IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 28 January 2024, accepted 17 February 2024, date of publication 26 February 2024, date of current version 14 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3370232

RESEARCH ARTICLE

Abusive Language Detection in Urdu Text: Leveraging Deep Learning and Attention Mechanism

ATIF KHAN[1], ABRAR AHMED[1], SALMAN JAN[2,3], MUHAMMAD BILAL[1], AND MEGAT F. ZUHAIRI[2], (Senior Member, IEEE)
[1]Department of Computer Science, Islamia College Peshawar, Peshawar 25120, Pakistan
[2]Malaysian Institute of Information Technology, Universiti Kuala Lumpur, Kuala Lumpur 50250, Malaysia
[3]Department of Computer Science, Bacha Khan University Charsadda, Peshawar 24540, Pakistan

Corresponding authors: Megat F. Zuhairi (megatfarez@unikl.edu.my) and Salman Jan (salman.jan@unikl.edu.my)

ABSTRACT The widespread use of the Internet and the tremendous growth of social media have enabled people to connect with each other worldwide. Individuals are free to express themselves online, sharing their photos, videos, and text messages globally. However, such freedom sometimes leads to misuse, as some individuals exploit this platform by posting hateful and abusive comments on forums. The proliferation of abusive language on social media negatively impacts individuals and groups, leading to emotional distress and affecting mental health. It is crucial to automatically detect and filter such abusive content in order to effectively tackle this challenging issue. Detecting abusive language in text messages is challenging due to intentional word concealment and contextual complexity. To counter abusive speech on social media, we need to explore the potential of machine learning (ML) and deep learning (DL) models, particularly those equipped with attention mechanisms. In this study, we utilized popular ML and DL models integrated with attention mechanism to detect abusive language in Urdu text. Our methodology involved employing Count Vectorizer and Term Frequency-Inverse Document Frequency (TF/IDF) to extract n-grams at the word level: Unigrams (Uni), Bigrams (Bi), Trigrams (Tri), and their combination (Uni + Bi + Tri). Initially, we evaluated four traditional ML models—Logistic Regression (LR), Gaussian Naïve Bayes (NB), Support Vector Machine (SVM), and Random Forest (RF)—on both proposed and established datasets. The results highlighted that RF model outperformed other conventional models in terms of accuracy, precision, recall, and F1-measure on both datasets. In our implementation of deep learning models, we employed various models integrated with custom fastText and Word2Vec embeddings, each equipped with an attention layer, except for the Convolutional Neural Network (CNN). Our findings indicated that the Bidirectional Long Short-Term Memory (Bi-LSTM) + attention model, utilizing custom Word2Vec embeddings, exhibited improved performance in detecting abusive language on both datasets.

INDEX TERMS Abusive language, Bi-GRU, Bi-LSTM, deep learning models, fastText, GRU, LSTM, NLP, TF/IDF, Urdu, Word2Vec.

I. INTRODUCTION

The internet is accessible to nearly sixty-two percent of the world's population[1], giving rise to social media and various online communication platforms used by 4.95 billion people globally[2]. With such an extensive user base, millions of comments flood these platforms daily. The popular social network X (formerly known as Twitter) had 368 million active

The associate editor coordinating the review of this manuscript and approving it for publication was Yizhang Jiang.

[1]https://data.worldbank.org/indicator/IT.NET.USER.ZS

[2]https://www.statista.com/statistics/617136/digital-population-worldwide/

users in 2022, generating around 200 billion X posts (tweets) annually.[3] Similarly, Facebook users collectively shared an average of 4.75 billion posts per day [1]. However, not all comments on such platforms are inspirational. Some comments, whether intended or unintended, insult audiences based on gender, race, religion, communities, or opposing political groups, etc. Such discussions commonly deteriorate into abusive language and involve negative comments about individuals or groups. They may arise from differing opinions, political and religious debates, and various controversial issues.

Abusive speech causes harm and annoyance. It can lead to severe anxiety, depression, or other mental illnesses in serious cases, and is a major source of stress. These effects can range from acute emotional responses such as anger or self-recrimination to long-term psychological damage that may result in low self-esteem and melancholy, provoking various minor psychological disturbances including sleeping difficulties, headaches, and anorexia [2]. This negative discussion contributes to making the social media environment less welcoming, necessitating measures to detect and filter abusive content. This step is crucial not only for individuals' well-being but also for the sustainability of social media platforms. One of the most notable incidents occurred in 2013 when approximately two hundred thousand users signed a petition, and several large companies withdrew their advertisements from Facebook. This action was in response to pages promoting violence against women, including messages like 'Encouraging people to perform violent acts on friends just for fun' or 'Advocating beating up girlfriends who refuse to make sandwiches' [3].

The sheer volume of daily social media comments makes it unfeasible to manually detect and filter out abusive language. This has prompted researchers to explore the application of various Natural Language Processing (NLP) techniques for automated detection of abusive language. However, achieving this remains challenging; it is not as straightforward as merely identifying abusive keywords, as intentional word obfuscation weakens the effectiveness of direct keyword detection [3].

Abusive language detection is a context-sensitive issue that can be handled through intelligent detection models. Words and expressions deemed offensive in one community might be deemed acceptable in another. In recent years, machine learning (ML) and deep learning (DL) models employing NLP techniques have demonstrated remarkable efficacy in addressing problems related to abusive language detection or hate speech. In this study, we utilized traditional ML methods as a benchmark and observed the improvements over the baseline achieved by DL models equipped with attention mechanisms. According to the literature, most work has been done on resource-rich languages like English. However, very little research focuses on languages with limited resources, such as Urdu. Urdu ranks approximately as the world's 10th most spoken language, with a total of about 230 million speakers.[4] It serves as the official language of Pakistan and is also spoken in India, Afghanistan, Bangladesh, and Nepal. Urdu employs a script derived from Arabic and Persian, written from right to left. Diacritical marks are used to distinguish vowels and unique sounds among the 35 letters of Urdu.

Previous studies focused primarily on identifying abusive language in languages with abundant resources, such as English. However, this study addresses the challenge of identifying abusive language in resource-scarce languages, specifically concentrating on Urdu. Thus, this research fills a gap and extends the scope of prior studies by emphasizing the significance of combating abusive language in Urdu text.

The absence of existing Urdu datasets for abusive language detection has hindered progress in this domain. To overcome this deficiency, we established a new dataset called the "Dataset of Urdu Abusive Language" (DUAL), comprising 12,082 X posts (tweets) in Urdu, consisting of 6,141 abusive and 5,941 neutral instances. This novel and expansive dataset makes a significant contribution to the broader research community, providing valuable resources for evaluating and refining models tailored for detecting abusive language in Urdu.

The literature emphasizes the use of ML and DL models in identifying abusive language, yet thorough assessments and comparisons, particularly in Urdu text, remain predominantly undiscovered. This research performs a systematic analysis of diverse models, covering traditional approaches such as Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), Gaussian Naïve Bayes (NB), and DL models integrating attention mechanisms such as Gated Recurrent Unit (GRU), Bidirectional Gated Recurrent Unit (Bi-GRU), Long Short-Term Memory (LSTM), and Bidirectional Long Short-Term Memory (Bi-LSTM). This research substantially enriches the existing knowledge base, enabling experts to discern the most appropriate model for abusive language detection in Urdu. Previous studies have not given sufficient attention to the influence of feature extraction techniques on the efficacy of models. Within this study, we rigorously examined the impact of various feature extraction methods, including TF/IDF, Count Vectorizer, and Word Embeddings such as Word2Vec and fastText. Our research aims to enhance the effectiveness of both ML and DL models by thoroughly evaluating their performance with diverse feature extraction techniques.

The following contributions best describe our work:

- This study highlights the importance of tackling abusive language, particularly within Urdu text, while extending the scope of previous research to cover a more extensive linguistic context.
- This study created a new dataset, "Dataset of Urdu Abusive Language" (DUAL), which includes 12,082 X posts (tweets) in Urdu. Among these, 6,141 were annotated as abusive and 5,941 as neutral, providing a valuable resource for research on abusive language detection.

[3]https://www.statista.com/statistics/303681/twitter-users-worldwide/

[4]https://www.statista.com/statistics/266808/the-most-spoken-languages-worldwide/

- This work evaluated and compared traditional ML, DL, and DL models coupled with attention mechanism using the proposed DUAL and established datasets. The aim was to identify the most effective models for detecting abusive language in Urdu text.
- This study explored the impact of various feature extraction methods and embeddings on the effectiveness of traditional ML, DL, and DL models coupled with attention mechanism for detecting abusive language in Urdu.

II. RELATED WORK

This section reviews prior research conducted in the domain of abusive language detection, which represents a relatively recent area of study with limited investigations conducted in this domain. The Urdu language, having a restricted digital presence, suffers from a scarcity of annotated corpora. Some researchers have explored this subject and proposed various methods for detecting abusive language, utilizing conventional ML and DL algorithms for this purpose. Most research efforts have been directed towards resource-rich languages, neglecting low-resource languages like Urdu. Urdu comments can be composed in two distinct ways: Roman Urdu (RU), which employs the English alphabet, and Urdu script which uses the Urdu alphabet. However, researchers have only conducted limited studies on both formats of Urdu comments.

