# A Novel Approach for Sentiment Analysis of a Low Resource Language Using Deep Learning Models

Naeem Ahmed

School of Software, Nanjing University of Information Science and Technology, Nanjing 210044, China. naeem.uoh@gmail.com

Rashid Amin*

Department of Computer Science & IT, University of Chakwal, Punjab, 48800, Pakistan. rashid4nw@gmail.com

Department of Computer Science, University of Engineering and Technology, Taxila, Pakistan

Hamza Aldabbas

Prince Abdullah bin Ghazi Faculty of Information and Communication Technology, Al-Balqa Applied University, Al-Salt- Jordan., AlDabbas@bau.edu.jo

Muhammad Saeed

School of Software, Nanjing University of Information Science and Technology, Nanjing 210044, China. saeedmughalssn@gmail.com

Muhammad Bilal

School of Computing and Communications, Lancaster University, Bailrigg, Lancaster LA1 4WA, United Kingdom. Email: m.bilal@ieee.org

Houbing Song

Security and Optimization for Networked Globe Laboratory (SONG Lab), Embry-Riddle Aeronautical University, Daytona Beach, FL, USA, Email: Houbing.Song@erau.edu

*Corresponding Author: Rashid Amin, Email: rashid4nw@gmail.com

Sentiment analysis is a process of dealing with people's opinions, remarks, and comments to extract valuable insights from them. Sentiment analysis can be used for various purposes like market analysis, campaign monitoring, decision-making, etc. In recent years, there has been much research on sentiment classification, particularly in English. However, these existing approaches used for the English language cannot be applied to the Urdu language. The substantial rise in communication traffic, including audio, text, video, and pictures, has significantly shifted the Internet of Things (IoT) from scalar to Multimedia Internet of Things (MIoT). So far, the integration of MIoT and

* Place the footnote text for the author (if applicable) here.

NLP systems has received less attention, but it has evolved as a novel research paradigm for smart applications. This article proposes deep learning techniques for sentence-level Urdu sentiment analysis (Urdu SA) for MIoT. Our approach consists of various phases, i.e., data gathering, text preprocessing, model training, testing, and evaluation. A data set of 25 thousand Urdu reviews are used for training the proposed models. This data set is built by scraping various Urdu blogs and social media platforms, and some part of the IMDB data set is used after translating it into the Urdu language. Native Urdu speakers do data annotation, and various preprocessing techniques, i.e., tokenization, stemming, etc., are applied. The two deep learning models, i.e., Convolutional Neural Network (CNN) and Long Short-term Memory (LSTM), are trained on preprocessed Urdu reviews to find their sentiments in this article. Both models are tested using various combinations of hyperparameters, and each model's accuracy and F1 scores are evaluated. The study results show that the LSTM model outperforms the CNN model by achieving a 96% accuracy and 91% F1 score.

**Additional Keywords and Phrases:** Urdu SA, Deep Learning, Data Analysis, Natural Language Processing

## 1 INTRODUCTION

Nowadays, the internet has become a popular platform for sharing information and selling products and services. In the era of information technology today, everyone now has access to the internet and can buy and sell his/her products or services around the globe [1]. Because of the internet, different social networking platforms like Facebook, Twitter, and various blogging websites play an essential role in expressing feelings and reviews. These reviews are critical as they reflect any service's status or ongoing trends about any matter [2]. This data can help producers of products services, or heads of any organization improve their product or service quality and decision-making. For example, in the days of the election, if the heads of any party want to know about his/her popularity status among people. People share their views about any party and leader on social media platforms. This information is crucial for the party leader to plan its next strategy for getting better results in elections [3].

The Multimedia Internet of Things (MIoT) is a collection of interfaces, procedures, and affiliated multimedia-related data interpretations. It is used in fully complying with apps and services in physical and virtual settings based on human-to-device and device-to-device interaction. Natural language processing (NLP) is the field of computational intelligence that deals with processing raw text into meaningful, full data. It has various applications, i.e., text summarization, sentiment analysis, machine translation, etc. Nowadays, the combination of MIoT and NLP is a trending research field. MIoT enables NLP applications to be more reliable, effective, and secure. Moreover, MIoT offers various techniques that can be blended with NLP models to enhance its effectiveness, such as implementing Chatbots in MIoT environments.

Opinion data is becoming increasingly essential from an application viewpoint to surmount and enhance the quality of their services and products exclusively in science and business intelligence systems based on MIoT. Sentiment analysis can be carried out on three primary levels [4]. These are sentence level, document level, and aspect level analysis. Document-level sentiment analysis involves evaluating the general sentiment of a complete document or text, offering a comprehensive viewpoint on the emotional tone. The research we conducted focuses on sentence-level sentiment analysis, which involves assessing individual sentences to gain a more detailed understanding of the stated sentiments inside particular chunks of text. Aspect-level sentiment analysis involves a more thorough examination of sentiments linked to particular characteristics or features stated in phrases or texts. The implementation of this multi-tiered methodology allows for a detailed examination of emotions in written information, accommodating a wide range of analytical requirements [5].

