline & proposed ensemble model.

integrate it in the ensemble model proposed for multi-class classification.

Last but not least, the proposed ensemble model performs better than all baseline models in almost all aspects. It has the best classification performances for hate, offence, neither categories, with AUC of 0.99, a proof of very good classification performance. The model achieved a Kappa score of 91.3%, thus proving that the model was in strong agreement with all classes.A common problem that affected all of the baseline models when trained individually was the inability to properly differentiate between "Hate" and "Offensive" classes since they both presented similar types of language. This was an issue of concern since the architectures had their weaknesses that made them fail to perform optimally when applied independently. This issue was well handled by the ensemble model as it takes the best features from multiple architectures and utilize them for gaining better results.

VI. CONCLUSION

This research work contributed a newly created, manually labeled Urdu dataset comprising 36,000 tweets. The raw data was collected directly from Twitter using its API. It introduced an ensemble-based multi-classification model to classify Nastaliq-scripted Urdu into three distinct classes: "Hate", "Offensive" and "Neither". The model incorporated cutting-edge transfer learning methods and diverse embeddings to mitigate common NLP challenges. The proposed model significantly outperformed the baseline models, achieving macro F1-score of 0.94.

In this research, only those models were integrated into the proposed ensemble framework which satisfied two main criteria: the competency in terms of classification accuracy and the ability to overcome NLP challenges. To address these challenges and improve classification performance, this study combined the distinct features of FastText, XLM-RoBERTa, ULMFiT and XGBoost in an ensemble learning based architecture.

Advanced techniques were adopted to overcome these challenges such as high dimensionality, sparsity, overfitting and OOV words. To manage high dimensionality, this study incorporated XGBoost, which performs implicit feature selection by evaluating feature importance during training. It also provides solutions to the problem of overfitting by leveraging L1 and L2 regularization. One of the problems that low-resource languages are often characterized by is the lack of available data which results into sparse vector representation. To address this, XLM-RoBERTa pre-trained multilingual contextual embeddings were utilized, which generate dense, context-aware feature vectors. This approach enables the model to perform well even when there is sparse or limited data, making it ideal for low-resource languages and small datasets. These contextual embeddings also help overcome the challenge of dialectal variations in Urdu, enabling accurate classification across different dialects. ULMFiT is employed to further improve the model for Urdu language and to capture task specific details. FastText is integrated in ensemble model to overcome the issue of OOV words by utilizing sub-word level embeddings.

This research not only provides a robust ensemble-based multi-classification model but also contributes valuable resources that can further strengthens the foundation for future work in this field. For future studies, to enhance the model's generalizability, the dataset could be increased by including more diverse sources and additional languages. Additionally, by introducing multi-label classification could help avoid misclassification and give more precise content moderation while improving the overall detection system. Moreover, the use of explainability methods like SHAP

(SHapley Additive exPlanations) will further aid in understanding how the model makes decisions and therefore increase trust confidence in the model and its predictions.

DECLARATION

The authors declare no conflict for this research.

ACKNOWLEDGMENT

This research was supported by Princess Nourah bint Abdulrahman University and Researchers Supporting Project number (PNURSP2025R346). The authors would like to acknowledge the support of AIDA Lab CCIS Prince Sultan University, Riyadh, Saudi Arabia, for APC support.

REFERENCES

[1] J. Lu, H. Lin, X. Zhang, Z. Li, T. Zhang, L. Zong, F. Ma, and B. Xu, "Hate speech detection via dual contrastive learning," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 31, pp. 2787–2795, 2023.

[2] P. Piot, P. Martín-Rodilla, and J. Parapar, "MetaHate: A dataset for unifying efforts on hate speech detection," in *Proc. Int. AAAI Conf. Web Social Media*, vol. 18, 2024, pp. 2025–2039.

[3] L. Yuan, T. Wang, G. Ferraro, H. Suominen, and M.-A. Rizoiu, "Transfer learning for hate speech detection in social media," *J. Comput. Social Sci.*, vol. 6, no. 2, pp. 1081–1101, Oct. 2023.