The authors of [4] presented a lexicon-based method to identify hate discourse. Their approach focused on identifying subjectivity in sentences and constructing a lexicon consisting of hate-related words, utilizing a rule-based method. Then, they trained a classifier using features derived from this lexicon and evaluated its effectiveness in identifying hate speech within documents. However, a significant drawback of this approach was its reliance on a lexical rule-based method, which disregarded the domain and context-specific usage of words in the text. The research by [5] explored the challenges involved in abusive language detection especially posed by complexities and subtleties of the natural languages, which cannot be easily understood by the machines. The authors explored how redundancy and the limited applicability of datasets can impact the performance of models. Moreover, they stressed the importance of building models that can generalize well across various domains of abusive language. The experimental results on a cross-domain datasets showed a significant finding: counterintuitively, increasing the size of a dataset with non-abusive samples can severely affect the model's ability to generalize.

ML and DL models have found extensive application in various classification tasks, including sentiment analysis, behavioral analysis [6], and counterfeit case detection [7]. Additionally, they have been employed to detect abusive language [8], [9], offensive language [10], cyberbullying [11], and threatening language in Urdu [12]. Studies conducted by [9] and [13] utilized traditional ML classifiers to detect abusive language in Urdu, achieving favorable outcomes. However, these traditional ML models exhibited strong performance primarily on smaller datasets. In a separate study, the authors of [11] proposed supervised ML techniques for detecting cyberbullying in Urdu X posts. Roman Urdu (RU), which employs English alphabets to transcribe Urdu, remains the predominant writing style among the majority of social media users. Studies conducted by researchers of () [14] and [15] focused on identifying hate speech in Roman Urdu. Another investigation by [16] employed various DL models—such as Recurrent Neural Network (RNN) based models and Convolutional Neural Network (CNN)—to detect instances of cyberbullying in Roman Urdu. Additionally, authors in [17] explored multiple ML and DL models, employing features derived from TF/IDF and word embeddings like fastText, Word2Vec, and GloVe, specifically targeting the classification of toxic comments in Roman Urdu. Remarkably, the Bi-LSTM model demonstrated superior performance, achieving an outstanding accuracy of 0.95 in classifying toxic comments in Roman Urdu.

The study conducted by [18] utilized Bidirectional Encoder Representations from Transformers (BERT) text representations combined with Logistic Regression to classify racist language in French, achieving an accuracy of 0.79. The authors of [10] evaluated seventeen ML models based on n-grams at both the word and character levels for detecting offensive and abusive language in Urdu and Roman Urdu text. They observed that character tri-grams outperformed other character and word level n-grams. Furthermore, Logistic Regression emerged as the top-performing model on both Urdu and Roman Urdu datasets. Similarly, research presented by [19] employed ML and DL techniques to identify instances of offensive content on social media. The authors analyzed the performance of several classifiers on nine datasets, which included publicly available datasets and one collected through crowdsourcing. The results indicated that DL models, particularly CNN, outperformed traditional ML models and rule-based classifiers.

Research conducted by [20] introduced a new dataset, RU-HSD-30K, designed specifically for identifying hate speech in Roman Urdu on social media. The dataset underwent validation by experts. The study investigated the impact of lexical normalization on the performance of various models, revealing a substantial improvement in the performance of each model. Additionally, the research proposed a context-aware DL model that combined a Bi-LSTM + Attention model with custom Word2Vec embeddings for the task of hate speech detection in RU. Another study, undertaken by [21], employed fine-tuned BERT for detecting abusive language in Urdu, achieving an F1-measure of 88. The results highlighted that fine-tuning a pre-trained model within the same domain significantly enhanced performance compared to training a model from scratch. Similarly, in a study by [22], transformer-based models were utilized to detect hate speech in RU. The researchers employed pre-trained BERT models using transfer learning and conducted pre-training of BERT from scratch on the RU dataset. The findings indicated that fine-tuning pre-trained BERT models remarkably improved performance compared to models trained from scratch. Moreover, they directly implemented the Transformer model as a block layer, which proved to be the most effective and accurate in detecting hate speech. The research conducted by [23] utilized Logistic Regression, Random Forest, SVM and transformer-based models such as BERT, XLM-Roberta to identify abusive and threatening language in Urdu. The results indicated that soft voting technique for transformers achieved the best performance with an F1-measure of 0.86.

In another study by [24], a novel CNN based architecture named CNN-gram was introduced for the task of identifying abusive content in Roman Urdu. The effectiveness of the proposed model was assessed by comparing it against seven well-known baseline techniques using the RUHSOLD dataset. The outcomes illustrated that the proposed model exhibited greater robustness compared to the baseline approaches. Moreover, the authors of [25] developed the RUBERT model by retraining the English BERT on Roman Urdu text. The study also involved building the BERT model exclusively for Roman Urdu text from scratch. Upon comparing performance, it was discovered that the BERT model, pre-trained on English text and fine-tuned on Roman Urdu text through transfer learning, outperformed the BERT model trained solely from scratch on Roman Urdu.

Similarly, research conducted in languages other than Urdu has extensively employed both ML and DL techniques. For instance, the authors of [26] utilized deep neural networks incorporating attention mechanisms to identify offensive and hate speech in Arabic text. Another study, carried out by [27], focused on identifying abusive language in Arabic. Moreover, the authors of [28] concentrated on utilizing ML attention-based models to identify offensive language in Greek.

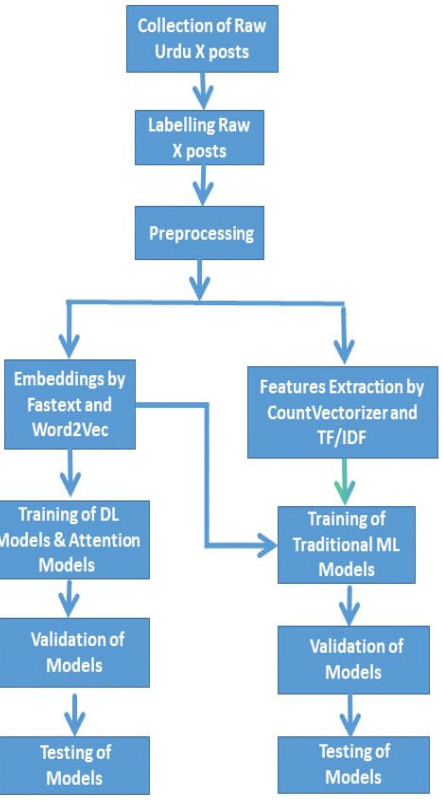
In addition, the authors of [29] conducted a study on offensive and hate speech detection in English. Similarly, the research conducted by [30] employed Support Vector Machines (SVM) and Long Short-Term Memory (LSTM) models to detect instances of hate speech in Italian. The study of [18] aimed to identify instances of racist discourse in French.

Furthermore, the study performed by [31] employed various ML techniques to identify hate speech and abusive language in the Indonesian language. Studies conducted by [32] and [33] explored the identification of offensive language in German, while [34] focused on detecting abusive language in Turkish within Instagram comments. Additionally, the research presented in [35] specifically aimed to identify offensive language in Turkish X posts (tweets). Finally, the work conducted by [36] and [37] focused on detecting abusive language in Russian.

III. METHODOLOGY

This section demonstrates the proposed method for detection of abusive language in Urdu text as depicted in Figure 1. The proposed method is composed of following steps:

A Dataset Collection
B Preprocessing
C Features Extraction/Embeddings
D Model's Training
E Model's Validation
F Model's Testing
G Model's Evaluation



Collection of Raw Urdu X posts
Labelling Raw X posts
Preprocessing
Embeddings by Fastext and Word2Vec
Features Extraction by CountVectorizer and TF/IDF
Training of DL Models & Attention Models
Training of Traditional ML Models
Validation of Models
Validation of Models
Testing of Models
Testing of Models

FIGURE 1. Proposed framework for abusive language detection in urdu tweets (X posts).

A. DATASET COLLECTION

Prior Urdu datasets, as indicated in Table 1, were limited in quantity. To address this gap, we collected additional X posts (tweets) and compiled a new dataset containing 12,082 Urdu tweets. Among these, 6141 tweets were classified as abusive, while 5941 were labeled as neutral. This dataset is named the 'Dataset of Urdu Abusive Language' (DUAL). To our knowledge, the DUAL dataset illustrated in Table 1 is the most comprehensive publicly accessible dataset in the literature for detecting abusive language in Urdu.

