Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606

Volume 08 Issue 02
2025

Research Article
https://doi.org/10.56979/802/2025

Sentiment Analysis of Urdu Text using Hybrid Deep Learning Techniques

Muhammad Altaf Ahmad[1], Saima Noreen Khosa[2], Muhammad Zubair Hadi[3], Tanzeela Kiran[1], Wajahat Hussain[3], Urooj Akram[1], and Muhammad Faheem Mushtaq[1,*]

[1]Department of Artificial Intelligence, The Islamia University of Bahawalpur, 63100, Bahawalpur, Pakistan.
[2]Department of Information Technology, Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan 64200, Pakistan.
[3]Department of Computer Science, The Islamia University of Bahawalpur, 63100, Bahawalpur, Pakistan.
[*]Corresponding Author: Muhammad Faheem Mushtaq. Email: faheem.mushtaq@iub.edu.pk

Received: January 15, 2025 **Accepted:** March 01, 2025

Abstract: Sentiment Analysis (SA) has become the rising topic of research in data mining. There is a massive increase in the use of the Internet and business e-commerce applications. With the excess amount of text content on the internet, Sentiment Analysis has begun to attract the attention of more people. Sentiment analysis of single sentence referred as short text, is a very tricky task because single sentence does not have much contextual information. Furthermore, not much work has been done in field of sentiment analysis of Urdu text because a very few amounts of data and recourses of Urdu text are available publicly. The purpose of this research is to contribute in the field of sentiment analysis of Urdu text. Mostly in Natural Language Processing (NLP), tasks words are the main focus. This research proposed a deep learning techniques combination of Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM) for the Sentiment Analysis of Urdu Text. In this technique, word embedding is performed using fastext API which poses a high dimensionality (300 dimensions). Long Short-Term Memory helps capturing the long-term dependencies and minimizes the loss of local information. We validated these proposed techniques on the Urdu sentiment analysis dataset and compare with Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM). The results shows that the proposed technique LSTM-CNN has significantly by achiving testing accuracy 96.9% training accuracy 99.5, precision P 96.0 , precision N 98.0, Recall P 98.0, Recall N 96.0, F1-score N 97.0 , and F1-score N 97.0.The findings shows that CNN is a strong option for the given problem as it improves classification accuracy when combined with LSTM.

Keywords: Sentiment Analysis; Urdu Text; Natural Language Processing; Neural Networks; Deep Learning.

1. Introduction

Sentiment Analysis (SA) is one of the NLP research areas that is currently active and often examines a broad variety of viewpoints on a number of topics that people express on different platforms [1]. The aims of sentiment analysis to identify the sentiment expressed in a text and assign it a label that corresponds to the specific topic being discussed. Urdu sentiment analysis faces key challenges, including a lack of large annotated datasets and limited NLP tools tailored for Urdu. The presence of Roman Urdu and code-mixed text further complicates analysis. Additionally, there is a shortage of pretrained language models and domain-specific research in this area. It has received a lot of attention lately since sentiment is thought to be crucial to preserving the standing of businesses who have direct contact with the public [2].

At the beginning of NLP, the English language has drawn a lot of attention for research at the international level because it is computationally cheap to perform experiments in a language with abundant resources[3]. English has seen the majority of extensive research on all facets of natural language processing during the past two decades because it is challenging to accommodate the word-level complex morphological structure of these languages [4]. Researchers have focused on morphologically rich, yet resource-scarce languages such as Urdu, Arabic, Turkish, Hindi, and Persian[5] . The limited work have been done on Urdu language with the lack of linguistic resources, complex morphology and language's unique features playing a crucial role [6].

There are numerous feature selection techniques that is effective with all types of machine learning classifier among the many available. When compared to typical machine learning models, Deep Learning (DL) models have demonstrated significant improvements in performance across a wide range of data[7]. Furthermore, the deep learning models use Graphic Processing Units (GPUs) to analyze data in parallel that perform better as compare to ML models and can learn complicated features straight from high-dimensional feature space. However, the potential of DL models in Urdu language sentiment analysis has yet to be explored [8].

In order to categorize sentiments in Urdu documents, this research presented a hybrid single layer Multifilter Convolutional Neural Network with Bidirectional Long Short-Term Memory (CNN-LSTM) that combines the architectures of multifilter CNN and [9]. The ability of the multifilter CNN architecture to extract local features in high-dimensional feature space and the potential of LSTM to fully comprehend the meanings of individual words in the context from both sides are the main benefits of combining the both models[10]. Furthermore, LSTM is incorporated to increase the accuracy of the hybrid model for Urdu sentiment classification since CNN disregards the contextual meaning of words in text-based classification [11].

A. Attributes of Urdu

With 300 million speakers worldwide, Urdu is a significant language and the national language of Pakistan. In the world, it is the twenty-first most spoken language. There are 38 characters displayed in its character set [12]. The following distinguishing characteristics, in addition to its intricate morphological structure, make it a difficult language for computer tasks. It is multilingual and has many borrowed words from languages like as English, Arabic, Persian, Turkish, and Sanskrit[13]. Other languages have a big influence on it. There are morphological variants, meaning that numerous words share a base term [14].

B. Classification of Sentiment Analysis Levels

There are three opposing levels in which the sentiment analysis is divided including document-level, sentence-level, and word-level. Word polarity, or whether a word is positive, neutral, or negative, is established at the word level. In the sentence-level, SA analyzes the sentiment score of a sentence by calculating the polarity of the entire sentence. Sentences are collections of words that express a person's viewpoint on a certain topic [15]. This is typically carried out for data that has been taken from social networking sites. Sentiment analysis seeks to categorize the polarity of an entire document at the document level by considering an increase in words and noise characteristics that can skew learning and make it more difficult to forecast the document's sentiment. The contribution of the research are as follows:

- A hybrid deep learning model is proposed by considering the advantages of Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM) for document-level sentiment analysis of Urdu text.
- The performance of the proposed and baseline deep learning models is analyzed on variable-sized dataset using evaluation parameters such as accuracy, precision, recall, and F1-score.
- Using popular DL classifiers, a comparative examination of CNN-LSTM performance is conducted with convolutional neural network, recurrent neural network, and long short-term memory.

