

A HYBRID BERT-CNN FRAMEWORK FOR THREAT DETECTION IN LOW-RESOURCE URDU TWEETS

Sajida Akbar¹, Ashraf Ullah², Kaleem Ullah³, Amjad Khan^{*4}, Mohib Ullah Khan⁵,
Saif Ullah Noor⁶

^{1,2, *4,5,6}Department of Computer Science, University of Science and Technology, Bannu, Pakistan

³Department of Computer Science, University of Agriculture, Peshawar

¹sajidamustafa2018@gmail.com, ²ashrafbth@gmail.com, ³kaleembacha800@gmail.com,

⁵mohibkhan10068@gmail.com, ⁶saifipathan194@gmail.com

DOI: <https://doi.org/10.5281/zenodo.21993765>

Keywords

Urdu tweets Threat detection
social media Deep learning
Transformer models, BERT.

Article History

Received: 27 January 2026

Accepted: 12 March 2026

Published: 26 March 2026

Copyright @Author

Corresponding Author: *

Amjad Khan

Abstract

The rapid rise of social media platforms has revolutionized global communication by enabling real-time information exchange and user interaction. However, this growth has also accelerated the spread of abusive, hateful, and threatening content, leading to significant societal and security concerns. Detecting such harmful material in low-resource languages like Urdu is particularly challenging due to its complex morphology, numerous dialects, frequent code-switching, and the lack of large annotated datasets. To address these issues, this study proposes a hybrid deep learning framework for threat detection in Urdu tweets, integrating transformer-based Bidirectional Encoder Representations from Transformers (BERT) embeddings with Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). A custom Urdu Threat Dataset consisting of 6,060 tweets was manually created and pre-processed using tokenization, normalization, stop-word removal, padding, and stemming. Contextual embeddings generated by BERT were combined with CNN and RNN architectures to capture both semantic and sequential linguistic features. Model performance was evaluated using standard metrics: accuracy, precision, recall, and F1-score. Experimental results show that the CNN-based hybrid model achieved the best performance, with 96% accuracy and an F1-score of 0.98, surpassing the RNN model (95% accuracy) and the standalone BERT model (90% accuracy). The proposed approach also outperformed traditional machine learning methods such as Support Vector Machines (SVM) and Naïve Bayes (NB). These findings confirm that transformer-enhanced deep learning architectures are highly effective for detecting threatening content in morphologically rich, low-resource languages like Urdu. The proposed framework offers a scalable and interpretable solution for automated threat detection and social media moderation. Additionally, this research contributes a valuable Urdu Threat Dataset to the research community and highlights the importance of hybrid transformer-based models for multilingual content monitoring and cyber safety applications.

1. Introduction

The ubiquitous nature of social media platforms like Twitter, Facebook, and Instagram has radically reshaped the human interaction process, making it easy to exchange information in real-time, engage in mass communication, and be connected globally. Although these platforms increase the opportunities to freely express and access information in real-time, they are also the sources of great danger, such as hate speech, cyberbullying, misinformation, and threatening content [1, 2].

Cases of this harmful communication have the possibility of ending in real-life consequences- emotional distress, social unrest, and worst, physical violence [3]. The need to detect threatening content is even more acute in a multilingual environment, where linguistic plurality is confronted with socio-political conflicts [4].

To address these threats, researchers have proposed an extensive range of computational methods for hate speech and threatening words recognition. First attempts have utilised traditional machine-learning methods, Support Vector Machines (SVM), Naive Bayes (NB), and Random Forests using manually engineered features: n-grammes, TF-IDF, and sentiment lexicons [5], [6]. Though such methods do reach some level of effectiveness, they are often unable to capture the depth of contexts and semantics of natural language.

With the introduction of deep-learning models and transformer-based models, the text classification area has been significantly boosted. Convolutional Neural Networks [CNN] and Recurrent Neural Networks [RNN] models have been successfully exploited to detect hate-speech and threats in a wide variety of languages [7,8]. More recently, transformer architectures, such as Bidirectional Encoder Representations from Transformers (BERT), XLM-RoBERTa, and other multilingual variants, have been attributed to achieving state-of-the-art performance in the detection of abusive and hateful content [9], [10]. The introduction of hybrid paradigms using contextual embeddings and traditional lexical characteristics or ensemble

classifiers has continued to increase detection performance as a result of a combination of in-depth semantic understanding and explicit lexical cues [11], [12]. In the case of Urdu, recent research has proposed both extensive corpora and introduced transformer-based techniques with promising results despite limited resources [13,14]. As an example, Hussain et al. [13] provide a comprehensive overview of the methodology of UHST-speech detection, with a focus on the benefits of transfer learning, and Rizwan et al. [14] present the RUHSOLD dataset and a CNN-based approach to abusive speech detection. Similarly, the studies of Khan et al. [15] and Adeeba et al. [16] examine deep-learning and ensemble systems on cyberbullying and threat detection in the Urdu language and present a significant improvement in performance compared to the traditional machine-learning systems. Hybrid BERT-based models, such as in the example of Linardatou et al. [17], have also been able to perform well in related emotion-detection paradigms, thus highlighting the potential of integrating deep contextual representations with traditional features.

Despite these efforts, the recognition of threats in the Urdu-language social media is relatively understudied. Urdu has unique challenges due to its complex morphology, non-standard types of dialects, and two-script orthography- Perso-Arabic and Roman Urdu [18,19]. Moreover, the current research focuses on generic hate speech or offensive language, thus pushing the topical problem of explicit threat recognition to the background. The scarcity of large, high-fidelity annotated corpora, as well as the difficulty of dealing with code-switching, limits the development of reliable real-time systems of detection [20, 21].

As a result, there is an acute need to collect special databases and develop models that can help to capture the linguistic and contextual specifics of Urdu, and thus provide more accurate threat identification and scaling of this task.

To address these issues, the current paper presents a deep-learning-based framework for identifying threatening material in Urdu tweets. We train a strong Urdu Threat Dataset and build a hybrid detection framework that integrates BERT

transformer-based embeddings of the context with CNN and RNN components, and in this way detects both contextual semantics and sequential linguistic structures.

Contribution

The main outputs of this research are as follows:

- Urdu Threat Dataset: A massive, balanced Urdu Threat Dataset was created and pre-processed, which depicts realistic threatening content found on Twitter.
- A hybrid, deep-learning model was created, a combination of BERT embeddings, Convolutional Neural Networks, and Recurrent Neural Networks, to give attention to both the contextual and sequential characteristics in language.
- An extensive performance analysis was performed, and the proposed method was compared to conventional machine-learning algorithms and modern transformer models, thus making it show its high effectiveness in threat detection on a low-resource language.

2. Literature Review

The problem of detecting abusive, hateful, and threatening content on social media has gained considerable academic interest in recent years. As the use of low-resource languages such as Urdu has become increasingly popular on social media platforms, researchers have explored both machine learning (ML) and deep learning (DL) methods to address associated security challenges. A review of the existing literature indicates that a combination of traditional ML algorithms, neural networks, and transformer-based models has been applied across various languages and datasets to improve the detection of harmful content.