Initially, we collected roughly 20,000 tweets using the Tweepy API, with a specific focus on tweets containing abusive keywords. After performing data cleaning and preprocessing, the dataset was annotated by native Urdu-speaking experts. A group of five experts performed the annotations and were provided with the guidelines to label the dataset as Abusive or Neutral. We employed the Fleiss Kappa statistic in order to assess the level of agreement among all annotators. Values approaching +1 imply stronger agreement between the annotations. The final labeling decision is depended on identifying the highest inter-annotator agreement value. We assigned the majority label as the final one. If the Kappa value was low for instances, then we performed re-labeling based on established guidelines. The dataset comprises two classes: 0 for Neutral and 1 for Abusive Tweets (X posts), as depicted in Figure 2.

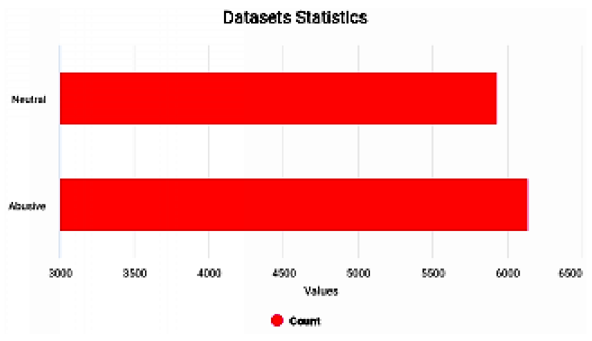


FIGURE 2. Statistics of proposed dataset.

TABLE 1. Abusive language datasets.

Ref.	Dataset Name	Domain	Hateful/ Abusive Comments	Neutral Comments	Size of Dataset
[1]	-	Offensive Language	3,750	3,750	7,500
[10]	UOD	Offensive Language	1109	1062	2171
[11]	-	Cyber bullying	-	-	7625
[21]	HASOC 2021	Abusive Language	1750	1750	3,400
[21]	HASOC 2021	Threatening Language	8160	1760	9,950
-	DUAL (Proposed)	Abusive Language	6141	5941	12,082

B. PREPROCESSING

In this phase, preliminary preprocessing is performed on the dataset in order to achieve optimal performance of the models. This phase comprises the subsequent steps.

X (Twitter) data contains noisy data like URLs, hashtags, mentions, emojis etc. The scraped data was cleaned from URLs, hashtags, mentions, Roman and Arabic digits. Duplicate X posts (tweets) were also removed from the data.

1) DATA CLEANIN

The dataset comprising X (Twitter) posts contains many types of noisy data, such as URLs, hashtags, mentions, emojis, and other elements. In order to cleanse the scrapped data, we eliminated all URLs, hashtags, mentions, as well as Roman and Arabic numerals. In addition, redundant X posts (tweets) were eliminated from the dataset.

2) WORDS TOKENIZATION

Each X post (tweet) is divided into word tokens using delimiters such as white space, colons, semicolons, tabs, periods, and commas.

3) STOP WORDS REMOVAL

Stop words are terms in a corpus that have limited impact on the effectiveness of a model. These words often occupy significant memory space and can extend processing time. Typically, articles, prepositions, conjunctions, and pronouns are classified as stop words. Removing stop words in NLP tasks is an important step that reduces the text size and vector space, hence reducing the processing time of the model.

4) FILTERING

In our dataset, we encountered several X posts (tweets) that had a minimal word count, often comprising just one or two words or a few characters. We omitted these tweets as they were inconsistent and did not contribute to enhancing the model's efficiency in classifying tweets. The remaining Urdu tweets, which are distinct, vary in length from 10 to 256 characters.

C. FEATURES EXTRACTION/EMBEDDING

Computers lack the ability to comprehend natural languages, photos, and videos. To facilitate computer's comprehension of such data, it needs to be translated into numerical representations. After completing data collection and preprocessing, the subsequent step involves feature extraction and embedding. During this phase, features are extracted from the training data, and numerical embeddings are assigned to each feature. Textual data is represented as a sequence of words, where the arrangement of words holds significant importance. The meaning of a word is influenced by its preceding and succeeding words, known as context. Several methods exist for transforming text into numerical values. For example, in word embeddings, each word is linked to a vector of values that represent various contexts within a given corpus. In this study, we employed a range of techniques, including Bag of Words (BoW), n-grams models, TF/IDF, and word embeddings (fastText and Word2Vec).

The Bag of Words (BoW) is a well know technique for extracting features from text data. It depicts text as a collection of individual words, disregarding the context and order of words. The BoW technique analyzes word frequencies in a document and preserves a lexicon that contains vocabulary words along with their respective counts. Each document in the corpus is represented as a numerical vector that designates the frequency of the words. However, the BoW approach lacks the capability to capture the semantic meanings of words.

In contrast, the Bag of n-grams method splits the text into continuous sequences of n words or phrases known as n grams. This method records the frequency of each n-gram in the text, capturing the contextual information. However, the vectors dimensions increase as the value of "n" increases. Unigrams are created from individual words, Bigrams are created from pair of words, and Trigrams are formed from three words, as illustrated in Table 2.

Term Frequency (TF) indicates the frequency of a word's occurrence in a document, while Inverse Document Frequency (IDF) computes the relevance of a word across all documents in the corpus. TF/IDF assigns a numerical value (weight) to a word based on its frequency in a document and its significance in the corpus.

Word embeddings express words with real-valued vectors that captures the contextual and semantic meaning of each word. Words with similar contexts are assigned similar vectors and are placed in close proximity in the vector space. The

TABLE 2. N-grams features.

N-gram	Example
Unigrams	کے 'تینو' ,'ہرامی','‘','بچے
Bigrams	کے 'تینو ہرامی ' ہرامی ‘ بچے کے '
Trigrams	کے 'تینو ہرامی ‘ بچے کے ہرامی ,
Uni + Bi + Trigrams	کے 'تینو' ,'ہرامی','‘','بچے کے 'تینو ہرامی ' ہرامی‘ بچے کے ' کے 'تینو ہرامی‘بچے کے ہرامی

context of a word is defined by the window size. TF/IDF and BoW lack the ability to capture the context or meaning of words. However, word embeddings have an advantage over TF/IDF and BoW in that they grasp the semantic meaning and context of words. This feature can be extremely beneficial in different NLP tasks, such as information retrieval or document classification.

D. MODEL'S TRAINING

In this phase, the dataset is partitioned into two portions, the training set and the testing set, in a proportion of 80:20. In order to achieve this task, we utilized the "train_test_split" function provided by the sklearn module in Python. All the models are trained using training set. The training set facilitates the model's training process, allowing it to learn and adjust its parameters, as well as discover hidden patterns/features in the data. During each epoch, all the models based on neural network are passed through several rounds, allowing them to learn from the same training data. This iterative process consistently improves the model's understanding of the complex and distinct features in the data.

E. MODEL'S VALIDATION

A 20% subset of the training data is allocated for the purpose of validation. The validation set is of critical importance during model training since it assists in fine-tuning the model's hyperparameters. In this work, we employed held-out cross-validation, a straightforward form of cross-validation that diminishes bias by assuring that the bulk of the data is utilized for model fitting. To keep track of the model's performance and progress throughout training and validation processes, we stored the accuracy and loss values in an object called "history" that can be visualized later.

F. MODEL'S TESTING

In this phase, a separate test set is provided to each trained model to evaluate its performance. This step aimed to assess how well the models generalize to new, unseen data. In order to prevent bias and overfitting in models, the test set is not utilized during the training process. The results obtained from evaluating each trained model against the test set are recorded for the purpose of analysis and comparison.

G. EVALUATION OF MODELS

In this phase, the performance of all the models was evaluated through well know evaluation metrics which are defined as follows:

Accuracy: It calculates the proportion of accurately predicted instances to the overall predicted instances.

Precision: It is calculated as the ratio of accurate positive predictions to the total positive predicted instances. It denotes the model' ability to reduce the false positive predictions.

Recall: It quantifies the model's capacity to correctly identify all true positive instances among the total instances that are truly positive. It maximizes the detection of true positives by minimizing the false negatives.

F1-Measure: This metric provides a fair assessment of a model's performance considering both precision and recall. It is very beneficial when working with imbalanced datasets when the number of instances in distinct classes changes significantly.

IV. EXPERIMENTAL SETTINGS

This study employed the Colab Pro Plus environment offered by Google, which provides support for Python 3. The allocated resources for this environment include a high-performance GPU with 500 compute units, 85 GB of RAM, and 200 GB of storage. All deep learning algorithms used in this study were implemented using the Keras library, employing the TensorFlow backend in the Python programming language. In addition, various Python libraries such as NLTK, Gensim, Numpy, Pandas, Sklearn and JSON were employed in this study.

This study performed a comparative analysis of traditional ML, DL models, and DL model coupled with attention mechanism to detect abusive language in Urdu. Word2Vec and fastText embedding techniques were utilized to extract features/embeddings from the proposed dataset. We extracted custom embeddings by training Word2Vec and fastText on our own abusive Urdu dataset.

The Word2Vec embedding technique employs a Word2Vec function to produces vector representations for individual words by considering their contextual information in an n-dimensional space. Terms that are semantically related exhibited a close proximity to each other within the vector space. The parameters of Word2Vec function includes a dimension size of 100 and a minimal count of 2. As a result, each word was represented with 100 dimensions vector.