The rest of the paper is structured as follows: The literature review is presented in Section II. Section III discusses the proposed methodology. Section IV explains the results and discussion of the research including experimental setup and performance evaluation. Section V conclude the conclusion and future work. 2.

2. Related Work

Sentiment Analysis (SA) has emerged as one of the most popular study subjects in recent years. This section examined the classification schemes for Urdu-written textual data that have been employed in the literature [16]. An analysis of the morphologically complex and rich Urdu dialect used a linguistically driven sentiment recognition algorithm [17]. Due to Urdu's complex and morphologically rich grammar, a better or completely new approach is required [18]. Due to that, the research focus on locating SentiUnits in the provided text rather than subjective phrases [19]. To connect SentiUnits to their targets, they employed chunking for shallow parsing. For their model training, they create an annotated sentiment lexicon of Urdu words [20]. A sentiment, such as good or negative is assigned to each lexicon [21]. User feedback from movies and electrical devices were compiled into two datasets [22].

Particularly when creating Urdu sentiment analyzer, they have meticulously outlined guidelines for allocating polarities to intensifiers if they are encircled by either a positive or negative term [23]. A total of 6025 sentences were gathered from 151 blog entries [24]. To give positive and negative terms from the data polarity, a lexicon was developed [25]. Their suggested solution obtained 83% accuracy in Urdu sentiment categorization by utilizing their own lexical resources in conjunction with a rule-based system [26] . Evidently supervised machine learning classifiers are less accurate than Urdu sentiment analyzers [27]. Implementing a lexicon-based strategy is simple, but creating a lexicon is a laborious process in contrast to using DL models [28].

The development of deep learning methods for Urdu SA is still in its early stages. For Roman Urdu and other resource-rich languages, a significant amount of work has been done utilizing machine learning and deep learning techniques [29]. The field of deep learning for sentiment analysis of native Urdu is still unexplored [30]. By using a hybrid deep learning technique that consists of a CNN layer with multi-size filters followed by a BiLSTM layer, the use of suggested architecture and baseline deep learning models tends to solve the gap that has been noted in the literature. The suggested method offers improved classification accuracy for data sets of various sizes [31].

The CNN's simplified architecture, which includes multisize filters, makes it possible to extract hierarchical features, which could improve classification performance and provide a more sophisticated understanding of document content [32]. The summary does not go into depth about the precise experimental outcomes and performance indicators, but it is anticipated that the study will provide light on how well the suggested model works in comparison to current techniques [33]. The overall relevance of the suggested document-level text classification technique would probably be enhanced by comments of the model's advantages, disadvantages, and applications as well as recommendations for future study avenues [34].

The Urdu Text Sentiment Analysis (UTSA) framework uses deep learning algorithms and word vector representations [35] . Sequential models employ stacked layers, and the function of unsupervised self-trained and pre-trained embedding models is investigated [36]. The results show that BiLSTM-ATT performs better than other models, with an impressive 72.7% F1 score and 77.9% accuracy. The work highlights the need for more research in morphologically rich languages like Urdu while highlighting the efficacy of deep learning techniques in Urdu sentiment analysis [37].

This language requires minimal resources used deep learning methods, such as Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN), to categorize Pashtu social media content into three groups: neutral, negative, and positive [38]. Sentiment analysis in Pashtu-script-like Urdu, a language with limited resources[39]. They identified issues like a lack of language resources and annotated datasets after analyzing more than 50 studies [40]. The study offers insights into the intricacies of Urdu's morphology and syntax by examining a variety of methodologies, such as lexicon-based and machine learning techniques [41].The summary of related work show in Table 1.

Table Error! No text of specified style in document.**1.** Summary of the Related Work

References	Models	Results
[42]	CNN+LSTM	Accuracy = 90.0%
[43]	MSA	Accuracy = 91.23%
[44]	BILSTM-ATT	Accuracy = 77.9%
[45]	LSTM	Accuracy = 75.963%
[46]	RNN	Accuracy = 71.0%
[47]	mBert	Accuracy = 77.61%

Sentiment analysis with regard to Roman Urdu and English dialects, particularly on social media platforms [48] . This deep learning architecture to address these problems by combining long short-term memory for long-term dependence preservation with a one-layer CNN model for local feature extraction [49]. According to the study's findings, which involve extensive testing on four corpora, the proposed model excels at classifying sentiment in both Roman and Urdu text. Specifically, the model receives high accuracy scores of 0.841, 0.740, and 0.748 on the MDPI, RUSA, RUSA-19, and UCL datasets, respectively [50]. The studies' findings demonstrate that the SVM classifier and the Word2Vec CBOW model perform effectively for sentiment analysis in Roman Urdu [51].

3. Proposed Methodology

The methodology starts with the collection of the datasets, and then proceeds complete preprocessing slaves such as label encoding, synonym replacement, text tokenization, padding the sequence, one-hot encoding and data augmentation to enhance the dataset as show in Fig.1. Once the data is processed, it is split into train and test datasets. The proposed hybrid deep learning model (CNN for feature extraction and LSTM for learn sequential dependencies) is applied to the classification. Finally, a performance of the model on text classification using precision, recall, F1-score, and accuracy measures.