Focusing on Urdu, Mehmood et al. (2023) proposed an intelligent threat-detection model using a stacking ensemble that combines Linear Regression as a meta-learner with Bernoulli Naive Bayes and Extra Trees classifiers. This hybrid architecture achieved an accuracy of approximately 74%, outperforming existing benchmarks and demonstrating the effectiveness of ensemble learning for detecting threatening language. Balakrishnan et al. (2022) not only

detected abusive text in Urdu and Roman Urdu but also tested various ML and DL models. They found that a single-layer convolutional neural network (CNN) architecture provided superior performance, achieving accuracy rates up to 96%, suggesting that less complex deep networks can outperform more complicated architectures in certain text-classification tasks. Das et al. (2021) studied transformer models for abusive and threatening content detection in Urdu and found that multilingual BERT (m-BERT) achieved the best results, with higher F1-scores compared to other models. Similarly, Amjad et al. (2021) introduced manually annotated Urdu datasets and baseline classifiers, reporting competitive performance using m-BERT while also releasing resources to support future research in low-resource language processing.

Beyond Urdu, abusive-language detection has been investigated in other languages as well. Ibrohim and Budi (2019) focused on Indonesian social media and analyzed abusive language using Twitter datasets. Their experiments revealed that the Naive Bayes classifier with unigram and bigram features achieved strong F1-scores, highlighting the continued relevance of classical ML approaches when combined with effective feature engineering. Akhter et al. (2020) employed TF-IDF feature extraction and enhanced preprocessing techniques for Bengali cyberbullying classification, achieving high accuracy in both multilabel and binary classification tasks. Extending this research, Ahmed et al. (2022) proposed binary and multiclass classification methods for detecting bullying in Bengali text, obtaining impressive results using neural networks and ensemble techniques in multilingual toxicity classification. Explainability and model robustness have also attracted research attention. Islam et al. (2022) proposed a cyberbullying detection framework that incorporated demographic factors such as gender and age together with Explainable AI (XAI) tools like LIME to improve interpretability and data quality. Carta, Giulioni, and Pilo (2019) addressed toxic comment detection using multi-class and multi-label classification of Wikipedia discussions, demonstrating that supervised

learning with customized word embeddings was more effective than traditional bag-of-words models. Georgeakopoulos et al. (2018) compared CNN architectures with bag-of-words techniques and concluded that CNNs achieve superior performance in offensive language classification. Further comparative work by Akash G et al. (2023) on deep learning architectures—including bidirectional LSTM, GRU, CNN-LSTM hybrids, and transformer-based RoBERTa—revealed that fine-tuned transformer models provide superior toxic comment detection performance.

Recent developments in transformer-based approaches have significantly enhanced hate speech detection. Caselli et al. (2021) proposed HateBERT, a BERT-based model retrained on abusive Reddit comments, which achieved better results than standard BERT across multiple datasets, demonstrating the benefits of domain-specific pretraining. Saha et al. (2021) introduced HateMonitor, a multilingual framework combining gradient boosting, BERT, and LASER embeddings to identify hate speech across Indo-European languages, achieving state-of-the-art results in HASOC shared tasks. Leite, Silva, and Bontcheva (2020) emphasized the importance of large monolingual datasets, achieving a macro F1-score of 76% using BERT for binary hate speech classification, which shows that high-quality language-specific datasets remain essential despite advances in multilingual models.

The identification of radical and extremist content has also been explored extensively. Kaur, Sharma, and Goel (2021) used a long short-term memory (LSTM)-based architecture combined with feedforward neural networks to detect radical content in news articles and blogs, achieving high precision in identifying radical, non-radical, and irrelevant content. These findings highlight the capability of recurrent neural frameworks to capture context-dependent extremist patterns across multiple sources.

Taken together, these studies demonstrate a gradual transition from traditional ML algorithms such as Naive Bayes and Logistic Regression toward advanced deep learning and transformer-based approaches. Although classical algorithms remain competitive in some scenarios, recent

findings suggest that hybrid architectures and transformer models such as BERT, HateBERT, and m-BERT consistently achieve higher accuracy, particularly for multilingual and low-resource languages like Urdu. Furthermore, the availability of publicly shared datasets, pretrained models, and Explainable AI techniques contributes to the development of robust and interpretable systems for detecting abusive, hateful, and threatening content on social media platforms.

3. Proposed Methodology

The suggested research is aimed at creating an efficient algorithm for threat content recognition in Urdu tweets through a highly designed pipeline. The first step is the building of a high-quality Urdu Threat Dataset, which is gathered on Twitter and annotated heavily to identify the ones that are threatening and the ones that are not threatening. A combination of precise labelling is used to make the data diverse but less biased and noised. Text pre-processing is essential in the cleaning of raw tweets by removing unnecessary things such as URLs, emoticons, and special characters, to have clean input.

This is followed by tokenising, padding, and normalization of the text in order to maintain the same format. Noise is alleviated by removing stop words, and words are stemmed and lemmatised to their base forms to increase standardisation. In the case of a BERT-based model, special tokens like [CLS] and [SEP] are used to represent sentence-level context and segmentation boundaries. Following the pre-processing step, BERT embeddings are used.

The original tweet written in Urdu is first embedded using BERT. Let $X = (x_1, x_2, \dots, x_n)$ be the input sequence. BERT

Creates contextual representations as:

$$H = \text{BERT}(X) = [h_1, h_2, \dots, h_n]$$

These embeddings represent semantic and contextual information of words in the sentence.

To produce dense vector representations of the Urdu text, representing deep contextual and semantic relationships by processing tokenized words when run through a pre-trained BERT model, which is later fine-tuned on the dataset

collected to further improve the performance of the particular task of threat detection.

The resultant embedded capabilities are subsequently input into two deep-learning networks complementary to each other: a Convolutional Neural Network (CNN) to detect local patterns and a Recurrent Neural Network (RNN) to find sequential relationships.

To learn local textual features, we use a Convolutional Neural Network (CNN) on the embeddings:

$$c_i = f(W \cdot h_{i:i+k-1} + b)$$

This operation extracts important n-gram features, where W is the convolution filter, b is the bias term, and f is a non-linear activation function.

A max-pooling operation is then applied to retain the most significant features:

$$\hat{c} = \max(c_1, c_2, \dots, c_n)$$

This reduces dimensionality while preserving the most relevant information.

To capture sequential dependencies in the text, a Recurrent Neural Network (RNN) is employed:

$$h_t = \tanh(W_h h_{t-1} + W_x x_t + b)$$

In this formulation, h_t represents the hidden state at time step t , h_{t-1} is the previous hidden state, and x_t is the current input. The weight matrices and bias are learned during training, enabling the model to capture contextual relationships across the sequence.

In order to achieve sound learning, the dataset is divided into training, validation, and test sets, whereas binary cross-entropy loss and Adam optimizer are utilised and optimised with the help of well-chosen hyperparameters to maximise the precision and stability of the model.