The fastText embedding technique is also employed with same parameters as Word2Vec to obtain character level embeddings from proposed DUAL dataset. The word embeddings are obtained by taking mean of the vectors representing their character-grams.

The architecture of deep learning models includes an embedding layer at the top, utilizing fastText/Word2Vec embeddings with a vector size of 100 dimensions. The CNN model comprises a 1D-CNN layer with 32 filters, a kernel size of 8 and ReLU activation. Subsequently, a max-pooling layer of size 2 is added. For RNN/Bi-RNN, LSTM/Bi-LSTM and GRU/Bi-GRU models, the configuration uses a recurrent layer with 64 units and ReLU activation.

All models shared a common architecture, including a dropout layer of 0.2, three dense layers with sizes of (64, 32, 16 units), each using ReLU activation. During the training process of models, 'Adam' was employed as the optimizer while 'binary-cross-entropy' served as the loss function. Finally, a sigmoid function was applied for binary classification.

The attention models have similar structure to the recurrent models except for the addition of an attention layer after each recurrent layer. The epoch value is set to 500 for each model, followed by implementing early stopping with a patience of 20 epochs to monitor validation loss. If a model converges before reaching 500 epochs based on early stopping criteria, the training stops at that point. This guarantees effective monitoring and termination in circumstances where there is no improvement over a consecutive 20 epochs.

V. RESULT AND DISCUSSIONS

We analyzed the effectiveness of ML and DL algorithm by employing both our proposed dataset and a well-established dataset. The extra assessment enabled us to test the effectiveness of the models in an environment containing external variables like dataset bias and language traits.

We chose the HASOC 2021 Urdu Abusive dataset,[5] in order to perform a more thorough assessment of our models. This dataset has been broadly used in prior research studies [9], [12], [21] and [23] for the task of abusive language identification in Urdu; this illustrates its relevance and applicability to our current study.

The established dataset has 1187 abusive and 1213 non abusive instances. A total of 2400 instances were used for testing our models. The statistical details of the dataset are presented in Table 3. After completing the training and testing of models on our proposed dataset, we evaluated the models using the established dataset.

TABLE 3. Statistics of established dataset.

Category	Number of instances
Abusive	1187
Non-Abusive	1213
Total	2400

A. TRADITIONAL MACHINE LEARNING (ML) MODELS

We applied four traditional ML models namely Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM) and Gaussian Naïve Bayes (NB) for abusive language detection in proposed abusive Urdu dataset. We used n-grams at the word level: (Uni, Bi, Tri, Uni + Bi + Tri) as features using Count Vectorizer and TF/IDF. In addition, Word2Vec and fastText techniques were used for extracting word embeddings/features from the Urdu text. The dataset was divided into a ratio of 80:20, allocating 80% for training various models and reserving the remaining 20% for subsequent model testing. The results illustrated in Table 4 indicate that Random Forest and SVM, employing Unigrams and Uni + Bi + Tri-grams (with both TF/IDF and Count Vectorizer) surpassed other traditional models in accuracy and F1-measure, achieving values of 0.96 for both metrics. Logistic Regression secured second place with a score of 0.95 for both metrics (accuracy and F1-measure) using Unigrams and Uni + Bi + Tri-grams extracted with TF/IDF. Gaussian NB showed the least performance among the traditional ML models. A comparison of conventional ML models used in this study with findings from prior studies can be seen in Table 4.

[5]https://github.com/MaazAmjad/Urdu-abusive-detection-FIRE2021

TABLE 4. Comparison of our ML models with previous studies.

Previous Studies	ML Models Used	Best Performing ML Model	F1-Measure
Proposed Study	LR, RF, SVM, Gaussian NB	RF, SVM	**0.96**
[8]	LR, NB, SVM, JRIP, IBK	SVM	0.90
[9]	Decision Tree, Logistic Regression, Bagging Classifier	Bagging Classifier and Logistic Regression	0.83
[12]	KNN, LR, SVM	LR	0.72
[13]	SVM	SVM-Poly	0.83
[21]	XGBoost, LGBM	Both	0.76
[23]	LR, SVM, RF	LR	0.80

In essence, the ML models that utilized both Unigrams and Uni + Bi + Tri grams have produced nearly identical and best results when employing Count Vectorizer and TF/IDF, as shown Tables 5-6. However, the ML models that used Bigrams and Trigrams have yielded the worst results in terms of accuracy and F1-measure.

Moreover, we thoroughly tested the effectiveness of ML models in identifying abusive language using an established dataset. The results illustrated in Tables 7-8 indicate that Random Forest consistently outperformed other models. It improved with both TF/IDF and Count Vectorizer, regardless of the n-gram variations. Random Forest obtained the highest accuracy and F1-measure, with scores of 0.79 and 0.77, respectively. On the other hand, Logistic Regression and SVM showed high recall, suggesting their effectiveness in detecting abusive sentences. However, SVM and Logistic Regression exhibited slightly inferior accuracy and F1-measure compared to Random Forest. The Gaussian Naïve Bayes (NB) model obtained the lowest scores in almost all metrics.

Tables 5-8 illustrates the findings of ML models employing various combinations of n-grams. Models employing unigrams consistently outperformed other n-gram combinations, showing remarkable accuracy and F1-measure. However, the performance of ML models utilizing bigrams exhibited inconsistency. Moreover, ML models using trigrams demonstrated lower accuracy compared to other n-grams but

TABLE 5. Results of ML models with different TF/IDF N-grams on proposed dataset.

Features	Models	Accuracy	F1-Measure	Recall	Precision
Uni	LR	0.95	0.95	**0.98**	0.93
	RF	**0.96**	**0.96**	0.96	**0.95**
	SVM	**0.96**	**0.96**	**0.98**	0.94
	NB	0.92	0.92	0.90	0.94
Bi	LR	0.74	0.71	0.85	0.61
	RF	0.74	0.70	0.86	0.59
	SVM	0.73	0.70	0.83	0.61
	NB	0.74	0.68	0.93	0.54
Tri	LR	0.60	0.71	0.56	0.98
	RF	0.60	0.71	0.56	0.97
	SVM	0.59	0.70	0.56	0.95
	NB	0.60	0.71	0.56	0.98
Uni + Bi + Tri	LR	0.95	0.95	**0.98**	0.93
	RF	**0.96**	**0.96**	0.96	**0.95**
	SVM	**0.96**	**0.96**	**0.98**	0.93
	NB	0.91	0.92	0.89	0.95

TABLE 6. Results of ML models with different count vectorizer N-grams on proposed dataset.

Features	Models	Accuracy	F1-Measure	Recall	Precision
Uni	LR	**0.96**	**0.96**	**0.98**	0.94
	RF	**0.96**	**0.96**	0.97	0.95
	SVM	**0.96**	**0.96**	0.97	0.94
	NB	0.93	0.94	0.92	0.95
Bi	LR	0.75	0.71	0.87	0.60
	RF	0.74	0.70	0.85	0.60
	SVM	0.74	0.70	0.85	0.59
	NB	0.75	0.69	0.93	0.55
Tri	LR	0.60	0.71	0.56	0.97
	RF	0.60	0.71	0.56	0.97
	SVM	0.59	0.70	0.56	0.95
	NB	0.60	0.71	0.56	**0.98**
Uni + Bi + Tri	LR	**0.96**	**0.96**	**0.98**	0.94
	RF	**0.96**	**0.96**	0.97	0.95
	SVM	**0.96**	**0.96**	**0.98**	0.94
	NB	0.93	0.93	0.91	0.96

showed a high level of precision, signifying their effectiveness in reducing false positives. Models combining unigrams, bigrams, and trigrams into a single feature produced results nearly identical to those using only unigrams. These findings illustrate the significance of choosing the proper n-gram depending on the specific objectives necessary for the task of detecting abusive language.

TABLE 7. Results of ML models with different TF/IDF N-grams on established dataset.

Features	Models	Accuracy	F1-Measure	Recall	Precision
Uni	LR	0.75	0.70	0.89	0.57
	RF	**0.79**	**0.77**	0.85	**0.71**
	SVM	0.75	0.69	0.89	0.56
	NB	0.69	0.66	0.72	0.62
Bi	LR	0.61	0.50	0.69	0.39
	RF	0.61	0.50	0.69	0.39
	SVM	0.58	0.61	0.56	0.68
	NB	0.63	0.49	0.77	0.36
Tri	LR	0.50	0.66	0.49	**0.99**
	RF	0.49	0.66	0.49	0.98
	SVM	0.49	0.59	0.49	0.75
	NB	0.50	0.66	0.49	0.99
Uni + Bi + Tri	LR	0.75	0.70	0.89	0.57
	RF	**0.79**	**0.77**	0.84	**0.71**
	SVM	0.75	0.69	**0.90**	0.56
	NB	0.70	0.67	0.74	0.61

TABLE 8. Results of ML models with different Count vectorizer N-grams on established dataset.