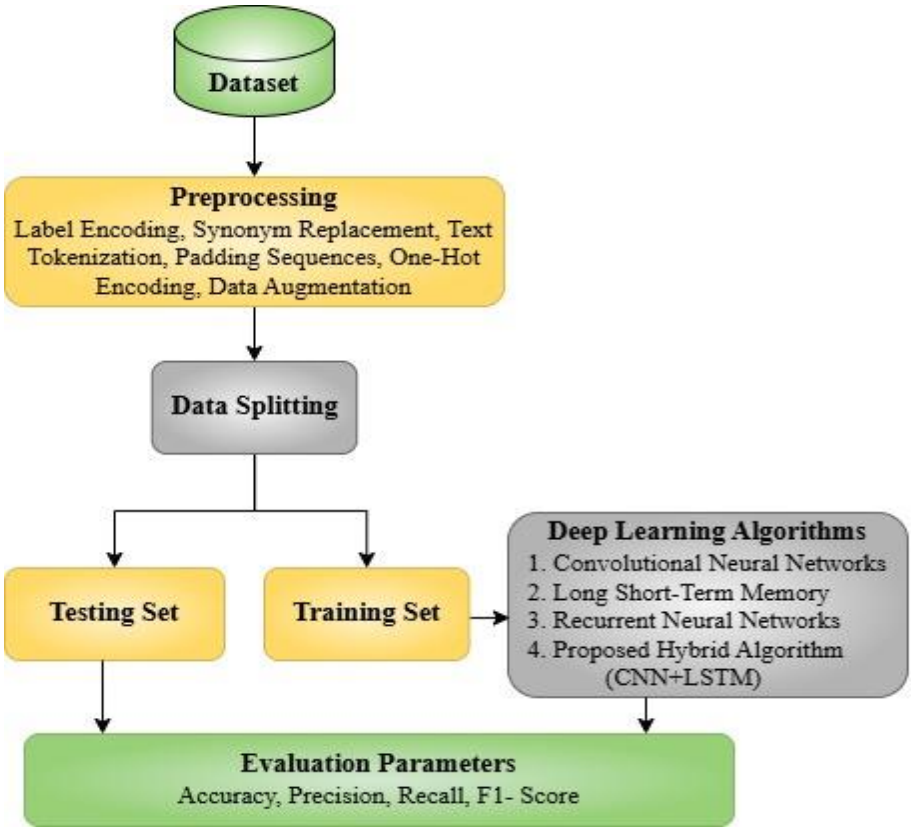

Figure 1. Proposed Methodology

3.1 Dataset Description

An Urdu sentiment analysis corpus this is textual data stored in a CSV file format with sentiment class labels. "Class," which includes sentiment labels like positive, negative, or neutral, and "Tweet," which has the matching textual data in Urdu, are the two main columns into which the dataset is arranged. The study can investigate several sentiment classification techniques and their efficacy on the provided corpus to this dataset, which forms the basis for assessing and categorizing sentiments expressed in Urdu tweets. Fig. 2 displays the dataset's visual representation.

3.2 Preprocessing

Preprocessing is performed on the dataset, which includes handling missing values, removing excess spaces, and using label encoder to encode sentiment labels. Class weights are calculated to guarantee equitable training since the distribution of classes is examined for possible imbalance. Furthermore, the dataset is improved by applying data augmentation techniques including synonym substitution, random word insertion, and deletion, which produce more varied samples and, in turn, improve model performance and generalization.

3.2.1. Label Encoding

The label encoding process to converts categorical sentiment labels into numerical values using scikit-learns Label Encoder. The encoder is fitted to the text labels, transforming them into integers. The number of unique classes is extracted to configure the neural network's output layer. To handle class imbalance, weights are computed based on label distribution. Numerical labels are converted to one-hot encoded vectors using Kera's' to categorical for compatibility with the model's categorical cross-entropy loss function. This preprocessing ensures textual labels are properly formatted for Deep Learning.

3.2.2. Synonym Replacement

The synonym replacement for text augmentation to increase dataset diversity. It works by. Splitting Urdu phrases into words, randomly selecting words for replacement, retrieving Urdu synonyms from WordNet, then replacing original words if appropriate synonyms exist. While this strategy theoretically expands training data, it has little effectiveness in Urdu. This phase comes before tokenization and normalization in the preprocessing pipeline.

3.2.3 Tokenization

Tokenization turns raw Urdu text into numerical sequences using Keres' Tokenizer. The tokenizer creates a vocabulary by selecting all of the dataset's unique words. It then converts each Urdu words into integer sequences, with words substituted by their respective indices. To ensure uniform input length, sequences are padded or shortened to a set length of 100 tokens. While this method effectively prepares text for neural networks, it employs basic whitespace splitting and lacks Urdu specific rules for complex words and punctuation handling a drawback that may be remedied with dedicated Urdu tokenizers.

3.2.4 Padding Sequence

The padding stage converts tokenized Urdu sequences into fixed-length vectors using Keres' pad sequence's function. All sequences are modified to a consistent length of 100 tokens shorter sequences have zero padding at the end, while longer sequences are shortened from the start to maintain recent context. This normalization maintains dimensional consistency for neural network input while preserving Urdu's right-to-left word order. The zero-padding strategy avoids introducing linguistic bias, but it does not expressly address Urdu punctuation signs. This crucial preprocessing step connects tokenization and model input, allowing for fast batch processing while balancing information preservation with computing constraints common to Urdu sentiment text analysis.

3.2.5 One-Hot Encoding

The numerical sentiment labels are converted into binary vectors using Kera's' to categorical function, with each class resulting in a vector with '1' at its index position and '0' elsewhere. This allows for compatibility with the model's categorical cross-entropy loss function while maintaining label semantics in Urdu sentiment analysis.

3.2.6. Data Augmentation

Three text-editing strategies are used to increase the Urdu dataset synonym replacement, random insertion and random deletion. Despite WordNet's English focus, these methods artificially improve sample diversity to minimize overfitting.

Figure 2. Urdu sentiment Dataset

3.3 Deep Learning Models

Deep learning sentiment analysis uses neural networks to automatically identify and categorize emotions in textual data, such reviews, tweets, or customer reviews. Convolutional neural networks and recurrent neural networks are examples of deep learning models that are trained on large labeled datasets to identify patterns that correspond to various sentiments, such as positive, negative, or neutral. These models are very useful for sentiment analysis tasks because they can accurately capture context, semantic meaning, and subtle linguistic cues.