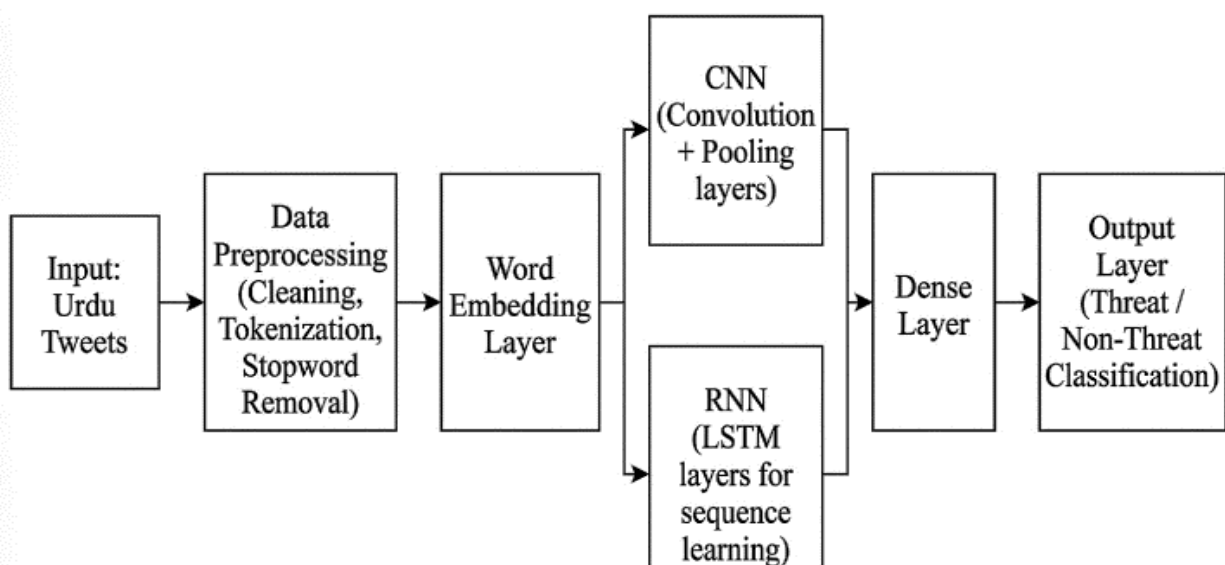
$$L = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

The final classification output is computed using the Softmax function:

$$\hat{y} = \text{softmax}(W_o h + b_o)$$

During the final stage, the trained models are tested on a benchmark test set to measure their ability to generalize to data outside of the training data using the influential measures of accuracy, precision, recall, and the F1 score. In addition, the interpretation of the model choice is performed using attention visualization techniques, thereby offering transparency.

Such a systematic approach to the problem with NLP pre-processing, BERT-based feature embedding, and hybrid deep-learning structures can be used to provide a reliable, explainable, and efficient solution to detect threatening content in Urdu tweets.



3.1 Urdu Threat Dataset

This study uses a curated dataset of Urdu tweets labelled as threats, non-threats, hate speech, or inappropriate language. The diverse categorization supports accurate model training, enhancing threat recognition and strengthening social media moderation in Urdu contexts.

This research relies on a carefully curated Urdu Threat Dataset consisting of 6,060 social media tweets gathered from publicly available Twitter data. Each tweet has been manually labelled by native Urdu speakers to ensure accuracy and reliability. The dataset is organized into three structured columns: (1) a serial number that provides a unique identifier for each tweet, ensuring proper indexing and easy tracking throughout the experimental process; (2) the tweet text, which contains the original Urdu content in its natural form, including informal expressions, code-mixed Urdu-English words, and region-specific vocabulary; and (3) a category label indicating the nature of the tweet, with classes such as Threat, Non-Threat, Hate Speech, and Inappropriate.

Language. This structured format supports both binary (threat vs. non-threat) and multi-class classification tasks, making it suitable for training and evaluating a variety of machine learning and deep learning models. The diversity of language styles within the dataset—including slang, abbreviations, and dialectal variations—was deliberately preserved to reflect real-world social media usage and enhance the model's ability to generalize to unseen data.

A text pre-processing pipeline was used to clean up the data before it was input into the models; the uncooked tweets were cooked to be analysed. Data filtering was done in the first stage, where expendable items like URLs, user mentions, hashtags, smiley faces, emoticons, and special characters were ruthlessly eliminated. These factors, though ubiquitous in social media, tend to add noise that can distract the models of essential textual contents. The cleaning was followed by the subjecting of the text to.

A tokenisation process is based on the subdivision of the tweets into smaller units or tokens, which are the words or subwords, in most cases, enabling

them to be described in numerical terms. In order to have even-length inputs to deep learning models, padding was used by inserting [PAD] tokens to short sequences, and overly long tweets were cut to a predefined length. This makes all the input sequences have equal dimensions, enhancing computational stability and efficiency. Stop words like is, the and in were further eliminated to improve the quality of the text. These words, common in everyday language but with little semantic content, may augment model complexity without enhancing predictive accuracy. Further, lemmatisation and stemming were done in order to minimise words to their foundations or roots. Simplification of words to their simplest root is referred to as stemming, whereas removing words with a common root is called lemmatisation.

linguistic rules to identify the words by mapping them to canonical equivalents in the dictionary. These actions reduce the dimensions of the feature space, enhance textual consistency, and allow the model to emphasise meaningful content as opposed to differences in words on the surface. In the case of BERT-based modelling, special tokens are used, including [CLS] (classification token).

To indicate the beginning of a sentence and denote the limits between text, [SEP] (separator token) was placed between segments. These tokens assist the model in obtaining sentence-level correlations and give a more subtle understanding of context. After pre-processing, embedding of features was done with a fine-tuned Bidirectional Encoder.

Transformers (BERT) model representations, specifically modified to perform tasks in the Urdu language. BERT learns thick, contextualized, and vector representations of every token based on the whole sentence in both directions, both left and right dependencies. This gives the embeddings the ability to portray not only the meaning of individual words but also their connections with the larger sentence, which is essential in identifying nuanced threats. The BERT model was also trained on the hired dataset to maximize its performance on the collected dataset-specific task threat detection in Urdu, and the embeddings

capture language and culture-specific nuances of the data.

Two complementary deep learning architectures to exploit these BERT embeddings were then used as inputs to the various features of the text. The former is a model that incorporates BERT embeddings and a Convolutional Neural Network (CNN). CNNs are especially efficient in local textual feature representation and important n-gram trends, which makes them very useful in providing them with the necessary level of accuracy in identifying particular keywords or brief phrases which indicate danger. The second model combines.

Embeddings of BERT utilize a Recurrent Neural Network (RNN), which is created to capture sequential information. relations and sequence of words in time. It is a capability to process context in long sequences.

RNN to identify finer linguistic information and patterns of threatening speech that can cut across words or phrases. phrases. In order to have a strong training and effective evaluation, the dataset was split into training, testing, and validation sets.

The training set generated the model parameters, and the validation set was utilized to monitor the performance. to avoid overfitting during training, to direct hyperparameter tuning. The test set, which is stored very apart based on the training and validation data provided, was an objective reference point for assessing the generalization of the model's ability. The binary cross-entropy loss, which is a loss function quite appropriate to this case, was used to optimize its models during training in binary and multi-class text classification problems, and the Adam optimizer, which is characterized by good convergence. properties. Hyperparameters, including a learning rate, batch size, and dropout rates, were tested to ensure. optimal accuracy-computational efficiency.

Models trained were also assessed strongly on the held-out test set, which is a balanced set representation of threatening and non-threatening tweets, so as to have equal evaluation. Performance was measured

Graphical presentation of the results through standard classification measures such as accuracy,

precision, recall, and F1-score to give an in-depth knowledge concerning the strengths and weaknesses of models.

Accuracy indicates the general accuracy of predictions, precision is the ratio of threats correctly identified to all the detected threats, recall measures the capability to capture the real threats, and the F1-score weighs between the precision and recall to report on one performance indicator of the model.