Features	Models	Accuracy	F1-Measure	Recall	Precision
Uni	LR	0.77	0.73	**0.85**	0.65
	RF	**0.78**	**0.76**	0.83	0.70
	SVM	0.66	0.69	0.63	0.77
	NB	0.68	0.66	0.69	0.64
Bi	LR	0.61	0.51	0.69	0.40
	RF	0.61	0.49	0.69	0.38
	SVM	0.59	0.57	0.59	0.55
	NB	0.63	0.51	0.74	0.39
Tri	LR	0.50	0.66	0.49	**0.99**
	RF	0.50	0.66	0.49	**0.99**
	SVM	0.51	0.65	0.50	0.93
	NB	0.50	0.66	0.49	0.99
Uni + Bi + Tri	LR	0.76	0.73	**0.84**	0.64
	RF	**0.78**	**0.76**	**0.84**	0.70
	SVM	0.67	0.69	0.64	0.75
	NB	0.69	0.67	0.71	0.63

Moreover, we derived custom word embeddings by training Word2Vec and fastText models on our proposed Urdu abusive data set. Subsequently, these custom word embeddings were employed to train ML models, allowing a more comprehensive assessment of their effectiveness. The experimental results given in Table 9 demonstrated that the ML models achieved competitive performance. Both custom fastText and Word2Vec revealed high levels of accuracy, F1-measure, recall, and precision.

TABLE 9. Results of ML models with Word2Vec and fastText on proposed dataset.

Features	Models	Accuracy	F1-Measure	Recall	Precision
Wor2Vec	LR	**0.91**	**0.91**	**0.91**	**0.91**
	RF	0.89	0.89	0.89	0.90
	SVM	0.90	**0.91**	**0.91**	**0.91**
	NB	0.86	0.87	0.91	0.83
fastText	LR	**0.91**	0.81	**0.91**	0.81
	RF	0.89	0.89	0.88	**0.90**
	SVM	**0.91**	**0.91**	**0.91**	**0.90**
	NB	0.86	0.87	0.90	0.85

The utilization of custom Word2Vec embeddings consistently resulted in superior performance in both Logistic Regression and Support Vector Machine (SVM) models on the proposed dataset. Their accuracy, F1-measure, recall, and precision scores consistently approached to 0.91. However, the RF model exhibited robust performance in accuracy and F1-mesure, achieving a score of 0.89 for both metrics. In contrast, Gaussian Naïve Bayes (NB) demonstrated rather modest performance, attaining an accuracy of 0.86 and an F1-measure of 0.87. The results of different ML models on the proposed dataset, utilizing custom Word2Vec and fastText embeddings, are illustrated in Figure 3.

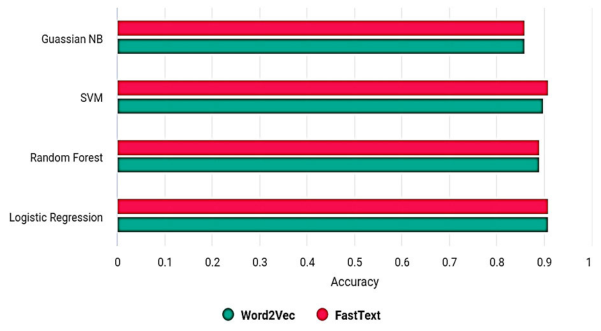


FIGURE 3. Comparative Results ML models with Word2Vec and fastText on proposed Dataset.

In contrast to Logistic Regression, the SVM model consistently exhibited high precision, recall and accuracy of 0.91. However, when utilizing custom fastText embeddings, the F1-measure and precision of Logistic Regression dropped, implying potential trade-offs in prediction. Meanwhile, the Random Forest model sustained good performance in accuracy and F1-measure, with score of 0.89 for both metrics. In contrast, the Gaussian Naïve Bayes classifier demonstrated a well-balanced performance, attaining an accuracy of 0.86 and an F1-measure of 0.87, along with a commendable recall of 0.90. These results emphasize the significant impact of embedding techniques on model performance. On our proposed Urdu abusive dataset, custom Word2Vec-based ML models exhibited superior precision and recall, whereas custom fastText-based models achieved a more balanced performance between precision and other metrics.

Following the training and testing of ML models on our proposed dataset using custom fastText and Word2Vec embeddings, we proceeded to assess our trained ML models on established dataset. The aim was to compare and analyze our results to gain insights into the performance of our ML models. For instance, the custom Word2Vec-based models (Logistic Regression, SVM) demonstrated good performance, achieving an F1-measure of 0.77, indicating their effectiveness in accurate classification. However, the custom fastText-based models (Random Forest, Gaussian NB) had certain limitations, as seen by F1-measures ranging from 0.58 to 0.43. These scores indicate challenges in correctly identifying positive instances within the dataset. These findings emphasize the important role of choosing the suitable ML model and word embedding technique tailored to the specific dataset characteristics. The comprehensive results are presented in Table 10 and visually represented in Figure 4.

TABLE 10. Results of ML models with Word2Vec and fastText on established dataset.

Features	Models	Accuracy	F1-Measure	Recall	Precision
Word2Vec	LR	0.74	**0.77**	**0.89**	0.68
	RF	**0.75**	0.72	0.66	**0.80**
	SVM	0.74	**0.77**	**0.89**	0.68
	NB	0.70	0.64	0.54	0.79
fastText	LR	**0.75**	**0.75**	0.75	0.75
	RF	0.69	0.58	0.44	0.87
	SVM	**0.75**	**0.75**	**0.77**	0.74
	NB	0.63	0.43	0.29	**0.88**

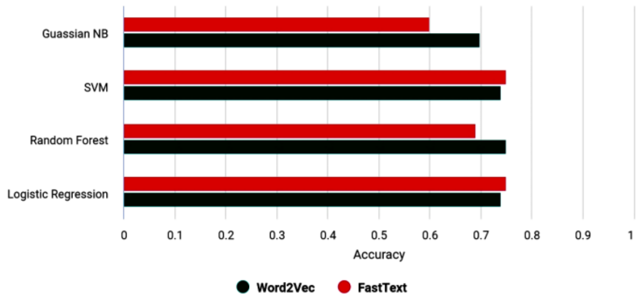


FIGURE 4. Comparative Results ML models with Word2Vec and fastText on Established Dataset.

B. DEEP LEARNING (DL) MODELS

This section presents experiments conducted on DL models utilizing custom-trained Word2Vec and fastText embeddings. The assessment of word embeddings revealed that both Word2Vec and fastText produced comparable results when utilized with deep learning models. Our experiments included various DL models integrated with Word2Vec and fastText, detailed in Table 11 and 12. The performance of each model was assessed using evaluation measures such as precision, recall, accuracy and F1-measure.

GRU and Bi-GRU consistently outperformed other models on the proposed dataset using both Word2Vec and fastText embeddings, as shown in Figure 5. It is worth noting that GRU achieved the highest accuracy and F1-measure, maintaining a balanced combination of recall and precision with both types of embeddings.

Moreover, the LSTM model showed remarkable performance, especially when employing Word2Vec. However, the CNN model, although generally effective, exhibited some variability in outcomes. RNN model showed comparatively lower performance despite its moderate accuracy.

Overall, our research findings emphasize the strong and reliable performance of GRU and LSTM architectures, particularly when combined with Word2Vec embeddings. This highlights the crucial significance of carefully selecting both the embedding technique and model architecture for the identification of abusive language in Urdu. The overall results are demonstrated in Table 11.

TABLE 11. Results of DL models on proposed dataset.

Features	Model	Accuracy	F1-score	Recall	Precision
Word2Vec	CNN	0.86	0.87	0.86	0.87
	RNN	0.81	0.83	0.89	0.78
	Bi-RNN	0.81	0.82	0.82	0.81
	LSTM	0.89	0.89	0.92	0.87
	Bi-LSTM	0.88	0.89	0.88	0.89
	GRU	**0.91**	**0.91**	0.90	**0.92**
	Bi-GRU	0.89	0.90	**0.93**	0.87
fastText	CNN	0.84	0.85	0.87	0.82
	RNN	0.83	0.84	0.84	0.84
	Bi-RNN	0.82	0.82	0.81	0.83
	LSTM	0.87	0.87	0.84	0.90
	Bi-LSTM	0.86	0.87	0.87	0.86
	GRU	**0.92**	**0.92**	**0.93**	**0.92**
	Bi-GRU	0.89	0.89	0.88	0.90

During the evaluation of DL models on well-established dataset, Figure 6 revealed intriguing insights. Specifically, models that integrated Word2Vec and fastText embeddings consistently demonstrated robust performance, particularly in LSTM and GRU models. The findings obtained from the established dataset highlight the adaptability and robustness of these models across diverse data sources. However, it is important to note that overall, our DL models exhibited comparatively lower performance on the established dataset compared to our proposed dataset. This discrepancy indicates a challenge in generalizing the models to diverse data characteristics.