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	?	0 (unbuilt)
conv1d (Conv1D)	?	0 (unbuilt)
max_pooling1d (MaxPooling1D)	?	0 (unbuilt)
dropout (Dropout)	?	0 (unbuilt)
conv1d_1 (Conv1D)	?	0 (unbuilt)
max_pooling1d_1 (MaxPooling1D)	?	0 (unbuilt)
dropout_1 (Dropout)	?	0 (unbuilt)
conv1d_2 (Conv1D)	?	0 (unbuilt)
max_pooling1d_2 (MaxPooling1D)	?	0 (unbuilt)
dropout_2 (Dropout)	?	0 (unbuilt)
flatten (Flatten)	?	0 (unbuilt)
dense (Dense)	?	0 (unbuilt)
dropout_3 (Dropout)	?	0 (unbuilt)
dense_1 (Dense)	?	0 (unbuilt)

Total params: 0 (0.00 B)
Trainable params: 0 (0.00 B)
Non-trainable params: 0 (0.00 B)

Figure 3. CNN model Layer Summery

3.3.1 Convolutional Neural Networks

Convolutional neural network based deep learning-based sentiment categorization model using a dataset of Urdu text. In order to preprocess the dataset, missing values are handled, columns are renamed, and sentiment labels are encoded. Data augmentation methods like synonym substitution, random word insertion, and deletion are used to create more training examples in order to rectify class imbalance and improve generalization. To ensure consistent input size, the text data is tokenized and transformed into padded sequences. Three convolutional layers, max pooling, regularization dropout layers, and a final dense output layer with the SoftMax activation function are all included in the architecture of a CNN model. To reduce bias, the model is trained with class-weighted balancing and categorical cross-entropy loss using the Adam optimizer. Accuracy measurements, a classification report, and a confusion matrix on the test set are used to assess model performance, and accuracy and loss graphs are used to depict the training process. The CNN model layer summery shown in Fig. 3.

3.3.2 Long Short-Term Memory

LSTM system the first phase of the process involves data preprocessing, including the encoding of class labels and missing values. To combat data imbalance and enhance generalization, data augmentation techniques such as substitution of synonyms, random insertion/deletion of words are applied. Tokenizing and padding the sequence prepares textual data for deep learning. An LSTM model with several layers and dropout regularization is trained with Adam optimizer with a small learning rate, categorical cross-entropy loss, and class weights to tackle imbalance problems.

Accuracy, classification reports, and a confusion matrix are used to assess the model, and loss and accuracy charts are used to illustrate training results. In order to avoid overfitting and guarantee ideal generalization, early halting is used. The CNN model layer summery shown in Fig. 4.

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	?	0 (unbuilt)
bidirectional (Bidirectional)	?	0 (unbuilt)
dropout (Dropout)	?	0 (unbuilt)
bidirectional_1 (Bidirectional)	?	0 (unbuilt)
dropout_1 (Dropout)	?	0 (unbuilt)
lstm_2 (LSTM)	?	0 (unbuilt)
dropout_2 (Dropout)	?	0 (unbuilt)
dense (Dense)	?	0 (unbuilt)
dropout_3 (Dropout)	?	0 (unbuilt)
dense_1 (Dense)	?	0 (unbuilt)

Total params: 0 (0.00 B)

Trainable params: 0 (0.00 B)

Non-trainable params: 0 (0.00 B)

Figure 4. LSTM MODEL Layer Summery

3.3.3 Recurrent Neural Network

Recurrent Neural Network based model for classifying sentiment in Urdu text that includes preprocessing, augmentation, training, and assessment of the data. The dataset is cleaned, tokenized, and padded to a specified length, prior to splitting into training and testing set. Using model generalization, three data augmentation techniques such as synonym replacement, random insertion and random deletion are used. A simple recurrent neural network model is built using an embedding layer, recurrent layer, and dropout layers followed by fully connected layers and compiled using categorical cross-entropy loss with the Adam optimizer. The class imbalance has been rectified by training the model on a smaller dataset (40 percent of the training data) along with calculated class weights. A performance evaluation was conducted using accuracy, loss curves, a classification report, and a confusion matrix to demonstrate the efficiency of the RNNs in sentiment analysis.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	?	0 (unbuilt)
simple_rnn_1 (SimpleRNN)	?	0 (unbuilt)
dropout_2 (Dropout)	?	0 (unbuilt)
dense_2 (Dense)	?	0 (unbuilt)
dropout_3 (Dropout)	?	0 (unbuilt)
dense_3 (Dense)	?	0 (unbuilt)

Total params: 0 (0.00 B)

Trainable params: 0 (0.00 B)

Non-trainable params: 0 (0.00 B)

Figure 5. RNN model Layer Summery

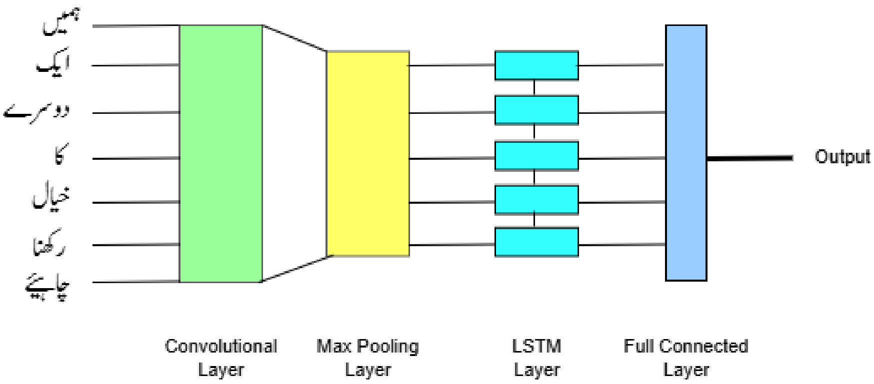

Figure 6. Proposed Hybrid CNN-LSTM Model

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	?	0 (unbuilt)
conv1d (Conv1D)	?	0 (unbuilt)
max_pooling1d (MaxPooling1D)	?	0 (unbuilt)
lstm (LSTM)	?	0 (unbuilt)
dropout (Dropout)	?	0 (unbuilt)
lstm_1 (LSTM)	?	0 (unbuilt)
dropout_1 (Dropout)	?	0 (unbuilt)
dense (Dense)	?	0 (unbuilt)
dropout_2 (Dropout)	?	0 (unbuilt)
dense_1 (Dense)	?	0 (unbuilt)