[4] S. Gite, S. Patil, D. Dharrao, M. Yadav, S. Basak, A. Rajendran, and K. Kotecha, "Textual feature extraction using ant colony optimization for hate speech classification," *Big Data Cognit. Comput.*, vol. 7, no. 1, p. 45, Mar. 2023.

[5] A. Das, S. Nandy, R. Saha, S. Das, and D. Saha, "Analysis and detection of multilingual hate speech using transformer based deep learning," 2024, *arXiv:2401.11021*.

[6] P. H. Duong, T. T. Nguyen, and H. T. Nguyen, "Fusion network for multimodal hate speech detection," in *Proc. 9th Int. Conf. Intell. Inf. Technol.*, 2024, p. 1.

[7] M. U. Arshad, R. Ali, M. O. Beg, and W. Shahzad, "UHated: Hate speech detection in Urdu language using transfer learning," *Lang. Resour. Eval.*, vol. 57, no. 2, pp. 713–732, Jun. 2023.

[8] M. Z. Ali, Ehsan-Ul-Haq, S. Rauf, K. Javed, and S. Hussain, "Improving hate speech detection of Urdu tweets using sentiment analysis," *IEEE Access*, vol. 9, pp. 84296–84305, 2021.

[9] S. Nasir, A. Seerat, and M. Wasim, "Hate speech detection in Roman Urdu using machine learning techniques," in *Proc. 5th Int. Conf. Advancements Comput. Sci. (ICACS)*, Feb. 2024, pp. 1–7.

[10] S. Hussain, M. S. I. Malik, and N. Masood, "Identification of offensive language in Urdu using semantic and embedding models," *PeerJ Comput. Sci.*, vol. 8, p. e1169, Dec. 2022.

[11] L. Khan, A. Amjad, N. Ashraf, and H.-T. Chang, "Multi-class sentiment analysis of Urdu text using multilingual BERT," *Sci. Rep.*, vol. 12, no. 1, p. 5436, Mar. 2022.

[12] F. Mehmood, H. Ghafoor, M. N. Asim, M. U. Ghani, W. Mahmood, and A. Dengel, "Passion-net: A robust precise and explainable predictor for hate speech detection in Roman Urdu text," *Neural Comput. Appl.*, vol. 36, no. 6, pp. 3077–3100, Feb. 2024.

[13] 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, p. 3909, Apr. 2023.

[14] S. Aziz, M. S. Sarfraz, M. Usman, M. U. Aftab, and H. T. Rauf, "Geo-spatial mapping of hate speech prediction in Roman Urdu," *Mathematics*, vol. 11, no. 4, p. 969, Feb. 2023.

[15] F. Razi and N. Ejaz, "Multilingual detection of cyberbullying in mixed Urdu, Roman Urdu, and English social media conversations," *IEEE Access*, vol. 12, pp. 105201–105210, 2024.

[16] K. Jawad, M. Ahmad, M. Alvi, and M. B. Alvi, "RUSAS: Roman Urdu sentiment analysis system," *Comput., Mater. Continua*, vol. 79, no. 1, pp. 1463–1480, 2024.

[17] H. Saleem, M. Javed, S. M. A. Haider, H. M. Khan, M. A. Jan, and A. Ullah, "Framework of hate speech identification for formal and informal text using lexical approach," *Kurdish Stud.*, vol. 12, no. 1, pp. 5079–5094, 2024.

[18] M. S. Khan, M. S. I. Malik, and A. Nadeem, "Detection of violence incitation expressions in Urdu tweets using convolutional neural network," *Expert Syst. Appl.*, vol. 245, Jul. 2024, Art. no. 123174.

[19] F. Adeeba, M. I. Yousuf, I. Anwer, S. U. Tariq, A. Ashfaq, and M. Naqeeb, "Addressing cyberbullying in Urdu tweets: A comprehensive dataset and detection system," *PeerJ Comput. Sci.*, vol. 10, p. e1963, Apr. 2024.

[20] M. H. Akram, K. Shahzad, and M. Bashir, "ISE-hate: A benchmark corpus for inter-faith, sectarian, and ethnic hatred detection on social media in Urdu," *Inf. Process. Manage.*, vol. 60, no. 3, May 2023, Art. no. 103270.