This study aims to enhance transparency and interpretability, which are usually weaknesses of deep learning models, by focusing on visualization technologies. These methods emphasize the words or phrases that are particularly used by the model, which is concerned with the timing of making predictions, providing insightful information on the decision-making process. By visualizing attention weights, researchers and practitioners can confirm that the model is based on context. significant characteristics over random noise, hence enhancing confidence in predictions of the system.

In short, this data collection, pre-processing, and contextual BERT are of high quality in this methodology. Embeddings and hybrid deep learning models (CNN and RNN) to create an explainable and reliable threat.

Urdu Twitter detection system. By making sure the system is not merely integrated, the integration of interpretability tools is made not only accurate but transparent enough, which preconditions its use in social media moderation in the real world. content filtering.

3.1.1 Analysis and Interpretation of Data

The data that were collected were analysed and interpreted to answer the research questions and assess how the proposed classification framework can be used to detect dangerous Urdu. tweets. The main aim was to evaluate threatening material and non-threatening material in the most accurate way possible. based on a mix of a natural language processing (NLP) approach and a deep learning model. To achieve this, several methods were critically analysed and contrasted so as to ensure the choice of the most efficient. and an accurate

mechanism for detecting threats in Urdu social media text.

Each of the experiments was conducted with the Python programming language, which was selected due to its versatility and ease of use. use and a rich library ecosystem needed to run machine learning and deep learning. Python's extensive. It was specifically facilitated by data pre-processing, model development, and performance evaluation suitable for this research. Jupiter Notebook was used in an open-source environment to create and execute. The code, which offers an interactive space in which it is possible to analyse, visualize, and test a model. This setup enabled the smooth combination of code execution and result interpretation, which made the process of code execution and result interpretation smooth and efficient.

workflow.

The training, validation, and testing were carried out on the dataset that comprised 6,060 Urdu tweets retrieved via Twitter to test the models. All the tweets were characterised as either threatening (value 1 was assigned) or not threatening. (assigned the value '0'). This dichotomous structure allowed the models to acquire different linguistic trends.

Concomitant with threatening conduct in screening out innocent material. In order to assess model performance and avoid overfitting, the data were separated into training, validation, and test subsets. The training set. The models were fitted with a method used to monitor and tune the hyperparameters, and the test set gave a fair evaluation of the accuracy of the last model on unseen data.

In the analysis part, a test of the models on the held-out test set was carried out to gauge their

capacity to generalize to new tweets. The evaluation metrics that were used to assess performance included accuracy.

precision, recall, and F1-score, which could be combined to give a complete picture of how effective. The system recognizes r dangerous material. Precision is the degree of overall correctness, and accuracy is indicated. The proportion of successfully recognized threats over all the threats predicted, recall, is the proportion of the actual ones. threats are properly identified, and the F1-score is used to balance the precision and recall to compensate for any imbalance between the classes.

Such a systematic treatment made it possible to study the classification opportunities of the model in a comprehensive manner, so that the results would be reliable and interpretable. Jupyter takes advantage of the advanced libraries of Python.

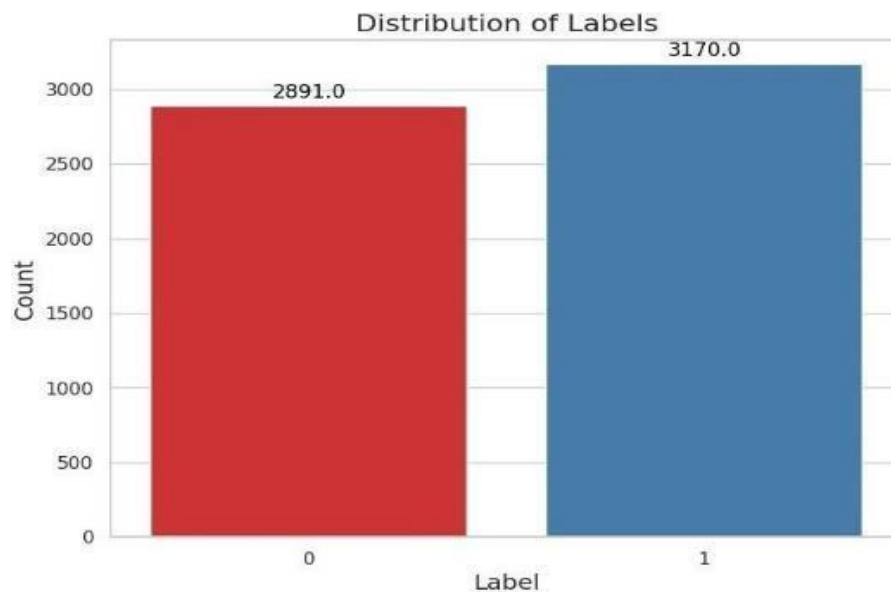
With the interactive space of the notebook and a properly prepared dataset, the analysis could show, in fact, in a successful way. The perspective of deep learning methods in identifying threats in Urdu social media text is provided. Set the stage for automated threat detection systems improvement.

Data Visualization

The data visualization process entails the representation of data in a graphical form or chart in terms of visual elements like graphs, charts, and so on.

maps. Visual presentation is relevant in the communication of information that has repetitive attributes.

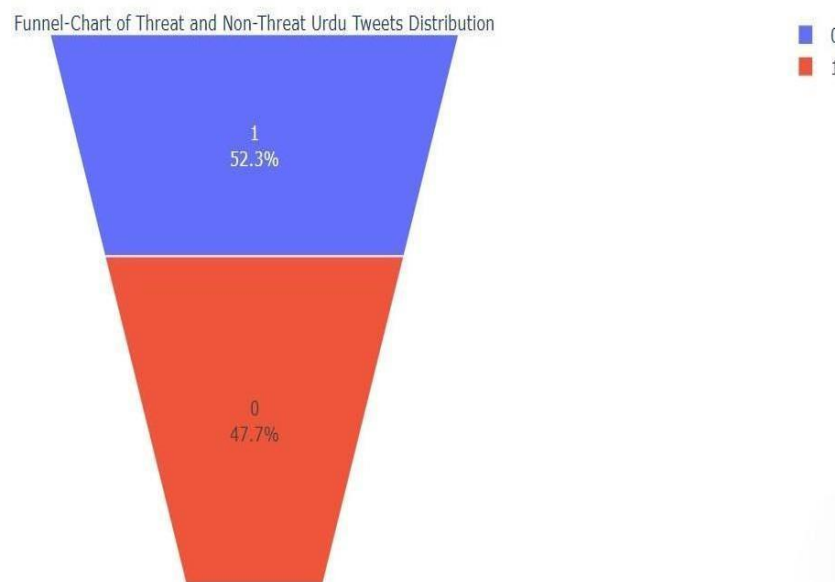
trends, and give insights that can be understood.



The Figure Funnel Chart of Urdu Tweets gives a brief description of the sources of our dataset. The statistics indicate that there were 6060 tweets altogether. are obtained from the social networking site, Twitter. Among all the tweets analysed, The number of those deemed to be non-

threatening was 2890, and those deemed to be ominous were 3170.

The essential aim of the current study is to identify and classify with the help of BERT and deep learning algorithms. train Urdu threat and non-threat tweets on a particular dataset



The data mining of the dataset indicates that non-threat tweets comprise approximately 52% of the total dataset, and about 47.7 % of the data comprises threat tweets. It is important to know

how this data is distributed to appreciate the equilibrium between the two classes. It plays a rather significant role in building the right Urdu threat detector models.