Total params: 0 (0.00 B)

Trainable params: 0 (0.00 B)

Non-trainable params: 0 (0.00 B)

Figure 7. CNN-LSTM model Layer Summery

3.3.4 Proposed Hybrid Model

A sentiment classification for Urdu texts, hybrid CNN-LSTM model is proposed that integrates data preprocessing, augmentation, training, and evaluation. The dataset is cleaned, tokenized, and padded to a predefined sequence length before being split into training and testing sets. To enhance model generalization, data enrichment techniques include synonym substitution, random insertion, and random deletion.

A CNN-LSTM model is constructed of an embedding layer, convolutional and max-pooling layers for feature extraction, and stacked LSTM layers for sequential learning with dropout regularization shown in Fig. 6. The model is trained for 40 epochs using the Adam optimizer and categorical cross-entropy loss, using class weights that have been adjusted to handle imbalance. Performance is assessed using accuracy, loss curves, a classification report, and a confusion matrix, demonstrating the effectiveness of CNN and LSTM in tandem for sentiment analysis. The CNN model layer summery show in Fig. 7.

3.4 Evaluation Parameters

The performance evaluation of intrusion detection algorithms includes a variety of criteria that assign numerical values to their effectiveness. These metrics represent different aspects of an algorithm's class correctness, resilience, and proficiency in identifying different types of cyberthreats. When evaluating intrusion detection, a few common parameters are employed.

The accuracy is calculated by dividing the number of correctly predicted events by the entire number of cases, indicates the overall accuracy of the model. FN, FP, TN, and TP are the acronyms for False Negatives, False Positives, and True Positives, respectively.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100 \tag{1}$$

By displaying the proportion of real positive predictions among all positive predictions, precision provides the percentage of positive forecasts.

$$Precision = \frac{TP}{TP + FP} * 100 \tag{2}$$

Recall assesses the model's ability to differentiate real positive cases from all other positive examples.

$$\mathrm{Re}call = \frac{TP}{TP + FN} * 100 \tag{3}$$

The F1-score, which is the harmonic mean of accuracy and recall, is the only statistic that balances false positives and false negatives.

$$F1 - score = 2 * \frac{\mathrm{Pr}ecision * \mathrm{Re}call}{\mathrm{Pr}ecision + \mathrm{Re}call} \tag{4}$$

4. Results and Discussion

The CNN, LSTM, and RNN models are all outperformed by the suggested CNN-LSTM model as shown in the Table 3. The CNN-LSTM hybrid is the most successful model for Urdu sentiment analysis, obtaining the best precision, recall, F1-score, and training and testing accuracy. The contextual understanding of sentiment in text is improved by the combination of CNN's capacity to extract spatial characteristics and LSTM's prowess in capturing long-term dependencies. Additionally, because of its effective learning of both local and sequential patterns, the CNN-LSTM model consistently demonstrate a performance gain in accuracy as the number of epochs grows. The thorough results demonstrate the CNN-LSTM model's supremacy in deep learning-based sentiment analysis by confirming its dependability in correctly classifying sentiment in Urdu text.

4.1 Performance of CNN Model

The CNN model's performance across various training epochs (10, 20, 30, 40, and 50) with accuracy values for multiple classes or metrics. As training progresses, accuracy generally improves, with the highest values observed around epochs 40 and 50. At epoch 10, the model achieves an average accuracy of approximately 95-98%, indicating strong initial performance. By epoch 20, there is a noticeable improvement, particularly in some metrics reaching 99%, indicating that the model. At epoch 40, the model recovers with the highest recorded accuracy values, showing stability and robustness in predictions. By epoch 50, accuracy remains consistent, reinforcing that the model has reached optimal learning capacity. According to the results, the model works best between epochs 40 and 50, which makes it a good range to finish training as shown in the Table 2.

Table 2. CNN Model Performance

Epoch	Testing %	Training %	Prec.(P) %	Prec.(N) %	Recall (P) %	Recall (N) %	F1 score(N) %	F1 Score(P) %
10	95.8	98.1	94.0	98.0	98.0	94.0	96.0	96.0
20	97.1	98.5	96.0	99.0	99.0	95.0	97.0	97.0
30	95.8	98.4	96.0	95.0	95.0	96.0	96.0	96.0
40	97.8	98.5	96.0	99.0	99.0	96.0	98.0	98.0
50	98.3	98.7	97.0	99.0	99.0	97.0	98.0	98.0

4.2 Performance Evaluation of RNN

The performance of the RNN model with accuracy values for several classes or evaluation criteria over several training epochs (10, 20, 30, 40, and 50). The model's accuracy at epoch 10 is moderate, varying somewhat across measures and falling between about 80% and 96.1%. The model appears to be learning and adjusting as training advances to epoch 20, as evidenced by the minor improvement in accuracy for the majority of criteria. Nevertheless, a slight decline in accuracy is noted around epoch 30, especially for lower values for some classes, which may be a sign of overfitting or model instability. Accuracy results vary little by epoch 40, when performance stabilizes but does not significantly improve. At epoch 50, the model's overall performance remains consistent with very minor accuracy variations. The results show that the model works best between epochs 20 and 40, but additional fine-tuning could be necessary to enhance generalization and prevent overfitting as shown in the Table 3.