Urdu is a language that is widely used by more than 100 million people in the world. Many people share their tweets, reviews, and comments in Urdu [6]. So, the sentiment analysis for the Urdu language becomes a critical domain. Today, many techniques and methods are available for text mining and sentiment analysis for the English language [7]. However, significantly less work has been done for sentiment analysis in other languages like Urdu, Hindi, Arabic, etc. One main reason for fewer tools for the Urdu language is its complex morphological structure and different script than English. In addition, Urdu has diverse linguistic features that are contradictory to the English dialect. So, it is clear that Urdu sentiment analysis cannot be done by techniques used for English without modification [8,9].

Sentiment analysis can be done using various forms of textual data like Unigrams, lemmas, negation, etc., [10,11]." Sentiment classification is a process in which sentiments are divided or classified. Emotions can be divided into three categories: positive, negative, and unbiased or neutral [12]. During this step, the sentiment of each sentence is detected and classified into any group. In the end, we get the classified form of sentences into one of three different groups. When sentences are successfully classified, the results must be in such a format that anyone can easily understand them [13,14]. After completing the text classification process, the results are displayed on graphs to be easily analyzed.

The proposed article focuses on sentence-level Urdu sentiment analysis using a deep-learning approach for MIoT. Two deep neural networks (CNN, LSTM) are used to analyze the collected data set after annotation and preprocessing. This study gathered Urdu data from various blogs and parts of the IMDB review data set for building an Urdu data set. The IMDB data is translated into Urdu using Google Sheets [15]. After preprocessing the dataset, we train and test the neural network models, i.e., CNN and LSTM, on the training and testing data set. The primary objectives of this research are:

- To emphasize the need for sentence-level sentiment analysis of Urdu language for MIoT platforms using deep learning as these models are widely used for sentiment analysis of many languages like English, Chinese, Arabic, etc. Still, it is not used in Urdu because of its complex morphological structure.
- To build an annotated and large data set for Urdu SA by gathering user reviews from various Urdu blogs, and social media platforms and reviews from the IMDB data set translated into the Urdu language using Google Sheets.
- Implementing Deep Learning models, i.e., CNN and LSTM, on Urdu SA data to build an efficient and high-accuracy system for Urdu SA.
- Building a reliable system for decision-making based on Urdu reviews of users of any product or service.
- Comparison of previously proposed traditional machine learning models with deep learning models for Urdu SA to ensure the efficiency of deep learning models.

The remainder of the article is divided into eight Sections. Section 2 discusses the literature review, while Section 3 presents the introduction to the Urdu language and its morphology. Section 4 presents the proposed system architecture, while Section 5 of the article discusses experiments and results. Section 6 presents the discussion of the proposed problem and solution while Section 7 presents concluding remarks of the article.

## 2 RELATED WORK

The sentiment classification work done so far can be divided into three categories, i.e., Machine learning, Deep learning, and semantic orientation. Various researchers on Urdu text classification have done some work using semantic orientation and machine learning techniques [16, 17]. Rehman et al. [2] presented a novel framework for Urdu SA based on Urdu comments. Their study uses various websites to collect Urdu comments for sentiment analysis. They use polarities of lexicons of Urdu words for this purpose. In their works, 7335 lexicons are used, two thousand six hundred seven lexicons are negative, while 4728 lexicons are positive. They analyze the proposed model on 124 Urdu comments that are classified subjectively by the model. In the study's first phase, numerous tokens are created for the input sentences. Finally, to determine the polarity of the words, those tokens are compared to Urdu language lexicons. Total polarity is the sum of all words or lexicons' polarity. The proposed model got 66% accuracy and a 0.73 F-measure value. Various challenges to Urdu SA are also discussed in this study.

Raheela et al. [18] proposed a supervised learning approach for Urdu SA. They used a decision tree algorithm for this purpose. Their framework is composed of two main steps. In the first step, data is preprocessed, and unnecessary words are removed to reduce data complexity. Stop words, hashtags, and other unnecessary words are removed in this step. In the next phase, feature vectors are generated. For this purpose, positive and negative comments and POS tags are identified. Finally, the decision tree algorithm is applied to the data for sentiment classification. The Urdu tweets are used in this study as a data set. The proposed decision tree model got 90% accuracy. Mukhtar et al. [1] also present an article on the Urdu SA using supervised ML techniques. They gathered data from various blogs and preprocessed it. After preprocessing, supervised ML algorithms, i.e., Support Vector Machine (SVM), Decision tree, and K-Nearest Neighbor (KNN), are used for sentiment classification. After comparing the performance of these models, they cannot get satisfactory results. Then, feature extraction is analyzed, and 152 features are extracted.

Then, after the classifiers' training, they got a 67% accuracy level using the KNN algorithm. So, in this study, KNN outperforms SVM and decision trees in sentiment classification of the Urdu language. The first step of this machine learning process is data gathering, where data is collected from different sources for analysis purposes. Then, text preprocessing is performed in the second phase, where various steps are performed to remove unwanted and unnecessary words from the gathered data. After this process, features are extracted, and machine learning models are trained on extracted features. When a model is successfully trained on the training data, model testing is performed to validate the model. If model performance is low in this phase, then training is done again to optimize the performance. At the end of the testing, the model can predict the sentiment of any given data. They used the LSTM model for their data roman Urdu data set. This study performs various experiments using varied hypermeters for the best results. The study results show that the LSTM model got satisfactory results by scoring 95% accuracy on the validation data set. Figure 1 shows the machine learning process of sentiment analysis.