Pre-processing was done using the NLTK resources to make the text refined so that the quality of the dataset could be refined was clean and standardized model-training. The tweets were tokenized, turned down to lowercase, and stripped of stop words and numeracy tokens, which leads to clean text that is uniform and free of noise, which is suitable to be used with classification and other purposes, natural language processing (NLP) problems. To identify threats, the article used a BERT-based CNN and

RNN-based models: The RNN-based models also made use of contextual embeddings as a method to capture both local and sequence features in the Urdu tweets. This mixed strategy provided good classification and good generalization on unseen data.

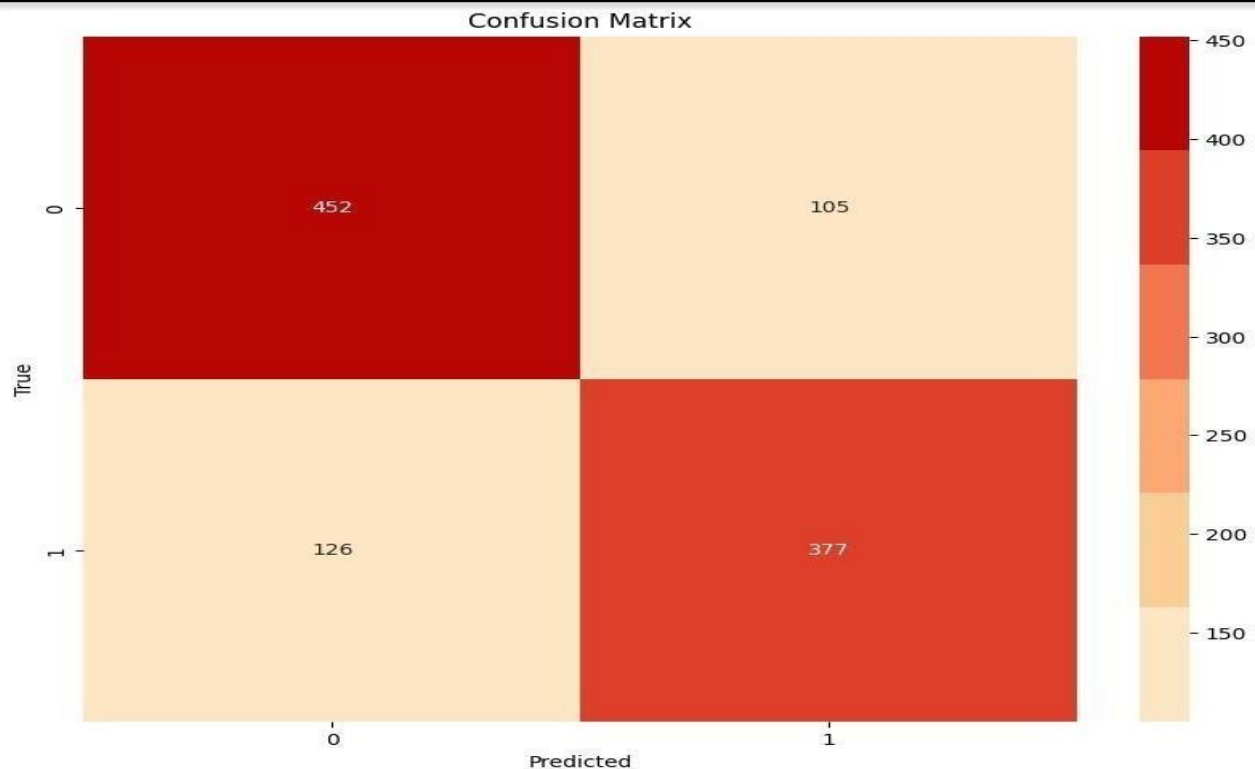
data. The BERT-based model was well-tuned with the use of the Bert-base-uncased architecture with a maximum sequence length of 128 tokens. Bert Config, Ber , Fast, and TF Auto Model were among the key components, which took care of text tokenization, sequence standardization, and loading pretrained BERT weights, so that the model was ready to perform various NLP tasks. The 10-model training was conducted with 133 batches/epoch training in TensorFlow and Keras, where the training accuracy gradually increased, the loss was reduced, and overfitting was reduced. The training of this process was about 90% accuracy, which is a strong learning and effective generalization.

A Convolutional Neural Network (CNN) was built to complement the use of BERT and increase classification performance. The Urdu tweets that were pre-processed were translated into a numerical sequence with a tokenizer. Padding was used to ensure an equal number of sequences in the samples, which allowed the CNN to learn well. from the data. The CNN structure was characterized by embedding, convolution, pooling, and dense layers and was embedded. trained for 10 epochs, the accuracy increased considerably, whereas the loss decreased, proving. effective learning and credible threat detection.

In order to assess the CNN performance, a confusion matrix has been created by means of scikit-learn, Matplotlib, and Seaborn. This heatmap visualization compared the predictive and actual classes, which has given a clear representation. of the accuracy of the classification and stating the strengths and weaknesses of the model to identify the threats within. Urdu tweets.

3.2 Visualizing Model Performance with a Confusion Matrix for Urdu Threat Detection

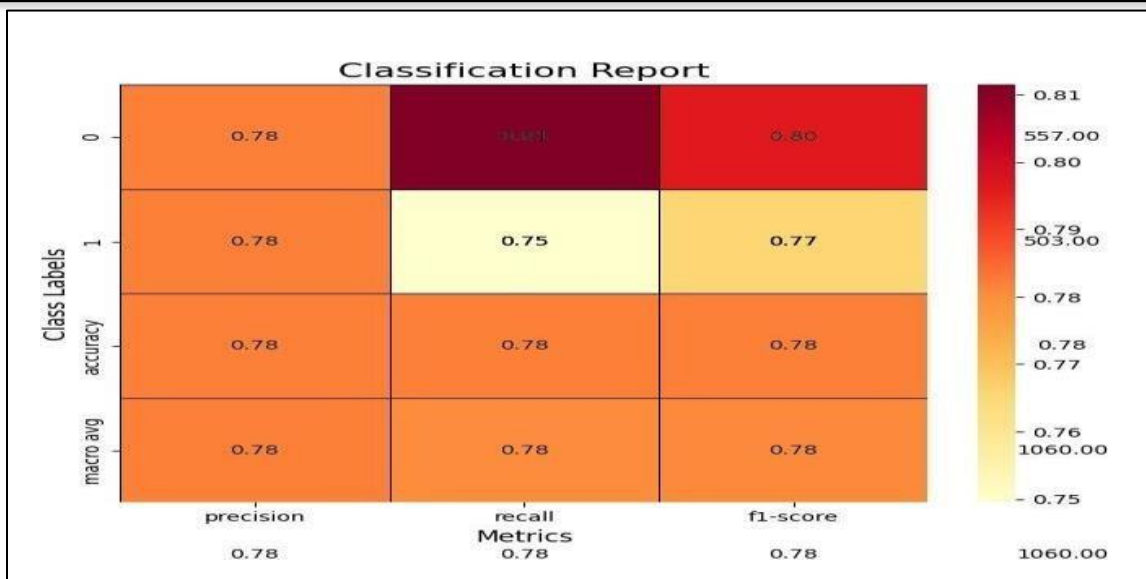
The CNN was assessed with the help of a confusion matrix created with scikit-learn, Matplotlib, and Seaborn. Comparing predicted and actual classes, the visualization in a heatmap showed the accuracy of classification, and the strengths and weaknesses in threat detection were outlined in Urdu.