Table 3. RNN model Performance

Epoch	Testing %	Training %	Prec.(P) %	Prec.(N) %	Recall(P) %	Recall(N) %	F1 Score (N) %	F1 Score (P) %
10	80.7	96.1	81.0	80.0	80.0	82.0	81.0	80.0
20	81.5	96.3	83.0	80.0	80.0	83.0	82.0	81.0
30	79.3	95.8	80.0	79.0	79.0	80.0	79.0	79.0
40	79.7	96.3	81.0	79.0	78.0	81.0	80.0	79.0
50	80.1	96.0	80.0	80.0	80.0	81.0	80.0	80.0

4.3 Performance Evaluation of LSTM

The LSTM model's performance throughout a variety of training epochs (10 to 50). At epoch 10, the model achieves a high accuracy of approximately 95.0% across most metrics, with a peak value of 99.0% in one instance. The accuracy exhibits minimal fluctuation as training progresses to epoch 20, increasing somewhat to 99.1% in one metric. At epoch 30, the model still exhibits comparable behavior, indicating that convergence has been attained. However, at epoch 40, a minor drop in one category (94.0%) is observed, suggesting a potential variation in generalization ability. There are no appreciable performance increases or losses by epoch 50, indicating that there are no benefits to additional training after epoch 30.

All things considered, these results demonstrate how stable and useful the model is; nonetheless, additional regularization or fine-tuning techniques could be explored to enhance generality as shown in the Table 4.

Table 4. LSTM Model Performance

Epoch	**Testing %**	**Training %**	**Prec.(P) %**	**Prec.(N) %**	**Recall(P) %**	**Recall(N) %**	**F1 score(N) %**	**F1 Score(P) %**
10	95.0	99.0	95.0	95.0	95.0	95.0	95.0	95.0
20	94.9	99.1	95.0	95.0	95.0	95.0	95.0	95.0

30	95.3	99.1	95.0	95.0	95.0	95.0	95.0	95.0
40	94.9	99.1	95.0	94.0	94.0	95.0	95.0	95.0
50	94.9	99.1	95.0	95.0	95.0	95.0	95.0	95.0

4.4 Performance Evaluation of Proposed Hybrid Model

The Proposed CNN-LSTM model consistently demonstrates good accuracy across several test runs, with values averaging between 96 and 99% for the various metrics assessed. It performs consistently, showing consistency and resilience in its predictions with little variation in accuracy. Although there are some minor differences across runs, they fall within a reasonable range, indicating that the suggested model is generalizing effectively across test sets. These outcomes demonstrate how well the proposed model handled the provided categorization task, providing compelling evidence of its applicability in practical settings as shown in the Table 5. The confusion matrix of proposed model shown in the Fig. 7.

Table 5. CNN-LSTM MODEL Performance

Epoch	Testing %	Training %	Prec.(P) %	Prec.(N) %	Recall(P) %	Recall(N) %	F1 score(N) %	F1 Score(P) %
10	96.6	99.3	95.0	98.0	98.0	95.0	97.0	97.0
20	96.6	99.4	96.0	98.0	98.0	95.0	97.0	97.0
30	96.9	99.5	96.0	98.0	98.0	96.0	97.0	97.0
40	96.5	99.5	95.0	98.0	98.0	95.0	96.0	97.0
50	96.9	99.5	96.0	98.0	98.0	96.0	97.0	97.0

4.5 Performance Analysis of All Models

The CNN-LSTM model, as proposed, performs well in sentiment analysis, obtaining an amazing testing accuracy of 96.9%. This hybrid model successfully captures both local patterns and long-term dependencies in textual data by fusing the advantages of Long Short-Term Memory (LSTM) networks for sequence learning and Convolutional Neural Networks (CNNs) for feature extraction. With F1 scores of 97.0%, the model demonstrates balanced precision, recall, and F1 scores for both positive and negative sentiments, underscoring its capacity to correctly identify sentiments across various classes. For applications involving both spatial and temporal feature processing, including sentiment analysis, the CNN-LSTM model is a potent option because it performs better than the conventional CNN and LSTM models separately. Its exceptional performance raises the possibility of wider uses in problems involving natural language processing as shown in Table.6.

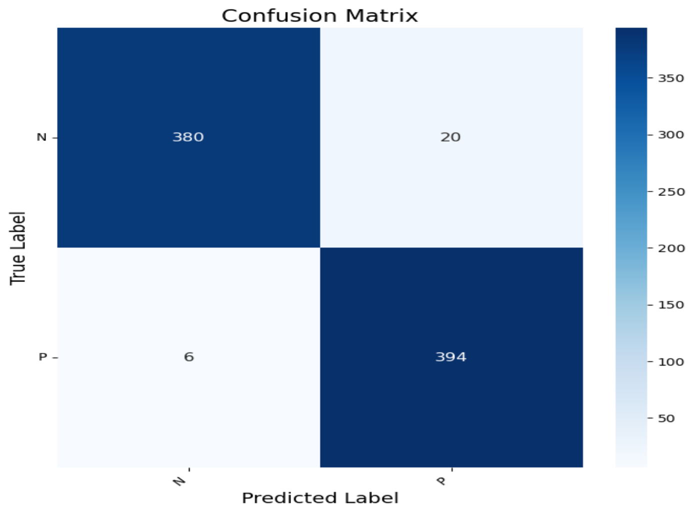

Figure 7. Proposed Hybrid CNN-LSTM Model

The CNN-LSTM model, which combines convolutional neural networks for feature extraction with long short-term memory networks for sequence modeling, had 99.5% training and 96.9% testing accuracy. In comparison, the FastText API, which employs a shallow neural network and represents text as a bag of words or n-grams, often has shorter training times and cheaper processing costs, making it suited for large-scale and real-time applications. However, FastText often provides moderate accuracy, ranging from 85% to 95% on similar tasks, depending on data quality and preprocessing. It is extremely efficient however it may not capture complex syntactic and contextual nuances as well as deep models such as CNN-LSTM. It is highly efficient but may not capture intricate syntactic and contextual nuances as effectively as deep models like CNN-LSTM.