[21] M. S. I. Malik, A. Nawaz, and M. M. Jamjoom, "Hate speech and target community detection in nastaliq Urdu using transfer learning techniques," *IEEE Access*, vol. 12, pp. 116875–116890, 2024.

[22] A. Q. A. Hassan, B. B. Al-Onazi, M. Maashi, A. A. Darem, I. Abunadi, and A. Mahmud, "Enhancing extractive text summarization using natural language processing with an optimal deep learning model," *AIMS Math.*, vol. 9, no. 5, pp. 12588–12609, 2024.

[23] A. Ullah, K. U. Khan, A. Khan, S. T. Bakhsh, A. U. Rahman, S. Akbar, and B. Saqia, "Threatening language detection from Urdu data with deep sequential model," *PLoS ONE*, vol. 19, no. 6, Jun. 2024, Art. no. e0290915.

[24] M. F. Bashir, A. R. Javed, M. U. Arshad, T. R. Gadekallu, W. Shahzad, and M. O. Beg, "Context-aware emotion detection from low-resource Urdu language using deep neural network," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 22, no. 5, pp. 1–30, 2023.

[25] D. Sharma, V. K. Singh, and V. Gupta, "TABHATE: A target-based hate speech detection dataset in Hindi," *Social Netw. Anal. Mining*, vol. 14, no. 1, p. 190, 2024.

[26] A. Chopra, D. K. Sharma, A. Jha, and U. Ghosh, "A framework for online hate speech detection on code-mixed Hindi-English text and Hindi text in Devanagari," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 22, no. 5, pp. 1–21, 2023.

[27] A. Nandi, K. Sarkar, A. Mallick, and A. De, "A survey of hate speech detection in Indian languages," *Social Netw. Anal. Mining*, vol. 14, no. 1, p. 70, 2024.

[28] K. E. Daouadi, Y. Boualleg, and O. Guehairia, "Systematic investigation of recent pre-trained language model for hate speech detection in Arabic tweets," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, 2024.

[29] A. Al Maruf, A. J. Abidin, M. M. Haque, Z. M. Jiyad, A. Golder, R. Alubady, and Z. Aung, "Hate speech detection in the Bengali language: A comprehensive survey," *J. Big Data*, vol. 11, no. 1, p. 97, 2024.

[30] G. Ramos, F. Batista, R. Ribeiro, P. Fialho, S. Moro, A. Fonseca, R. Guerra, P. Carvalho, C. Marques, and C. Silva, "Leveraging transfer learning for hate speech detection in Portuguese social media posts," *IEEE Access*, vol. 12, pp. 101374–101389, 2024.

[31] G. Uludogan, A. E. Yüksel, Ü. Tunçer, B. Isik, Y. Korkmaz, D. Akar, and A. Özgür, "Detecting hate speech in Turkish print media: A corpus and a hybrid approach with target-oriented linguistic knowledge," in *Proc. 7th Workshop Challenges Appl. Automated Extraction Socio-Political Events Text (CASE)*, 2024, pp. 205–214.

[32] N. B. Çam and A. Özgür, "Evaluation of ChatGPT and BERT-based models for Turkish hate speech detection," in *Proc. 8th Int. Conf. Comput. Sci. Eng. (UBMK)*, Sep. 2023, pp. 229–233.

[33] M. O. Ibrohim and I. Budi, "Hate speech and abusive language detection in Indonesian social media: Progress and challenges," *Heliyon*, vol. 9, no. 8, Aug. 2023, Art. no. e18647.

[34] O. S. Robert, G. C. Nwode, and B. Ugoala, "Hate speech, a source of linguistic, religious and ethnic intolerance among the sub-Saharan African peoples: The case of Nigeria," in *Sub-Saharan Political Cultures of Deceit in Language, Literature, and the Media, Volume II: Across National Contexts*. Cham, Switzerland: Springer, 2023, pp. 125–140.

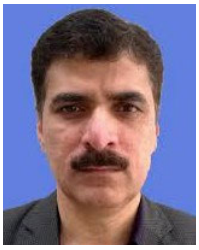
[35] I. Mollas, Z. Chrysopoulou, S. Karlos, and G. Tsoumakas, "ETHOS: A multi-label hate speech detection dataset," *Complex Intell. Syst.*, vol. 8, no. 6, pp. 4663–4678, Dec. 2022.