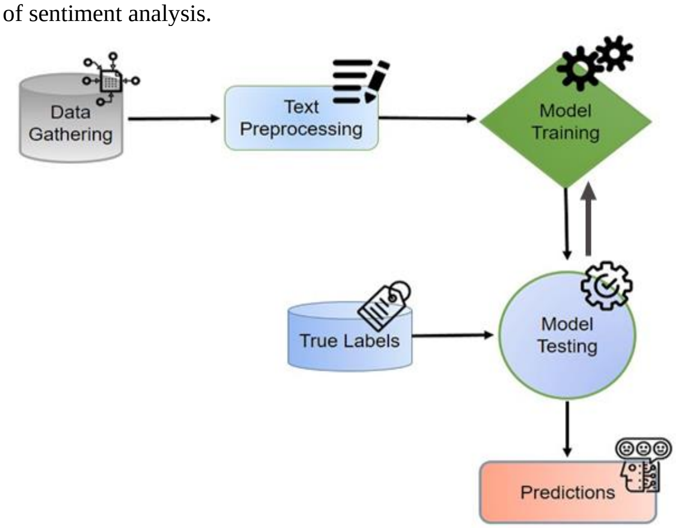


Figure 1 Process of Machine learning for sentiment analysis

Syed et al. [5] present an article on sentiment analysis using sentiUnits. Shallow parsing is used in this article to identify SentiUnits from data. SentiUnits are phrases that provide details about the emotion contained in the sentence. A Lexicon of Urdu language is made in this article and then used for sentiment classification. This paper highlights the linguistic, i.e., grammar and morphology of the Urdu language and technical aspects of this problem. The evaluation of the system is done with various test text data and got satisfactory results. This article can be used as a baseline for lexicon-based sentiment analysis.

A novel lexical normalization technique for Roman Hindi and Urdu sentiment analysis is presented by Mehmood et al. [7]. They introduce a novel approach called Transliteration-based Encoding for Text normalization. After normalizing the data using their novel technique, they performed text classification. They used various machine learning models as baseline models and performed analysis on Roman, Hindi, and Urdu data. After the evaluation, their proposed technique outperforms traditional machine learning models. Figure 2 illustrates the basic process of the lexicon-based approach for sentiment analysis.

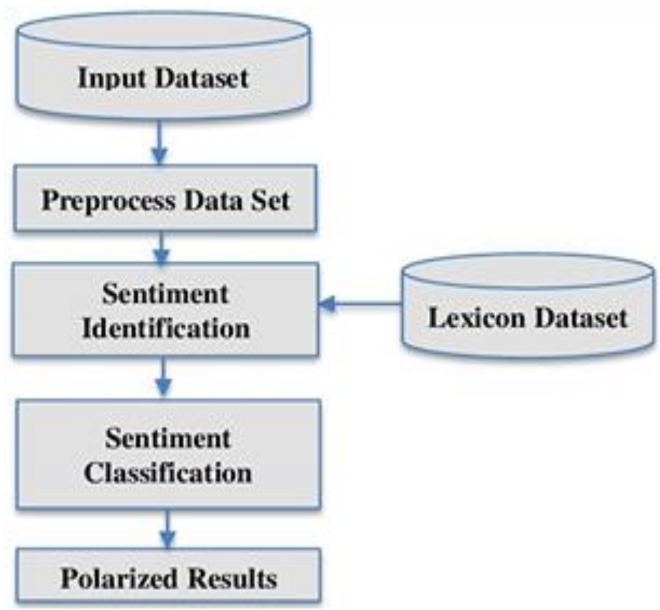


Figure 2 Process of lexicon-based sentiment analysis

As illustrated in Figure 2 input data set is preprocessed for removing stop words, URLs, numbers, etc., in the initial phase of this technique. After that, sentiment identification is made by using the previously created lexicon data set. After identifying sentiments from sentences, sentence classification is done, and output is produced. Mukund et al. [19] presented research on sentiment by using Urdu blog data. This research uses structural correspondence learning to transfer Urdu sentiment data from Urdu news wire to Urdu blog data. These two platforms' data are not trivial as newswire data is written in Latin script and exhibits code-mixing and code-switching behavior. So, for making pivots, two oracles are used in this study. The first is the Transliteration oracle, and the second one is called the translation oracle. The transliteration oracle is for script variation, while the translation is used for code-switching and codemixing behavior. In [20], they introduce a new part-of-speech tagging process that helps them identify words based on POS categories, representing code-mixing behavior. Finally, they evaluate their model against a supervised learning model and compare results based on various performance measures. After evaluation, they got a 59.4% precision value and a 62.4% recall value for the proposed technique.

Altaf et al. [21] present a sentiment analysis study for the Urdu language that exploits linguistic features. A dataset was developed containing idioms, proverbs, and news sentences. After linguistic analysis of Urdu, part-of-speech tags, boolean and numeric features were extracted without relying on n-grams. This approach focuses on contextual information and classifies idioms/proverbs which can have strong sentiments without sentiment words. Classifiers like J48 were tested on this feature set, achieving 90% accuracy and 88% F-1 score for sentence-level sentiment classification. Ahmed et al. [22] present a contextually enriched meta-learning ensemble (MLE) model for Urdu sentiment analysis. The paper proposes a new two-tiered meta-learning ensemble technique that combines predictions from basic ML and deep learning models on separate levels. It compares the performance of deep baseline classifiers to the ensemble model. Findings are expanded by contrasting the proposed ensemble approach against other advanced ensemble techniques. The MLE approach reduces complexity and overfitting. Results show classification accuracy of baseline deep models is significantly improved by the proposed MLE approach.