Classification Report of CNN

A deep learning model is prepared to handle a scenario related to classification; an illustrated report is made. endangered the Urdu language. The name YIOrRd is used as a heatmap to demonstrate class. charting information of the custom colour scheme. Each row in the heatmap represents a certain metric. value, and the importance of each point is described by the

colour. Each cell has an annotation that contains. The metric measure of the corresponding value and the abscissa and ordinate labels are utilized to signify groups. These visual aids a summary of the model performance with reference to categorizing hazards in Urdu text. The report on the classification of the Convolutional Neural Network is shown in the figure.



RNN-Based Text Classification for Enhanced Urdu Threat Detection

The essence of the given stage is the development of a Recurrent Neural Network (RNN) model to classify Urdu text, which in this context will be considered in terms of potential threat-relevant content. This model generation is a vital part of a larger research project that should determine and discuss warning signs in the Urdu language.

The first one is feature extraction, where the raw text data is processed and converted into a numerical representation that is processed effectively by the neural network. This change enables the model to define linguistic patterns in a machine-readable format, which is central to the deep learning usages (such as RNNs).

An LSTM (Long Short-Term Memory) layer is incorporated in the model in order to analyse sequential text data effectively. The LSTM architecture is used to allow the network to learn long-range dependencies and word context in a sentence. The LSTM layer acquires these sequential patterns, and this understanding of the development of meaning over word sequences is crucial in identifying subtle threats in text.

After the LSTM layer, there are dense (fully connected) layers that are introduced to increase pattern recognition. These layers help the network to determine the relationships between the encoded text features, which are complex, to enhance the overall classification performance.

The last layer has a SoftMax activation function to produce probabilities of the classes so that the model can be able to differentiate between the threat and non-threat classes.

The model is put together with the Adam optimizer, which facilitates efficient and adaptive learning, and the categorical Cross-Entropy loss function, which is normally applied in multiclass and binary classification issues. The options are beneficial in training models to reduce loss and enhance accuracy.

In sum, the RNN model processes the Urdu text sequentially, learns the dependencies on the context, and correctly predicts whether the language contains threats or not. The method builds on automated threat detection within the Urdu setting, which helps in developing more successful and smart text classification systems.

4.1 Model Training with RNN

Ten (10) epochs were used to train the model to determine the learning ability of the model in generalization. In the first epoch, the model had a training accuracy value of 64.44 and a validation accuracy of 76.34 with a training loss and validation loss of 0.6153 and 0.4906, respectively.

These preliminary outcomes suggest that the

model began to learn significant patterns, though they still needed to be optimised.

During the second epoch, there were considerable improvements. The accuracy of the training increased to 87.06, and the accuracy of the validation was 77.59. The training loss was minimized to 0.3060, and the validation loss was slightly low at 0.4851, indicating improved convergence of the model.

In the last epoch, the model achieved its best training accuracy of 95.21%, indicating that it has learned the dataset. But, the accuracy of the validation decreased a notch to 74.37, which can be indicative of slight overfitting, whereby the model was more accurate on training data as compared to the unseen validation data. The loss at this stage was 0.1370, and the loss at the validation was 0.6540

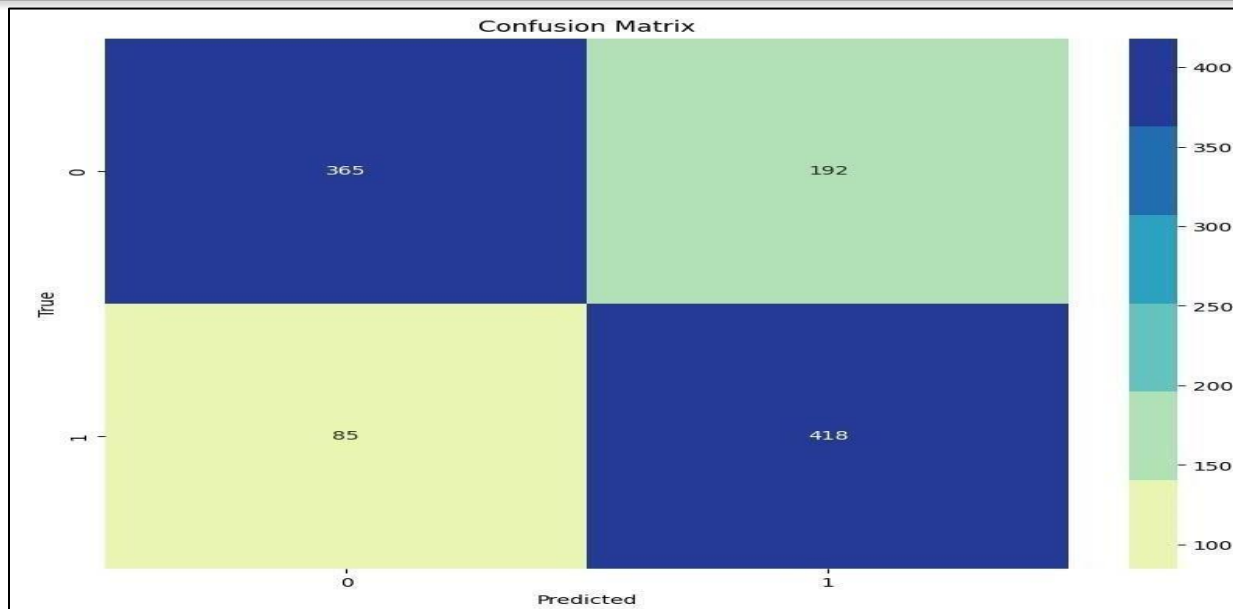
On the whole, training and validation metrics comparison can give helpful information concerning the work of the model. The fact that loss decreases gradually, and accuracy increases with the epochs, proves the fact that the model learns the information. In the meantime, the minor difference between training and validation outcomes presents the aspects in which the generalisation may be improved. The findings can be used to determine the effectiveness of the model in predicting new or unobserved data to guide further optimisation or further tuning.

3.3 Performance Metrics Evaluation for Urdu Threat Detection

To test the text classification ability of the RNN model, the confusion matrix was created when the model was trained to classify text. This matrix gives the graphical representation of the predictions of the model against the actual labels to enable seeing how the model is accurate in distinguishing between the defined classes.

It starts with the comparison of the actual class labels of the dataset and the predicted labels generated by the RNN. The findings are then displayed in a heatmap as a blue-to-green scale to show the spread of right and wrong predictions. This shows the relations between the true positives, false positives, true negatives, and false negatives in a better way, and through this, one can gain a greater understanding of the classification tendencies of the model.

According to the results of the confusion matrix provided in the figure below, it is possible to note that the developed RNN-based model is quite effective as it correctly recognized the relevant classes and correctly classified the majority of the instances. This illustrates the great ability of the model to discriminate various text types, and it was confirmed to be useful in the case of the Urdu threat detector.