[36] K. M. Ali, T. A. Khan, S. M. Ali, A. Aziz, S. A. Khan, and S. Ahmad, "An exhaustive comparative study of machine learning algorithms for natural language processing applications," *Eng. Proc.*, vol. 76, no. 1, p. 79, 2024.

[37] M. A. Wani, M. ElAffendi, and K. A. Shakil, "AI-generated spam review detection framework with deep learning algorithms and natural language processing," *Computers*, vol. 13, no. 10, p. 264, Oct. 2024.

[38] A. Sabah, S. Tiun, N. S. Sani, M. Ayob, and A. Y. Taha, "Enhancing web search result clustering model based on multiview multirepresentation consensus cluster ensemble (mmcc) approach," *PLoS ONE*, vol. 16, no. 1, Jan. 2021, Art. no. e0245264.

KIFAYAT ULLAH received the B.S. degree in electrical computer engineering from COMSATS University Islamabad, Abbottabad, Pakistan, in 2013. He is currently pursuing the master's degree with the Department of Computer Science, University of Engineering and Technology (UET), Lahore, Pakistan. He is also serving as Deputy Director (IT) at the Power Information Technology Company (PITC), Ministry of Energy (Power Division). He has over ten years of experience in software design and development, team leadership and building. His research interests include natural language processing, transfer learning, and the development of advanced models for multilingual text analysis. He was awarded the ICT Research and Development Fund Scholarship for the B.S. studies, from 2009 to 2013.



MUHAMMAD ASLAM received the Ph.D. degree in computer sciences from CINVESTAV-IPN, Mexico, in 2005. He has more than 15 years of experience in software architecture design, team leading, team building, and software project; and 14 years of experience in research and development and teaching at postgraduate level (supervising Ph.D. and M.Sc. thesis). He won the merit scholarship from the Board of Intermediate and Secondary Education, Sargodha Division, Pakistan, from 1984 to 1986. He was also received the Cultural Exchange Scholarship between Pakistan and Mexico, from 2000 to 2004, for the Ph.D. studies. He already has supervised six Ph.D. and more than 50 master's thesis and final year projects. He also won the research grants from HEC and UET. He has published his research findings in number of well reputed impact factor journals of Springer, IEEE, and Elsevier. His research and teaching interests include artificial intelligence, distributed intelligence, knowledge-based systems, expert systems, intelligent agents, human–computer interaction, machine learning, computer-supported cooperative work, cooperative writing and authoring, cooperative learning, and distributed computing. He won Silver Medal from the Faculty of Agricultural Engineering, University of Agricultural, Faisalabad, Pakistan, from 1987 to 1991.

MUHAMMAD USMAN GHANI KHAN received the Ph.D. degree from Sheffield University, U.K., concerned with statistical modeling for machine vision signals, specifically language descriptions of video streams. He is currently an Associate Professor with the Department of Computer Science, University of Engineering and Technology Lahore. He has been studying on spoken language processing using statistical approaches with applications, such as information extraction from speech and speech summarization. His recent works are concerned with multimedia, incorporating text, and audio and visual processing into one frame work.

FATEN S. ALAMRI received the Ph.D. degree in system modeling and analysis in statistics from Virginia Commonwealth University, USA, in 2020. Her Ph.D. research included Bayesian dose-response modeling, experimental design, and nonparametric modeling. She is currently an Assistant Professor with the Department of Mathematical Sciences, College of Science, Princess Nourah Bint Abdulrahman University. Her research interests include spatial area, environmental statistics, and brain imaging.

AMJAD REHMAN KHAN (Senior Member, IEEE) received the Ph.D. degree, specializing in information security using image processing techniques from the Faculty of Computing, Universiti Teknologi Malaysia (UTM), Malaysia, in 2010. He is currently an Associate Professor with CCIS, Prince Sultan University, Riyadh, Saudi Arabia. He is also a PI in several projects and completed projects funded by MoHE, Malaysia, Saudi Arabia. His research interests include bioinformatics, the IoT, information security, and pattern recognition. He received the Rector Award for the 2010 Best Student from UTM.