Muhammad and Burney [23] conducted research on Urdu sentiment analysis using machine and deep learning techniques. The data is preprocessed, features are extracted, and machine/deep learning algorithms are applied for binary classification. Results from different techniques are compared to identify innovations. Hybrid methods are also proposed. The study aims to advance sentiment analysis for the widely spoken yet under-researched. Khan et al. [25] introduce a new multi-class Urdu sentiment analysis dataset of 9312 user reviews from various domains manually annotated as positive, negative or neutral. Baseline results are established using rule-based, machine-learning and deep learning techniques with different text representations. Additionally, Multilingual BERT (mBERT) is fine-tuned for Urdu sentiment analysis. Models are trained and evaluated on two datasets. Results showed the proposed mBERT model using BERT embeddings outperformed other classifiers, achieving the highest F1 score of 81.49%.

Mukhtar et al. [24] proposed a lexicon-based technique for handling negations in Urdu text to improve Urdu SA's accuracy. They used 6025 Urdu sentences for this purpose. Firstly, morphological negations are included in the negative text document of the Urdu Sentiment Lexicon, which was created to conduct Sentiment Classification on Urdu blogs. Secondly, implicit and explicit negations are handled using a rule-based method. Rules are designed to effectively deal with both implicit and explicit rebuttals in this study. The proposed method enhanced the accuracy of Urdu sentiment analyzers from 73.88% to 78.32 % in this article. Various machine-learning models are trained on specific data provided by the system [26]. The data is gathered from multiple websites of hotels and preserved in a database [27]. The original corpus is used to create the testing and training datasets. After that, different machine learning classifiers are used to detect the polarities of Roman Urdu text. They used various metrics to assess and compare the results, i.e., accuracy, precision, recall, and the F1 measure. As a result, their SVM model achieves an accuracy rate of 85 percent on validation data. Amjad et al. [29] used Twitter news data for Urdu SA. They emphasized sentiment analysis of news tweets in the Urdu language from Pakistan's leading media sources. They created an Urdu sentiment lexicon by collecting Twitter data over ten months. Furthermore, based on the cumulative sentiment score of the text, they construct an algorithm that divides Urdu content into positive, negative, or neutral classes. The accuracy of their sentiment analysis system is 77%.

Hashim et al. [30] proposed a sentence-level Urdu SA method using nouns. They used Urdu news data for their lexicon-based method for Urdu SA. Urdu nouns are used for the detection of sentiments in the sentences. Their proposed technique got 86.8 % accuracy and testing data set. Table 1 compares and summarizes the previously proposed methods for Urdu SA.

Table 1: Comparison of related work

| Author | Data set | Machine Learning | Deep Learning | Lexicon | Accuracy |
|---|---|---|---|---|---|
| **[2]** | 7335 Urdu Comments | ● | ● | | 66% |
| **[18]** | Urdu com-Ments | | ● | ● | 90% |
| **[1]** | Urdu Blog Data | | ● | ● | 67% |
| **[19]** | Roman Urdu Data | | | ● | 95% |
| **[5]** | Urdu sentiment data | ● | ● | | 78% |
| **[24]** | Urdu Blog Data | | ● | | 62% |
| **[25]** | Urdu Tweets | | ● | | 86% |
| **[28]** | Urdu sentiment data | ● | ● | | 85%. |
| **[30]** | Urdu blogs Data | ● | ● | | 82%. |
| **[21]** | Roman Urdu Hotel Reviews | | ● | ● | 85% |
| **[22]** | Urdu news tweets | ● | ● | | 77% |
| **[23]** | Urdu News | ● | ● | | 86% |

## 3 DATA SET COMPARISON

Various researchers built data sets for Urdu SA using different platforms. Most of the data sets are built for lexicon-based Urdu SA approaches. The data set used in this study is a combination of the IMDB standard data set and Urdu blogs data gathered during the study. Various Urdu blogs are scraped for user reviews and saved in a .csv file. The IMDB data set was then translated into Urdu to enhance the data set's size. The IMDB movie reviews data set was downloaded from Kaggle, and native Urdu speakers annotated both data sets. The related studies used smaller data sets than the proposed study for Urdu SA. Some related studies that use Urdu data are discussed below:

Mukhtar et al. [1] used machine learning techniques for the sentiment analysis of Urdu data. They used 6025 sentences that were collected from online Urdu blogs. Rehman et al. [2] used 7335 Urdu Sentiment words for their lexicon-based method of Urdu SA. Their data set contains 2607 positive and 4728 negative words. Khalid et al. [14] used 1200 news documents for Urdu SA using fast text embedding. Their data set contains 10841 unique sentences of Urdu. Raheela et al. [18] used 600 Urdu tweets for their Decision tree model for Urdu SA. Nasim et al. [32] used 6816 Urdu tweets for their proposed Markov chains model. Figure 3 presents the comparison of data sets of the previous and proposed study.