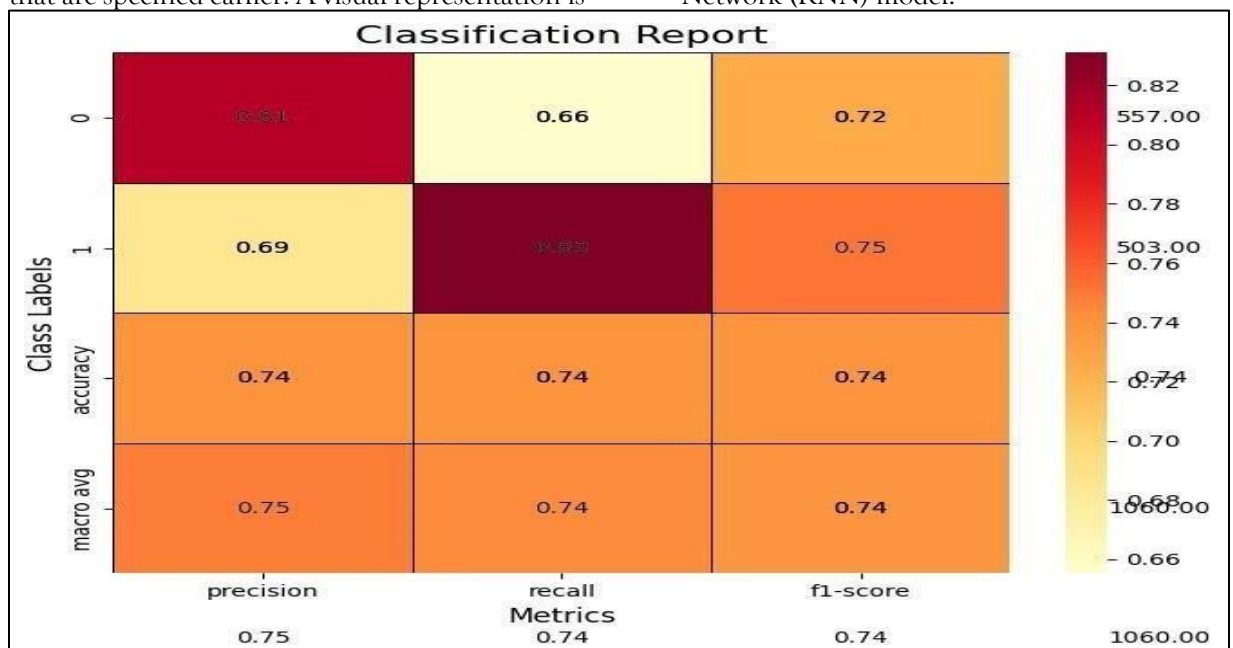


Classification Report of RNN

The classification report contains many metrics, such as precision, recall, and F1-score, and are intended to measure the performance of every class within a deep learning model. The technique uses a standardized colour scheme named Lord to generate a heat map that effectively displays the metrics used in the classification of various classes that are specified earlier. A visual representation is

more useful in making it easier. compares the performance of the model on different classes, and this may offer some insightful information. of the natural strengths and weaknesses of the model.

This improves the readability of the information since the specified cells display the rounded measure. values. The figure displays the classification report of the Recurrent Neural Network (RNN) model.

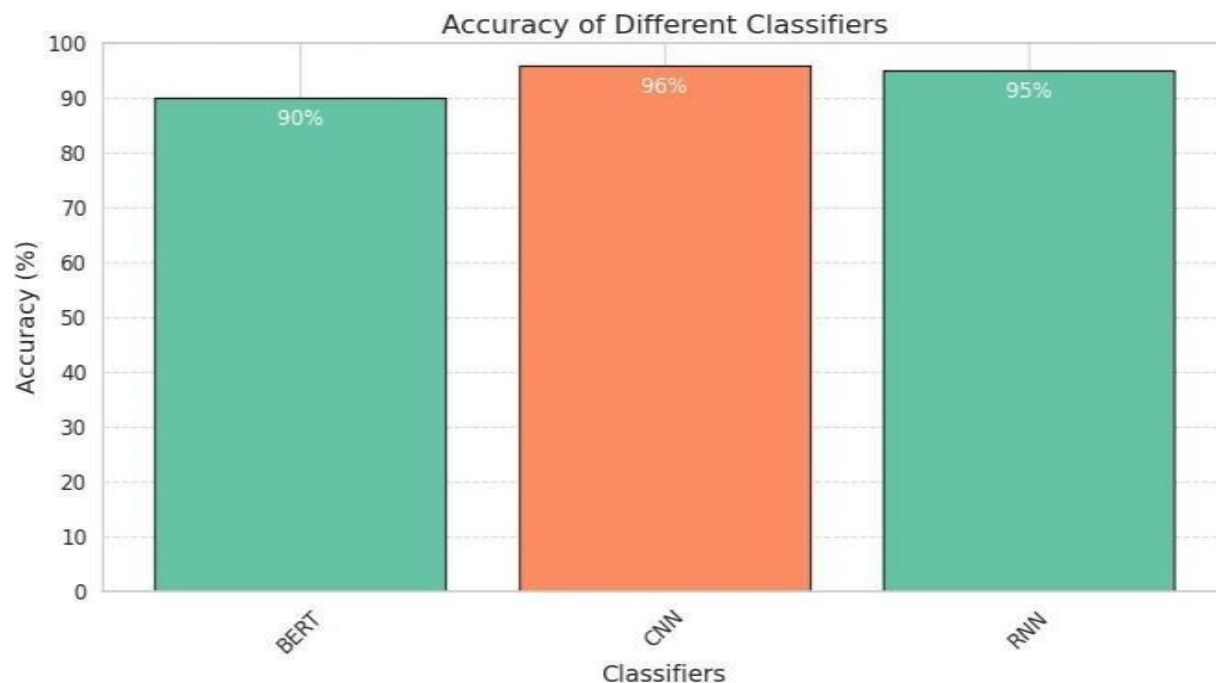


Performance Comparison: BERT vs. CNN vs. RNN Accuracy

The accuracy of BERT, CNN, and RNN models was compared through the practical and usable sample that was given in order to detect the Urdu threats. provides a short analysis of the accuracy obtained.

Regarding the classifiers, which include BERT, CNN, and RNN. The accuracy scores are expressed as follows: percentages serve the purpose of a measure of the performance of each classifier in the specified task. Under this current scenario, the BERT model achieved an accuracy of 90.0 in classifying, the CNN model achieved an accuracy of 96% in classifying, the RNN model had a classification accuracy of 95%. In this instance, a bar chart is applied. to

serve to render the data more graphical so that one may easily and immediately pass judgment on the performance of the two models. The chart is designed keeping several aspects in mind; these refer to the labelling, aesthetic characteristics of the chart, and the grid lines to augment the chart. aesthetic worth and, hence, simplify it. The decisions related to this are easy to make. choice of a model due to the short and accurate description of the accuracy score of the 'BERT', The CNN and the RNN models and the codes are transparent to future studies. The figure shows the comparison of the accuracy of the suggested models adopted by this study. While table shows the model accuracy in tabular form.



Proposed Model Accuracies

Model	Accuracy (%)	Precision	Recall	F1 Score
BERT	90	0.89	0.88	0.90
CNN	96	0.97	0.96	0.98
RNN	95	0.95	0.93	0.94

5. Conclusion

The experimental evaluation was conducted using the Urdu Threats Dataset, which contained a balanced distribution of threatening and non-

threatening text samples. The models were trained and tested on preprocessed textual data represented through BERT embeddings. The performance of each model was assessed using

standard metrics, including accuracy, precision, recall, and F1-score, to ensure a fair and comprehensive comparison.