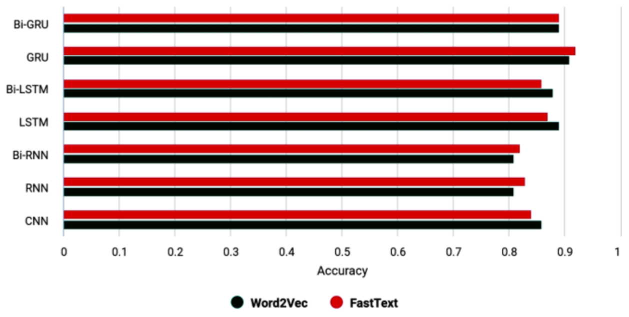


FIGURE 5. Comparative Results of DL Models on Proposed Dataset.

These observations emphasize the need for potential fine-tuning or adaptation to optimize these models for broader applications. In addition, they highlight the critical significance of ensuring model robustness and generalizability in real-world scenarios. The comprehensive outcomes of DL models on established dataset can be found in Table 12.

TABLE 12. Results of Dl models on established dataset.

Features	Models	Accuracy	F1-Measure	Recall	Precision
Word2Vec	CNN	0.76	0.76	0.74	0.75
	RNN	0.78	0.76	9.72	0.82
	Bi-RNN	0.78	**0.79**	**0.82**	0.75
	LSTM	0.79	0.77	0.71	**0.85**
	Bi-LSTM	**0.80**	0.78	0.73	**0.85**
	GRU	**0.80**	0.78	0.72	0.84
	Bi-GRU	0.79	0.78	0.72	0.84
fastText	CNN	0.79	0.79	0.80	0.77
	RNN	0.78	0.79	**0.81**	0.77
	Bi-RNN	0.77	0.76	0.73	0.79
	LSTM	**0.80**	**0.80**	0.78	0.81
	Bi-LSTM	0.79	0.76	0.69	**0.86**
	GRU	**0.80**	0.79	0.73	**0.86**
	Bi-GRU	**0.80**	0.79	0.77	0.81

Our evaluation of DL models aimed at detecting abusive language in Urdu involved investigating the impact of custom-trained Word2Vec and fastText embeddings. The CNN model demonstrated consistent behavior in terms of its performance when trained with fastText and Word2Vec embeddings. However, there were marginal differences observed in the training and validation accuracies between the two embedding techniques.

The CNN model utilizing fastText demonstrated superior performance in terms of training accuracy (0.93) and validation accuracy (0.88) in comparison to the CNN model

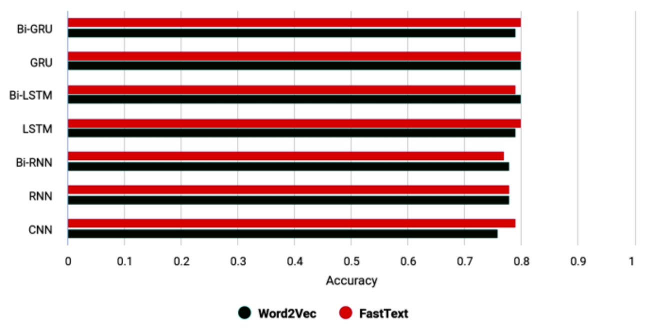


FIGURE 6. Comparative Results of DL Models on Established Dataset.

employing Word2Vec, which exhibited a training accuracy and a validation accuracy of 0.88 and 0.84, respectively.

Moreover, the RNN models utilizing custom Word2Vec demonstrated effective learning on the training data, achieving a peak accuracy of 0.89. However, they encountered difficulty in improving validation accuracy, fluctuating within the range of 0.80 to 0.83. Similarly, RNN models using fastText exhibited a comparable trend, achieving a maximum training accuracy of 0.84 but facing challenges in effectively generalizing to unseen data. The Bi-RNN model exhibited indications of overfitting. The training accuracy of the Bi-RNN model utilizing Word2Vec consistently improved from 0.78 to 0.91 on the training dataset, however the validation accuracy remained unchanged at 0.81. In contrast, the Bi-RNN model employing fastText embeddings achieved a peak validation accuracy of 0.82.

Experiments with LSTM and Bi-LSTM models using Word2Vec and fastText embeddings demonstrated consistent trends. The LSTM with Word2Vec embeddings initially performed well, achieving a peak training accuracy of 0.90. However, the model started to exhibit overfitting when the validation accuracy declines beyond the 7th epoch. On the other hand, the LSTM model with fastText embeddings generalized well to the training data, attaining a training accuracy of 0.88 and a validation accuracy of 0.85. This implies that the model's performance on the training data does not exhibited signs of overfitting. Moreover, the Bi-LSTM model with Word2Vec embeddings achieved its peak training accuracy of 0.92, along with a validation accuracy of 0.89, indicating an overall good fit. Similarly, the Bi-LSTM with fastText also exhibited an increasing trend and generalized effectively, achieving the maximum accuracy of 0.88 for training and 0.86 for validation, respectively.

The experiments conducted with GRU models utilizing both types of embeddings demonstrated consistent performance. The GRU model with Word2Vec embeddings exhibited an increasing trend throughout the training process, achieving the maximum training accuracy of 0.92. Similarly, the GRU model with fastText embeddings outperformed other models by achieving the peak training accuracy of 0.93, along with a validation accuracy of 0.91. The Bi-GRU model with Word2Vec demonstrated robust performance during training, boasting an accuracy score of 0.92 and a validation accuracy of 0.90. However, the Bi-GRU model with fastText embeddings reached a maximum training accuracy of 0.92. However, its validation accuracy of 0.89 is marginally inferior to that of other GRU models. Overall, GRU models demonstrated an increasing trend and generalized well to training data.

C. DEEP LEARNING MODELS + ATTENTION MECHANISM

In this section, we examine the impact of integrating attention mechanisms with DL models on their performance. As stated earlier, we utilized custom embeddings derived from fastText and Word2Vec. The integration of attention mechanisms demonstrated that GRU and Bi-GRU models still outperformed other DL models on the proposed dataset. They achieved a remarkable accuracy of 0.94 with both fastText and Word2Vec, as given in Table 13 and illustrated in Figure 7.

After analyzing the performance of DL models incorporating attention mechanisms on the established dataset, we observed a significant difference in performance compared to our proposed dataset, as illustrated in Table 14 and depicted in Figure 8.

TABLE 13. Result of DL models with Attention ON proposed dataset.

Features	Models	Accuracy	F1-Measure	Recall	Precision
Word2Vec	RNN	0.58	0.56	0.51	0.62
	Bi-RNN	0.94	0.94	0.92	0.97
	LSTM	0.94	0.94	0.92	0.97
	Bi-LSTM	0.95	0.95	0.92	0.98
	GRU	0.94	0.94	0.91	0.98
	Bi-GRU	0.94	0.94	0.92	0.97
fastText	RNN	0.51	0.68	1.0	0.51
	Bi-RNN	0.86	0.87	0.89	0.84
	LSTM	0.93	0.93	0.91	0.94
	Bi-LSTM	0.92	0.92	0.88	0.97
	GRU	0.94	0.94	0.93	0.95
	Bi-GRU	0.94	0.94	0.91	0.97

Models employing both Word2Vec and fastText embeddings demonstrated reduced performance on the established dataset in terms of measures like accuracy, F1-measure, and recall. This indicates that despite the good performance of DL models with attention mechanisms on our proposed dataset, they encountered difficulties in generalizing to the established dataset, most likely due to its unique characteristics and contextual subtleties.

Across various models—RNN, Bi-RNN, LSTM, Bi-LSTM, GRU, and Bi-GRU—there was a consistent trend of achieving lower scores across all evaluation measures when applied to the established dataset. Although models such as Bi-LSTM and Bi-GRU sustained relatively superior performance compared to others, they exhibited a decline in performance

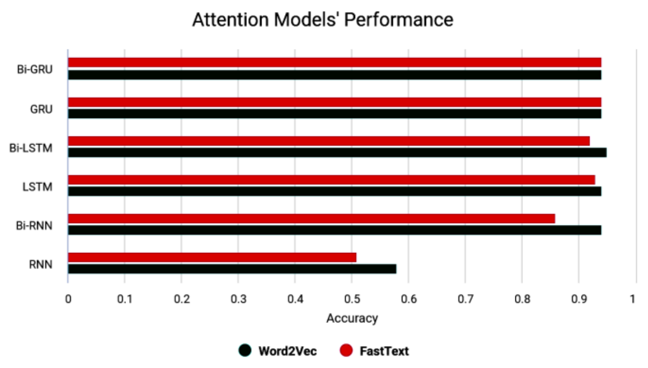


FIGURE 7. Comparative results of attention models on proposed dataset.

compared to their effectiveness on our proposed dataset. These fluctuations highlight the critical importance of testing models across diverse datasets and refining them to adapt to varying data contexts. A comprehensive analysis of the performance trends observed in deep learning (DL) models coupled with attention mechanisms offers invaluable insights.

TABLE 14. Results of DL+ attention models on established dataset.