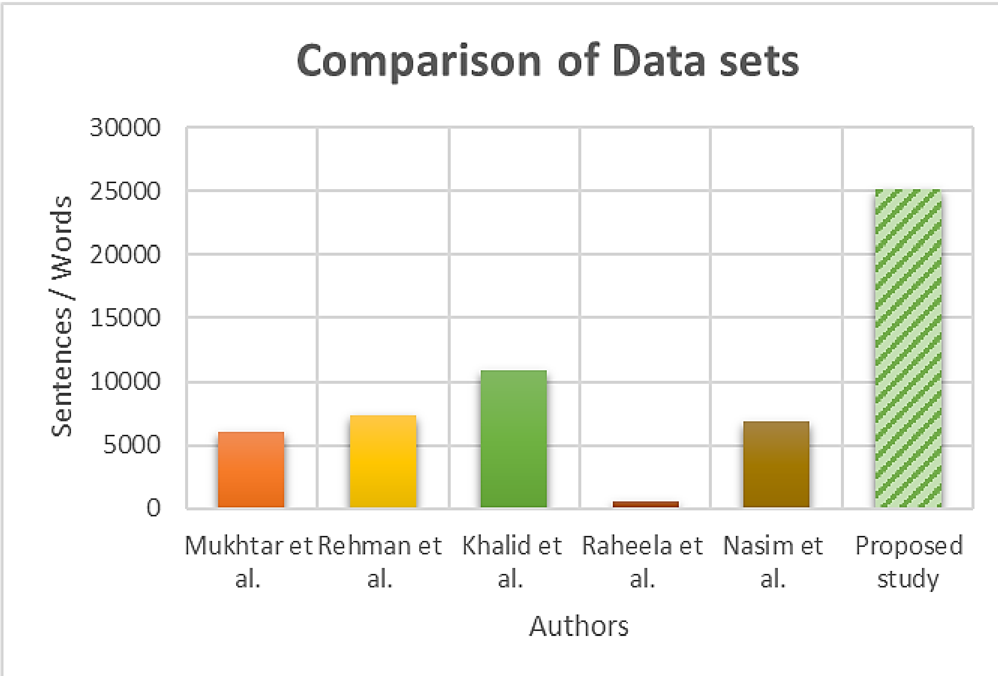


Figure 3 Comparison of Data set

In 2018, India had around 62 million Urdu speakers, mainly in Jammu, Kashmir, and Maharashtra. In 2015, about 752,000 Urdu speakers in Nepal, and approximately 450,000 Urdu in Bangladesh in 2018 [29]. Other countries with Urdu speakers include South Africa, Mauritius, and Guyana. Urdu is Pakistan's national language, which is closely related to the Hindi language. While Urdu has more Persian and Arabic vocabulary than Hindi, and Hindi has more Sanskrit vocabulary. Since the 12th century, Urdu has been written in a Perso-Arabic script in the Nastaliq format. Urdu is a Turkish term that means" foreign" or "horde."Figure 4 presents the alphabet of the Urdu language.

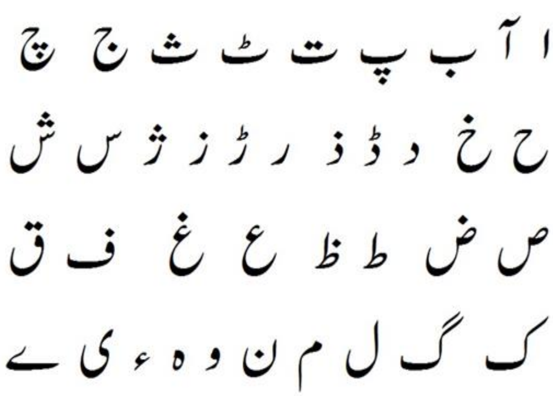


Figure 4 Alphabets of Urdu Language

The Urdu alphabet, referred to as the "Perso-Arabic script," comprises 38 characters as illustrated in Figure 4. which include consonants and vowels. The Urdu alphabet consists of individual letters, each representing a distinct phonetic sound. Additionally, the script is written in a right-to-left direction. The Urdu script adeptly conveys the many intricacies of the language, mirroring its abundant linguistic heritage. The alphabets, adorned with elegant curves and loops, contribute to the distinctive visual identity of Urdu.

## 4 PROPOSED SYSTEM ARCHITECTURE

This Section presents the proposed system and process of the study, i.e., data collection details, preprocessing, and model training. First, data preprocessing is performed to remove unwanted words and symbols. Then cleaned data is prepared for the training of machine learning models. We trained two deep learning models, i.e., CNN and LSTM. After model building, both models are compared on standard evaluation measures.

### 4.1 Data gathering

There are not as many standard data sets for the Urdu language. Very few data sets are available for the concerned language, but they are very small. So the data is gathered from various blogging and social media websites. These websites include Daily Pakistan, hamariweb.com, BBC Urdu, and many other Urdu blogs. Some part of the standard IMDB reviews data set [30] is also translated into Urdu to build a larger data set for the training and testing of the proposed models. The websites from where data is collected is shown in Table 2. The data collection from these websites is done using the "Beautiful soup" Python library and stored in a .csv file. The final data set contains more than 25000 rows and two columns. The classification of the data set is shown in Figure 5.