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

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

$$\text{Recall} = \frac{TP}{TP + FN}$$

$F1 = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$
The Convolutional Neural Network (CNN) demonstrated the best overall performance, achieving an accuracy of 96% and an F1-score of 0.98. This indicates that CNN effectively captured spatial word dependencies through convolutional feature extraction. The Recurrent Neural Network (RNN) model, which incorporated Long Short-Term Memory (LSTM) layers, achieved 91% accuracy, showing strong ability in identifying sequential dependencies within Urdu text. Meanwhile, the BERT-only model attained 90% accuracy, reflecting that while transformer-based contextual embeddings are powerful, additional learning layers such as CNN or RNN further enhance performance for this specific classification task.

When compared with traditional machine learning baselines such as Support Vector Machines (90.7%) and Naïve Bayes (86.3%), the proposed deep learning models outperformed conventional techniques across all evaluation measures. This improvement can be attributed to the contextual representations captured by BERT embeddings and the hierarchical learning capabilities of CNN and RNN architectures.

The results also indicate that CNN performs slightly better than RNN in this task due to its ability to extract discriminative n-gram features and reduce overfitting through pooling operations. However, RNNs demonstrated superior performance in understanding longer contextual dependencies, suggesting that a hybrid CNN-RNN model could potentially yield even stronger results.

In addition, qualitative analysis revealed that both models were highly effective in detecting implicit and context-dependent threats, a task where traditional classifiers typically fail. The

experimental outcomes confirm that transformer-based embeddings, when combined with deep neural architectures, significantly enhance threat detection performance in morphologically complex and low-resource languages such as Urdu. Overall, the proposed framework provides a scalable foundation for multilingual social media moderation systems. Future research may extend this approach by integrating attention mechanisms or fine-tuning multilingual BERT models to improve real-time performance and adaptability across diverse linguistic domains.

5. REFERENCES

- A. Ullah, K. U. Khan, A. Khan, S. T. Bakhsh, A. U. Rahman, S. Akbar *et al.*, "Threatening language detection from Urdu data with a deep sequential model," *PLOS ONE*, vol. 19, no. 6, p. e0290915, 2024.
- M. Amjad, A. Zeila, G. Sidorova, A. Labnehs, S. Butta, H. I. Amjad, and A. Gelbukh, "Overview of abusive and threatening language detection in Urdu at FIRE 2021," *arXiv preprint*, 2022.
- T. Caselli, S. Tulkens, and W. Daelemans, "Retraining BERT for abusive language detection: HateBERT," *Proc. Workshop on Online Abuse and Harms, ACL Anthology*, 2021.
- S. Butta *et al.*, "Emotions and threat detection in Urdu at FIRE 2022 (EmoThreat): Task overview and resources," *CEUR Workshop Proceedings*, CEUR-WS, 2023.
- W. Ashiq, S. Kanwal, A. Rafique *et al.*, "Roman Urdu hate speech detection using hybrid machine learning models and hyperparameter optimization," *Scientific Reports*, vol. 14, p. 28590, 2024.
- A. Rashid, "Urdu toxicity detection: A multi-stage and multi-label approach," *Algorithms*, MDPI, 2025.
- E. Shoukat, R. Irfan, I. Basharat, M. A. Tahir, and S. Shaukat, "Attention-based bidirectional GRU hybrid model for inappropriate content detection in Urdu," *arXiv preprint*, 2025.

- M. Humayoun, "Abusive and threatening language detection in Urdu using supervised machine learning and feature combinations," *arXiv preprint*, 2022.
- M. S. Khan et al., "Detection of violence incitement expressions in Urdu tweets," *Knowledge-Based Systems*, ScienceDirect, 2024.
- F. K. Saddozai, S. K. Badri, and Daniyal et al., "Multimodal hate speech detection: A deep learning framework for multilingual text and images," *PeerJ Computer Science*, 2024.
- S. Butt, M. Amjad, F. Balouchzahi, N. Ashraf, R. Sharma, G. Sidorov, and A. Gelbukh, "Overview of EmoThreat: Emotions and threat detection in Urdu at FIRE 2022," *CEUR Workshop Proceedings*, CEUR-WS, 2022.
- A. B. Bhaumik and M. Das, "Emotions and threat detection in Urdu using transformer models (FIRE 2022 task report)," *CEUR Workshop Proceedings*, CEUR-WS, 2022.
- M. U. Arshad, R. Ali, M. O. Beg, and W. Shahzad, "UHated: Hate speech detection in Urdu using transfer learning," *Language Resources and Evaluation*, SpringerLink, 2023.
- M. Bilal et al., "Roman Urdu hate speech detection using transformer models," *Sensors*, MDPI, 2023.
- V. Rajput et al., "Detection of abusive language for YouTube comments in Urdu and Roman Urdu," *Procedia Computer Science*, Elsevier, 2025.
- H. Saleem, "Performance assessment of ML and DL models in detecting hate speech in mixed English-Roman Urdu datasets," *OAJ AI & ML*, ResearchGate, 2023.
- H. Rizwan, M. H. Shakeel, and A. Karim, "Hate-speech and offensive language detection in Roman Urdu," *Conference Report (LUMS)*, ResearchGate, 2020.
- B. Pervez, "Urdu language abusive text dataset," *GitHub Repository*, 2020. [Online]. Available: <https://github.com/>
- F. Khan, F. K. Saddozai, and S. K. Badri, "Multimodal approaches to hate speech detection in low-resource languages," *PeerJ Computer Science*, 2024.
- J. A. Leite et al., "On multilingual BERT performance for toxicity detection: Monolingual vs. multilingual trade-offs," *Conference Proceedings / Journal Article*, ResearchGate, 2023.
- S. Das et al., "Exploring transformer and ML approaches to abusive and threatening content in Urdu," *Conference/Workshop Proceedings*, ResearchGate, 2022.
- F. Ahmed et al., "Cyberbullying detection in Bengali: Binary and multiclass approaches," *Journal / Conference Proceedings*, ResearchGate, 2022.
- S. Carta et al., "multi-label toxicity classification in online discussions: Lessons from Wikipedia talk pages," *CEUR Workshop Proceedings*, CEUR-WS, 2020.
- K. Georgakopoulos et al., "CNNs vs bag-of-words for offensive language detection: Empirical evidence," *Journal/Conference*, CEUR-WS, 2018.
- G. Akash et al., "Comparative assessment of LSTM, GRU, and transformer models for toxic comment classification," *Conference Proceedings*, CEUR-WS, 2022.
- S. Saha et al., "HateMonitor: Language-agnostic hate speech detection using BERT and LASER embeddings," *FIRE 2019 / HASOC Shared Task Report*, Groningen Research Portal, 2019.
- T. Caselli et al., "HateBERT: Retraining BERT for abusive language detection," *Proc. Workshop on Online Abuse and Harms*, ACL Anthology, 2021.
- L. Khan, A. Amjad, N. Ashraf, and H. T. Chang, "Multiclass sentiment analysis of Urdu text using multilingual BERT," *Scientific Reports*, ICCK, 2022.
- PeerJ / HateInsights Team, "Understanding hate speech: HateInsights dataset and model (Urdu dataset resources)," *PeerJ Computer Science*, 2024.

M. U. Arshad, M. O. Beg, W. Shahzad *et al.*,
“Transfer learning approaches for Urdu
hate-speech and threat detection: Datasets
and baselines,” *Language Resources and
Evaluation*, SpringerLink, 2023.