Table 6. Performance Analysis of All Models

Technique	**Testing %**	**Training %**	**Prec.(P) %**	**Prec.(N) %**	**Recall(P) %**	**Recall(N) %**	**F1 score(N) %**	**F1 Score(P) %**
CNN	98.3	98.7	97.0	99.0	99.0	97.0	98.0	98.0
RNN	79.7	96.3	81.0	79.0	78.0	81.0	80.0	79.0
LSTM	94.9	99.1	95.0	95.0	95.0	95.0	95.0	95.0
Proposed Model	96.9	99.5	96.0	98.0	98.0	96.0	97.0	97.0

5. Conclusion and Future Work

This research proposed hybrid model for Urdu sentiment analysis that combines Long Short-Term Memory networks with Convolutional Neural Networks (CNN). While the LSTM component analyzes the text's long-term dependencies, the CNN component effectively collects important features and local word patterns. The hybrid approach improves the generality and accuracy of sentiment classification by combining these models. Throughout several test runs, the suggested model regularly achieves excellent accuracy; the highest recorded accuracy was 99.5%. This peak performance shows how well the model handles the specified categorization task and how robust it is. Strong generalization skills are demonstrated by the small accuracy changes between runs, which makes the suggested CNN-LSTM ensemble a dependable option for practical Urdu sentiment analysis applications. Sentiment analysis of Urdu text is difficult because of its limited labeled data, intricate language, and rich morphology. In order to improve sentiment categorization, deep learning approaches automatically extract features. The proposed hybrid model improves accuracy and generalization by processing long-term dependencies while CNN captures local patterns. By surpassing conventional methods, our model achieves state-of-the-art performance. Future research will examine multi-modal sentiment analysis, integrate transformer-based models such as BERT, increase datasets, and optimize for real-time social media and opinion mining applications.

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

References

1. M. A. Javaid et al., "LSD: Discrimination of Coal Mining Accident's Causes Based on Ensemble Machine Learning," in International Conference on Soft Computing and Data Mining, Springer, 2022, pp. 390–400.
2. A. Khan et al.,"Abusive language detection in urdu text: Leveraging deep learning and attention mechanism," IEEE Access, 2024.
3. K. Aziz, D. Ji, B. Li, F. Li, and J. Zhou, "Advancing Urdu NLP: Aspect-Based Sentiment Analysis with Graph Attention Networks," in 2024 International Joint Conference on Neural Networks (IJCNN), 2024, pp. 1–8.
4. A. Ali et al., "Smart stove automation using wireless network and mobile application," in Modern Industrial IoT, Big Data and Supply Chain: Proceedings of the IIoTBDSC 2020, Springer, 2021, pp. 369–388.
5. A. Ullah et al., "Threatening language detection from Urdu data with deep sequential model," PLoS One, vol. 19, no. 6, p. e0290915, 2024.
6. B. Mughal et al.,"Classification of Breast Lesions in Combination with Metamorphic Segmentation and Saliency Feature Block," in International Conference on Intelligent Technologies and Applications, Springer, 2019, pp. 573–580.
7. M. A. Abid et al., "Comparative analysis of TF-IDF and loglikelihood method for keywords extraction of twitter data," Mehran University Research Journal of Engineering and Technology, vol. 42, no. 1, pp. 88–94, 2023.
8. Amna et al., "The Comparative Performance Analysis of Clustering Algorithms," in International Conference on Soft Computing and Data Mining, Springer, 2022, pp. 341–352.
9. A. Ishaq et al., "Extensive hotel reviews classification using long short term memory," J Ambient Intell Humaniz Comput, vol. 12, pp. 9375–9385, 2021.
10. M. Khan, A. Naseer, A. Wali, and M. Tamoor, "A roman Urdu corpus for sentiment analysis," Comput J, vol. 67, no. 9, pp. 2864–2876, 2024.
11. M. F. Mushtaq et al., "Key schedule algorithm using 3-dimensional hybrid cubes for block cipher," International Journal of Advanced Computer Science and Applications, vol. 10, no. 8, 2019.
12. W. Hussain et al., "Ensemble genetic and CNN model-based image classification by enhancing hyperparameter tuning," Sci Rep, vol. 15, no. 1, p. 1003, 2025.
13. M. R. Ashraf et al., "Revolutionizing Urdu Sentiment Analysis: Harnessing the Power of XLM-R and GPT-2," IEEE Access, 2024.
14. U. Iqbal et al., "Functional expansions based multilayer perceptron neural network for classification task," Computación y Sistemas, vol. 22, no. 4, pp. 1625–1635, 2018.
15. S. H. H. Huspi et al., "Sentiment analysis on roman urdu students' feedback using enhanced word embedding technique," Baghdad Science Journal, vol. 21, no. 2 (SI), p. 725, 2024.
16. M. Saeed et al., "In-depth Urdu Sentiment Analysis Through Multilingual BERT and Supervised Learning Approaches," IECE Transactions on Intelligent Systematics, vol. 1, no. 3, pp. 161–175, 2024.
17. S. Y. Siddiqui et al., "Intelligent breast cancer prediction empowered with fusion and deep learning," Computers, Materials and Continua, vol. 67, no. 1, pp. 1033–1049, 2021.
18. M. Shahroz et al., "RUTUT: roman Urdu to Urdu translator based on character substitution rules and unicode mapping," IEEE Access, vol. 8, pp. 189823–189841, 2020.
19. L. Khan et al., "Empowering Urdu sentiment analysis: an attention-based stacked CNN-Bi-LSTM DNN with multilingual BERT," Complex & Intelligent Systems, vol. 11, no. 1, p. 10, 2025.
20. M. F. Mushtaq et al., "BHCNet: neural network-based brain hemorrhage classification using head CT scan," IEEE Access, vol. 9, pp. 113901–113916, 2021.
21. A. Altaf et al., "Aspect-based sentiment analysis in Urdu language: resource creation and evaluation," Neural Comput Appl, pp. 1–17, 2024.
22. F. Rustam et al., A. D. Jurcut, and I. Ashraf, "Denial of service attack classification using machine learning with multi-features," Electronics (Basel), vol. 11, no. 22, p. 3817, 2022.
23. A. Ali, M. Khan, K. Khan, R. U. Khan, and A. Aloraini, "Sentiment Analysis of Low-Resource Language Literature Using Data Processing and Deep Learning.," Computers, Materials & Continua, vol. 79, no. 1, 2024.
24. M. A. Abid et al., "Recommending domain specific keywords for twitter," in Recent Advances on Soft Computing and Data Mining: Proceedings of the Fourth International Conference on Soft Computing and Data Mining (SCDM 2020), Melaka, Malaysia, January 22– 23, 2020, Springer, 2020, pp. 253–263.
25. M. Zulqarnain et al., "Efficient processing of GRU based on word embedding for text classification," JOIV: International Journal on Informatics Visualization, vol. 3, no. 4, pp. 377–383, 2019.
26. S. F. Rubab et al., "The Comparative Performance of Machine Learning Models for COVID-19 Sentiment Analysis," in International Conference on Soft Computing and Data Mining, Springer, 2022, pp. 371–379.
27. T. A. Khan et al., "SN2SMS: Location-Aware Social Network Based SMS Health-Care System," International Journal of Computer Science Issues, vol. 11, no. 4, 2014.
28. S. Aziz et al., "Roman Urdu sentiment analysis using machine learning with best parameters and comparative study of machine learning algorithms," Pakistan Journal of Engineering and Technology, vol. 3, no. 2, pp. 172–177, 2020.
29. N. Ahmed, R. Amin, H. Aldabbas, M. Saeed, M. Bilal, and H. Song, "A Novel Approach for Sentiment Analysis of a Low Resource Language Using Deep Learning Models," ACM Transactions on Asian and Low-Resource Language Information Processing, 2024.