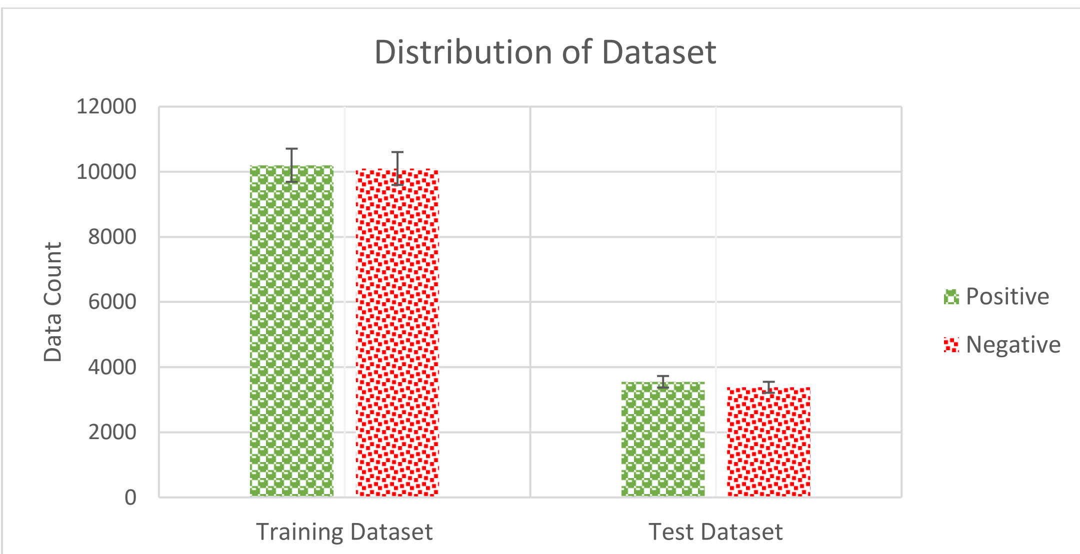


Figure 5 Classification of Data Set

Figure 5 illustrates the data distribution utilized in our study. The diagram depicts the distribution of data during the training, validation, and testing stages. Specifically, 70% of the dataset is used for training the proposed models, enabling the creation of robust sentiment analysis algorithms. In addition, 10% of the dataset is set aside for the validation phase, which ensures that the model's performance is optimized as it is being trained. The 20% of the data is used for the testing phase, enabling a thorough assessment of the models' ability to generalize. The dataset is divided into two columns: the first column represents individual reviews, while the second column contains the sentiments expressed in the text. The dataset consists of around 10,100 rows of negative reviews and 10,200 rows of positive reviews. This balanced representation of sentiments ensures a strong foundation for training and evaluating predictive models. The test dataset consists of 3380 reviews classified as negative and 3550 reviews classified as positive. Table 2 presents the sources and collected data.

Table 2: Sources of data collection

| **Sources** | **Reviews** |
|---|---|
| www.bbc.com/urdu, www.urdunews.com, www.siasat.pk | 1436 |
| www.blog.urdumaza.com | 1932 |
| www.urdupod101.com/urdu-vocabulary-lists | 2307 |
| www.facebook.com, www.urdughr.com/2020/03/best-quotes-in-urdu.html , https://www.itdarasgah.com/ | 2958 |
| https://poetryinurdu.pk/ , www.youtube.com | 1348 |
| IMDB Review data set (Converted into Urdu) | 15225 |

Table 2 presents an in-depth overview of the numerous sources from which our dataset was collected, together with the corresponding quantity of reviews gathered from each source. The sources include a range of platforms, demonstrating the abundance and variety of Urdu information available on the internet. The varied sources comprise credible news outlets such as www.bbc.com/urdu , www.urdunews.com, and www.siasat.pk collectively contain 1436 reviews. The dataset is enhanced with 2932 reviews from blogs accessible at www.blog.urdumaza.com, www.urdumanzil.com, and www.urdudaan.blogspot.com. The websites www.hamriweb.com www.sachiidosti.com/forum/ www.wifaqulmadaris.org and www.urdupod101.com/urdu-vocabulary-lists offer a total of 2307 reviews on forums, educational platforms, and vocabulary lists. Platforms such as social media and curated content websites, specifically www.facebook.com,www.urdughr.com/2020/03/best-quotes-in-urdu.html and https://www.itdarasgah.com/ together yield a total of 2958 reviews. Platforms that present creative content, such as https://poetryinurdu.pk/ and www.youtube.com, include a total of 1348 reviews. In addition, we included 15225 reviews from the IMDB dataset that were translated into Urdu.