Features	Model	Accuracy	F1-score	Recall	Precision
Word2Vec	RNN	0.49	0.66	1.0	0.49
	Bi-RNN	0.77	0.75	0.67	0.84
	LSTM	0.78	0.76	0.67	0.86
	Bi-LSTM	0.79	0.76	0.68	0.87
	GRU	0.76	0.74	0.64	0.87
	Bi-GRU	0.78	0.76	0.66	0.88
fastText	RNN	0.49	0.66	1.0	0.49
	Bi-RNN	0.78	0.76	0.70	0.83
	LSTM	0.79	0.76	0.68	0.86
	Bi-LSTM	0.79	0.77	0.70	0.84
	GRU	0.79	0.76	0.68	0.86
	Bi-GRU	0.79	0.77	0.70	0.84

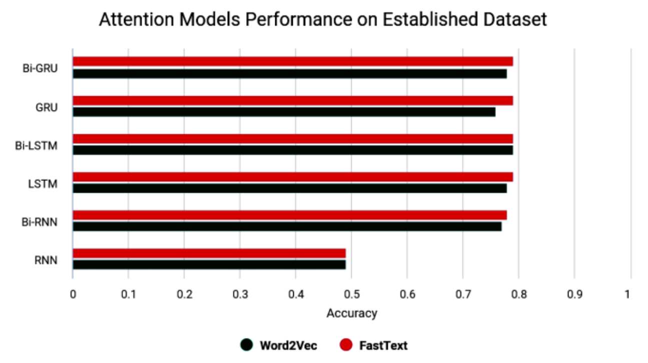


FIGURE 8. Comparative results of attention models on established dataset.

In examining both RNN and Bi-RNN models, a distinct trend emerges where models utilizing Word2Vec embeddings demonstrated superior outcomes compared to those employing fastText. Specifically, the RNN model equipped with an attention mechanism trained on Word2Vec achieved a peak accuracy of 0.74 at epoch 13, while its fastText-based counterpart struggles, maintaining an accuracy plateau around 0.50. However, both models failed to accurately classify the testing data, resulting in inferior performance compared to other models.

Conversely, the Bi-RNN model, including an attention mechanism and Word2Vec, demonstrated robust performance, attaining a training accuracy of 0.94 and a validation accuracy of 0.93. In contrast, its fastText-based counterpart exhibited inferior performance, with the validation accuracy fluctuating before stabilizing at a peak of 0.86.

Shifting our focus to LSTM and Bi-LSTM models highlights a consistent pattern. Models utilizing Word2Vec embeddings demonstrated robust performance. For instance, the LSTM model, equipped with attention and Word2Vec, achieved a remarkable peak training accuracy of 0.95, along with a validation accuracy of 0.94. This performance is remarkably superior when compared to its fastText counterpart, which achieved a validation accuracy of 0.92. Additionally, the Bi-LSTM model, coupled with attention and Word2Vec, outperformed other models on testing data, achieving a notable training accuracy of 0.95 and a validation accuracy of 0.94. However, the Bi-LSTM model with attention and fastText embeddings demonstrated relatively less impressive results. It showed slower convergence and attained lower peak accuracies of 0.92 for training and 0.91 for validation.

In contrast to preceding recurrent models, both GRU and Bi-GRU consistently exhibited an increasing pattern while employing both fastText and Word2Vec embeddings. This consistent behavior indicates their ability to comprehend and adapt to the inherent patterns within the training dataset. It is important to note that both GRU and Bi-GRU achieved a validation accuracy surpassing 0.93 across both fastText and Word2Vec embeddings. These findings indicate that the consistent and remarkable performance of the GRU models strongly emphasizes their suitability and reliability for the challenging task of identifying abusive language in Urdu.

VI. CONCLUSION

This study addresses the challenging task of detecting abusive language in Urdu text—a critical concern considering the extensive use of the internet and social media platforms as well as the harmful effects of abusive content on individuals and communities. Despite the limited linguistic resources available for Urdu, this study investigates the capabilities of ML and DL, and DL with attention mechanisms to assess their effectiveness in detecting abusive language in Urdu.

Initially, our study performed a comparative analysis of four traditional ML models employing word-level n-grams with both TF/IDF and Count Vectorizer. Particularly, models such as RF and SVM utilizing Unigrams, as well as Uni + Bi + Tri-grams, obtained the highest accuracy and

F1-measure of 0.96. Logistic Regression (LR) also exhibited commendable performance, attaining 0.95 accuracy for both Uni-gram and Uni + Bi + Tri-grams. However, Gaussian Naïve Bayes (NB) showed relatively inferior performance among the traditional models. In addition, we extended our analysis by training these four ML models using embeddings derived from Word2Vec and fastText. The results given in Table 9 indicated that LR outperformed other models, with an accuracy and F1-score of 0.91.

In order to improve our results, we deployed DL models employing custom fastText and Word2Vec embeddings. Both types of embeddings produced similar outcomes. The GRU model demonstrated superior performance, obtaining 0.92 accuracy with fastText and 0.91 with Word2Vec. The Bi-GRU model closely followed, obtaining an accuracy of 0.89 on both types of embeddings.

When integrating attention mechanisms in DL models combined with Word2Vec and fastText embeddings, both GRU and Bi-GRU consistently exhibited outstanding performance, reaching a remarkable 0.94 score for both accuracy and F1-measure.

Surprisingly, the Bi-LSTM model incorporating an attention mechanism and employing Word2Vec features, demonstrated slightly better performance than the GRU and Bi-GRU models. Thus, it demonstrated the highest level of effectiveness in detecting abusive language in Urdu.

Moreover, our comprehensive assessment of these models on a well-established dataset (HASOC 2021) confirmed the robustness of DL models. Importantly, both GRU and Bi-GRU consistently outperformed other models, highlighting the significant potential of DL models, particularly the GRU, in identifying abusive language within Urdu text. This research significantly contributes to addressing the gap in abusive language detection within resource-scarce languages like Urdu. It emphasizes the critical importance of leveraging ML and DL techniques to promote safer online environments.

FUTURE DIRECTIONS

In the future, we intend to implement transformer-based models like BERT, which have shown promising results in detecting abusive speech in the English language. Moreover, our primary goal is to collect more extensive annotated datasets of abusive content specifically tailored for Urdu. This major initiative aims to enhance the performance of both machine learning and deep learning algorithms in detection of abusive content in Urdu text. Exploring advanced architectures such as transformer-based models and incorporating ensemble methods to combine predictions could reveal valuable insights for detecting abusive content in Urdu text. Additionally, we aim to expand our work to include other languages, such as Pashto. However, this endeavor necessitates the creation of an annotated dataset for abusive language in Pashto.

REFERENCES

[1] 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.

[2] M. Amjad, N. Ashraf, G. Sidorov, A. Zhila, L. Chanona-Hernandez, and A. Gelbukh, "Automatic abusive language detection in Urdu tweets," *Acta Polytechnica Hungarica*, vol. 19, no. 10, pp. 143–163, 2022.

[3] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive language detection in online user content," in *Proc. 25th Int. Conf. World Wide Web*, Apr. 2016, pp. 145–153.

[4] N. D. Gitari, Z. Zhang, H. Damien, and J. Long, "A lexicon-based approach for hate speech detection," *Int. J. Multimedia Ubiquitous Eng.*, vol. 10, no. 4, pp. 215–230, Apr. 2015.

[5] S. D. Swamy, A. Jamatia, and B. Gambäck, "Studying generalisability across abusive language detection datasets," in *Proc. 23rd Conf. Comput. Natural Lang. Learn. (CoNLL)*, 2019, pp. 940–950.

[6] S. Jan, S. Musa, T. Ali, M. Nauman, S. Anwar, T. A. Tanveer, and B. Shah, "Integrity verification and behavioral classification of a large dataset applications pertaining smart OS via blockchain and generative models," *Expert Syst.*, vol. 38, no. 4, Jun. 2021, Art. no. e12611.

[7] T. Ali, S. Jan, A. Alkhodre, M. Nauman, M. Amin, and M. S. Siddiqui, "DeepMoney: Counterfeit money detection using generative adversarial networks," *PeerJ Comput. Sci.*, vol. 5, p. e216, Sep. 2019.

[8] M. P. Akhter, Z. Jiangbin, I. R. Naqvi, M. AbdelMajeed, and T. Zia, "Abusive language detection from social media comments using conventional machine learning and deep learning approaches," *Multimedia Syst.*, vol. 28, no. 6, pp. 1925–1940, Dec. 2022.

[9] M. O. Raza, Q. Khan, and G. M. Soomro, "Urdu abusive language detection using machine learning," Tech. Rep., 2021.

[10] M. P. Akhter, Z. Jiangbin, I. R. Naqvi, M. Abdelmajeed, and M. T. Sadiq, "Automatic detection of offensive language for Urdu and Roman Urdu," *IEEE Access*, vol. 8, pp. 91213–91226, 2020.

[11] S. Khan and A. Qureshi, "Cyberbullying detection in Urdu language using machine learning," in *Proc. Int. Conf. Emerg. Trends Electr., Control, Telecommun. Eng. (ETECTE)*, Dec. 2022, pp. 1–6.