30. U. Akram et al., “An improved pi-sigma neural network with error feedback for physical time series prediction” Int. J. Adv. Trends Comput. Sci. Eng, vol. 8, pp. 1-7, 2019.
31. L. Khan, A. Amjad, N. Ashraf, and H.-T. Chang, “Multi-class sentiment analysis of urdu text using multilingual BERT,” Sci Rep, vol. 12, no. 1, p. 5436, 2022.
32. L. Khan, A. Amjad, K. M. Afaq, and H.-T. Chang, “Deep sentiment analysis using CNN-LSTM architecture of English and Roman Urdu text shared in social media,” Applied Sciences, vol. 12, no. 5, p. 2694, 2022.
33. M. Akhtar, R. S. Shoukat, and S. U. Rehman, “A machine learning approach for Urdu text sentiment analysis,” Mehran University Research Journal of Engineering & Technology, vol. 42, no. 2, pp. 75–87, 2023.
34. M. Asghar et al., “Finding correlation between content based features and the popularity of a celebrity on Twitter,” International Journal of Computer Science Issues, vol. 11, no. 04, p. 177-181, 2014.
35. I. Safder et al., “Sentiment analysis for Urdu online reviews using deep learning models,” Expert Syst, vol. 38, no. 8, p. e12751, 2021.
36. T. Kiran et al., “Textile Defect Detection in the textile industry using Deep Learning,” Journal of Computing & Biomedical Informatics, vol. 8, no. 02, 2025.
37. K. Bin Muhammad and S. M. A. Burney, “Innovations in urdu sentiment analysis using machine and deep learning techniques for two-class classification of symmetric datasets,” Symmetry (Basel), vol. 15, no. 5, p. 1027, 2023.
38. U. Sehar, S. Kanwal, K. Dashtipur, U. Mir, U. Abbasi, and F. Khan, “Urdu sentiment analysis via multimodal data mining based on deep learning algorithms,” IEEE Access, vol. 9, pp. 153072–153082, 2021.
39. H. Ghulam, F. Zeng, W. Li, and Y. Xiao, “Deep learning-based sentiment analysis for roman urdu text,” Procedia Comput Sci, vol. 147, pp. 131–135, 2019.
40. H. Sattar et al., “Smart wound hydration monitoring using biosensors and fuzzy inference system,” Wirel Commun Mob Comput, vol. 2019, no. 1, p. 8059629, 2019.
41. L. Khan, A. Amjad, N. Ashraf, H.-T. Chang, and A. Gelbukh, “Urdu sentiment analysis with deep learning methods,” IEEE access, vol. 9, pp. 97803–97812, 2021.
42. U. Naqvi, A. Majid, and S. A. Abbas, “UTSA: Urdu text sentiment analysis using deep learning methods,” IEEE Access, vol. 9, pp. 114085–114094, 2021.
43. A. Karim et al., “Anticipating impression using textual sentiment based on ensemble LRD model,” Expert Syst Appl, vol. 263, p. 125717, 2025.
44. A. Ghaffar et al., “Refined Sentiment Analysis by Ensembling Technique of Stacking Classifier,” Proceeding in International Conference on Soft Computing and Data Mining, Springer, 2022, pp. 380–389.
45. M. Rizwan et al., “Depression classification from tweets using small deep transfer learning language models,” IEEE Access, vol. 10, pp. 129176–129189, 2022.
46. M. I. Liaqat, M. A. Hassan, M. Shoaib, S. K. Khurshid, and M. A. Shamseldin, “Sentiment analysis techniques, challenges, and opportunities: Urdu language-based analytical study,” PeerJ Comput Sci, vol. 8, p. e1032, 2022.