*4.1.1 Data Annotation.*

In this Section, we manually annotate the data set for adding values to our data set. Three native Urdu speakers annotate all the data. Annotators classified sentences into three categories, i.e., positive, negative, and neutral. The sentences are classified based on the number of positive, negative, and neutral words contained in the sentences. A few rows of the training data set are shown in Figure 6.

| review | sentiment |
| --- | --- |
| میں نے دو وجوہات کی بنا پر یہ ڈی وی ڈی کرایہ پ... | negative |
| انگمار برگ مین کی اسکمین دیکھنے کے بعد ، مجھے ... | positive |
| خوفناک نفسیاتی تھرلر جس کو دیکھنے کے لئے تقریب... | negative |
| مجھے اسٹیون سیگل پسند ہے لیکن میرے پاس کوئی فل... | negative |
| ، یہ کارٹی کی لامتناہی لائن میں شامل ہوجاتا ہے... | negative |
| جوجن بیک نے اس معنی پر مبنی فلم کی اس چھوٹی سی... | negative |
| مجھے واقعی یہ شو پسند ہے۔ یہی وجہ ہے کہ مجھے ح... | positive |

Figure 6 Some rows of training data

Neutral sentences are usually unbiased sentences [31]. Therefore, the two annotators shall label a sentence in the case of an understanding on the same label. In the case of a conflict, i.e., if both annotators give different labels to the same statement, a third annotator of expertise is used. Based on the third annotator's judgment, sentences are labeled positive, negative, or neutral if he agrees with the first or second annotator. However, if the third annotator cannot label the sentence, that sentence is discarded. Inter-annotator compatibility is measured using the [32, 33]. In addition, the Kappa statistic is used to assess inter-rater reliability [13]. Interrater reliability calculates the level to which different raters allocate a similar score to the same attribute. Figure 7 illustrates the proposed system of Urdu SA.

As illustrated in Figure 7, the data gathering is performed in the first step of the study. In the second phase, data analysis is done on the gathered data. In the data analysis phase, the data is translated and annotated for performing data preprocessing. Preprocessing is done after data analysis, where stop words removal, tokenization, stemming, etc., is done for training the deep learning models. Then, deep learning models are trained, validated, and analyzed. At the end of the study, outputs are visualized for better understanding.

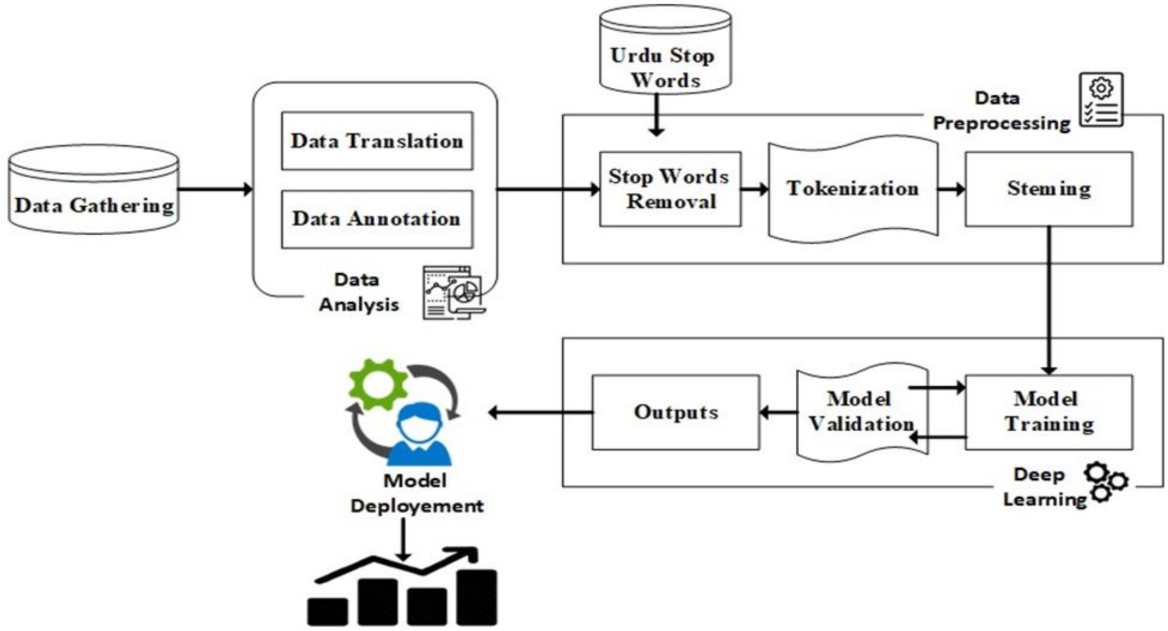


Figure 7 Proposed Methodology

We utilized Python as our main programming language, incorporating key modules like Pandas for effective data management and NumPy for numerical computations. We used other libraries such as NLTK for text preparation, TensorFlow and Keras for the implementation of our deep learning models, specifically the Convolutional Neural Network (CNN) and Long Short-term Memory (LSTM). The Urdu Hack library was utilized to address the complicated structure of Urdu language, while XLM-RoBERTa was employed to leverage pre-trained language representations for enhanced model performance. In addition, we used Matplotlib and Seaborn libraries to create visual representations of the data.

#### *4.1.2 Data Preprocessing*

Data preprocessing is a method of converting unstructured or raw data into a valuable and standard form. The data can have several insignificant and useless parts for the creation of results. Data cleaning techniques transform raw data into suitable formats for the machine learning process [34]. After data gathering, data preprocessing is performed using the Urdu hack Python library in this study. Urdu hack [35] is an Urdu NLP library with various features for Urdu language processing. Stop words are the most common words used in any language to complete sentences. Those words are removed from the annotated data set. That is because Urdu stop words play a significant role in the completeness of sentences [36]. The removal of stop words is based on the assumption that the absence of those words reduces the feature space that helps produce precise results. A list of Urdu language stop words to remove stop words from the data set is shown in Table 3. Then hashtags and URLs are removed. This process is done because these terms cannot affect the sentiment expressed in the sentence. Also, the data size becomes less, which plays a significant role in model training.