[12] A. Karthikraja, A. S. Kumar, B. Bharathi, J. Bhuvana, and T. Mirnalinee, "Abusive and threatening language detection in native Urdu script tweets exploring four conventional machine learning techniques and MLP," in *Proc. CEUR Workshop*, 2021, pp. 1–7.

[13] M. Humayoun, "Abusive and threatening language detection in Urdu using supervised machine learning and feature combinations," 2022, *arXiv:2204.03062*.

[14] M. M. Khan, K. Shahzad, and M. K. Malik, "Hate speech detection in Roman Urdu," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 20, no. 1, pp. 1–19, Jan. 2021.

[15] T. Sajid, M. Hassan, M. Ali, and R. Gillani, "Roman Urdu multi-class offensive text detection using hybrid features and SVM," in *Proc. IEEE 23rd Int. Multitopic Conf. (INMIC)*, Nov. 2020, pp. 1–5.

[16] A. Dewani, M. A. Memon, and S. Bhatti, "Cyberbullying detection: Advanced preprocessing techniques & deep learning architecture for Roman Urdu data," *J. Big Data*, vol. 8, no. 1, p. 160, Dec. 2021.

[17] H. H. Saeed, M. H. Ashraf, F. Kamiran, A. Karim, and T. Calders, "Roman Urdu toxic comment classification," *Lang. Resour. Eval.*, vol. 55, no. 4, pp. 971–996, Dec. 2021.

[18] N. Vanetik and E. Mimoun, "Detection of racist language in French tweets," *Information*, vol. 13, no. 7, p. 318, Jun. 2022.

[19] H. Chen, S. McKeever, and S. J. Delany, "A comparison of classical versus deep learning techniques for abusive content detection on social media sites," in *Proc. 10th Int. Conf. Social Inform.* Cham, Switzerland: Springer, 2018, pp. 117–133.

[20] M. Bilal, A. Khan, S. Jan, and S. Musa, "Context-aware deep learning model for detection of Roman Urdu hate speech on social media platform," *IEEE Access*, vol. 10, pp. 121133–121151, 2022.

[21] M. Das, S. Banerjee, and P. Saha, "Abusive and threatening language detection in Urdu using boosting based and BERT based models: A comparative approach," 2021, *arXiv:2111.14830*.

[22] 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.

[23] S. Kalraa, Y. Bansala, and Y. Sharmaa, "Detection of abusive records by analyzing the tweets in Urdu language exploring transformer based models," in *Proc. CEUR Workshop*, 2021, pp. 1–7.

[24] H. Rizwan, M. H. Shakeel, and A. Karim, "Hate-speech and offensive language detection in Roman Urdu," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 2512–2522.

[25] U. Khalid, M. O. Beg, and M. U. Arshad, "RUBERT: A bilingual Roman Urdu BERT using cross lingual transfer learning," 2021, *arXiv:2102.11278*.
[26] B. Haddad, Z. Orabe, A. Al-Abood, and N. Ghneim, "Arabic offensive language detection with attention-based deep neural networks," in *Proc. 4th Workshop Open-Source Arabic Corpora Process. Tools, A Shared Task Offensive Lang. Detection*, 2020, pp. 76–81.
[27] H. Mubarak, K. Darwish, and W. Magdy, "Abusive language detection on Arabic social media," in *Proc. 1st Workshop Abusive Lang.*, 2017, pp. 52–56.
[28] Z. Pitenis, M. Zampieri, and T. Ranasinghe, "Offensive language identification in Greek," 2020, *arXiv:2003.07459*.
[29] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, "Hate speech detection with comment embeddings," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 29–30.
[30] F. Del Vigna, A. Cimino, F. Dell'Orletta, M. Petrocchi, and M. Tesconi, "Hate me, hate me not: Hate speech detection on Facebook," in *Proc. 1st Italian Conf. Cybersecurity (ITASEC)*, 2017, pp. 86–95.
[31] M. O. Ibrohim and I. Budi, "Multi-label hate speech and abusive language detection in Indonesian Twitter," in *Proc. 3rd Workshop Abusive Lang.*, 2019, pp. 46–57.
[32] N. Seemann, Y. S. Lee, J. Höllig, and M. Geierhos, "Generalizability of abusive language detection models on homogeneous German datasets," *Datenbank-Spektrum*, vol. 23, no. 1, pp. 15–25, Mar. 2023.
[33] M. Wich, S. Räther, and G. Groh, "German abusive language dataset with focus on COVID-19," in *Proc. 17th Conf. Natural Lang. Process.*, 2021, pp. 247–252.
[34] H. Karayiğit, Ç. İ. Acı, and A. Akdağlı, "Detecting abusive Instagram comments in Turkish using convolutional neural network and machine learning methods," *Expert Syst. Appl.*, vol. 174, Jul. 2021, Art. no. 114802.
[35] A. Özberk and I. Çiçekli, "Offensive language detection in Turkish tweets with BERT models," in *Proc. 6th Int. Conf. Comput. Sci. Eng. (UBMK)*, Sep. 2021, pp. 517–521.
[36] B. Andrusyak, M. Rimel, and R. Kern, "Detection of abusive speech for mixed sociolects of Russian and Ukrainian languages," in *Proc. RASLAN*, 2018, pp. 77–84.
[37] K. Saitov and L. Derczynski, "Abusive language recognition in Russian," in *Proc. 8th Workshop Balto-Slavic Natural Lang. Process.*, 2021, pp. 20–25.

ATIF KHAN received the M.Sc. degree in computer science from the University of Peshawar, Pakistan, in 2004, and the Ph.D. degree in computer science (text mining) from Universiti Teknologi Malaysia (UTM), Johor Bahru, Malaysia, in 2016. Since 2016, he has been an Assistant Professor with Islamia College Peshawar, Khyber Pakhtunkhwa, Pakistan. His current research interests include data mining, text mining, sentiment analysis and opinion mining, recommender systems, and machine learning. He is a technical committee member of several international conferences. He was a recipient of the Best Student Award and the Pro-Chancellor Award at UTM, during the Ph.D. degree, for his excellent contribution in the field of text mining. He is also serving as an Associate Editor for *ACM Transactions on Asian and Low-Resource Language Information Processing*.

ABRAR AHMED received the B.S. degree in software engineering from Islamia College Peshawar, in 2023, where he is currently pursuing the M.S. degree. His research interests include deep learning and natural language processing.

SALMAN JAN received the master's degree in computer science from the University of Peshawar and the Ph.D. degree from the MIIT, Universiti Kuala Lumpur, in 2019.

His Ph.D. thesis was on android malware analysis and its integration with blockchain for preserving the integrity of the behavioral logs and ultimately to secure the classification results of the behavioral logs through the employed deep learning models, including DCGAN, CNN, and FCNN. He is currently a Senior Lecturer of cyber security and technological conversion with Malaysian Institute of Information Technology, University Kuala Lumpur, Malaysia. He has acquired skills and published in several well-reputed journals in areas of machine learning, deep learning, artificial intelligence, blockchain, the IoT, and security augmented and virtual reality. He is also actively working on international projects in the capacity of consultant and successfully completed four international projects. His publications and citations can be found at: https://scholar.google.com/citations?user=4n2MC74AAAAJ\&hl=en\&oi=ao.

MUHAMMAD BILAL received the M.Sc. degree in computer science from the University of Peshawar, Pakistan, in 2007, and the M.Phil. degree in computer science (database/data mining) from the Institute of Computer Science and IT (Formerly called IBMS), The University of Agriculture, Peshawar, in 2015. He is currently pursuing the Ph.D. degree in the discipline of computer science with Islamia College Peshawar. His current research interests include data mining, NLP, and machine learning. He published a research article with the title "Sentiment Analysis of Roman Urdu Opinion by using Naïve Bayesian, Decision Tree and KNN Classification Techniques," in 2015, which laid base in the area as far as Roman Urdu script is concerned. He has published an article as the coauthor with the title "Machine Learning for Fake News Classification with Optimal Feature Selection" published in *Soft Computing* journal, in 2022. He has published an article in IEEE Access 2022 with the title "Context-Aware Deep Learning Model for Detection of Roman Urdu Hate Speech on Social Media Platform." He has recently published an article on "Roman Urdu Hate Speech Detection Using Transformer-Based Model for Cyber Security Applications" in *Sensors* journal, in 2023.

MEGAT F. ZUHAIRI (Senior Member, IEEE) received the M.S. degree in communication networks and software from the University of Surrey, U.K., in 2002, and the Ph.D. degree in electronics and electrical engineering from the University of Strathclyde, in 2012. He is currently an Associate Professor with Malaysian Institute of Information Technology, Universiti Kuala Lumpur. He is also a Faculty Member of the Informatics and Analytics Department. He is also an Active Researcher and a Certified Cisco Network Academy Instructor. His research interests include computer networking, process mining, blockchain, and system performance and modeling.