Table 3: Stop words

| **English** | **Urdu** |
|---|---|

| | |
|---|---|
| Your | تمہارا |
| June | جون |
| Can | سکتا |
| Front | سامنے |
| Few | کچھ |
| If | اگر |

In Urdu, words in the data set are separated based on white spaces and punctuation. There are also some non-Urdu words in the data set and multiple words without any white space or punctuation marks. Tokenization is referred to as the division of a text into its smaller parts, i.e., tokens. For this purpose, the Urdu Hack Python library is used [37]. The tokens of the data are generated and assigned with a unique value. A list of some generated tokens is shown in Figure 8.


'طویل': 593,  'چلتا': 595, 97, ': 598, 'آتی': 599:
'درخواست': 602, 'سورج': 603, 'فری': 604, 'سوال':
605, 'شادی': 606,, 'ایوارڈ': 608, 'پیاری': 609,
'لانیو': 610, ' , 'دفتر': 612, 'کتاب': 613, ', 'ملنے':
616, 'خود': 617, ', 'الفاظ': 619, 'سچ': 620, 'آہ':
621, 'ٹن': 622, ', 'مسکرا': 626, 'لڑکے': 627,
'روکنے': 628, 'برا': 629, 'فاتح': 630, 'بوسہ': ,
'مطابق': 633,  , 'زبردست': 635, 'دونگا': 636,
'مصحف': 637, 'بولدار': 638, 'منتقل':

Figure 8 Some tokenized words from data set

The method of reducing a word towards its root is known as stemming. It frequently eliminates derivational affixes [38]. To merge several similar word forms, 'affix elimination' based stemming is used. A list of affixes comprising 425 prefixes and 75 suffixes from the Urdu language is used for this task, which decreased the data set by 26%. After steaming, sentiment words can be easily detected from the data set. Table 4 shows the stemmed words.

Table 4: Stemmed words

| S.No | Suffixes | Words | Words | Prefixes |
|---|---|---|---|---|
| 1 | مند | غیرت | لائق | نا |
| 2 | ی | افطار | گمان | بد |

| | | | | |
|---|---|---|---|---|
| 3 | کار | ادا | امن | پر |
| 4 | خور | گوشت | وضو | با |
| 5 | ور | نام | کار | بد |

- Separately choose the token's first and last characters and look for all of them in the affixes list. If no results are found, merge the second and second final characters and check again.
- Continue the process till an affix is located inside the list, then scan the 'tokenization list' for the rest of the expression. Hold it and discard the remainder of the token (prefix or suffix) if found.
- If prefixes and suffix strings are not found, take the original term as it is.

*4.1.3 Data Representation*

At first, all reviews consist of variable lengths of words. Each data set review is converted into a specific length of words in this study phase. For this purpose, we use the XML-Roberta model. In this process, each review is represented by a 2D vector of dimension n x d., where "n" is the number of words in each review and "d" is the dimension of word vectors. The Zero padding technique ensures the same size of all reviews. Using this process, each review size is "n' x d," where n'=60 and d=300. Next, the words from reviews are converted to corresponding vectors. Some words in the reviews did not have word embeddings. So, for the generation of missing embedding's we use the FastText library. All three grams of the unknown term are collected in the word embedding model and looked up one by one. After that average of identified embedding's is calculated and used in place of missing ones.

### 4.2 Model Training

This section discusses the proposed model's architecture and training process. We have used two deep learning models, i.e., CNN and LSTM, for the problem statement.

*4.2.1 CNN Model*

Convolution neural network CNNs are widely used for classification problems because of their high performance. These models can extract essential features from data that enhance the model's accuracy in classification problems [39]. In the study's second phase, a (CNN) is built to classify Urdu reviews. The CNN model's input is word vectors mapped to a matrix of size "n x d." Here "n" is used for the number of review words while d is used for the dimension of embedding space (d=300). The proposed model uses a specific filter size "f" and the number of filters "m." The size of the feature maps produced by each vector is "n – f + 1", where "n" represents the total number of words in the review. Then max-pooling is applied on obtained feature maps, due to which the model extracts the essential n-grams effectively in the embedding space. Then a fully connected hidden layer having ReLU as an activation function is used. A 50% dropout probability dropout layer is used in the training phase to regularize the network and prevent overfitting. Finally, a Softmax layer with two output units is used in the proposed system to predict the results, as shown in Figure 9. The Pseudo-code for the proposed CNN model is given in Algorithm 1.

ALGORITHM 1: CNN model for Urdu SA

Procedure: CNN

```
Class CNN [features, UrduSA, Reviews]
Vectorizer ← Transform (Reviews)
UrduSA ← Analysis (Reviews)
X_train, X_test, train_y, test_y ← train_test_split(data['review'],data['sentiment'], test size=0.3)
Int MAX_SEQUENCE_LENGTH=300
x_train = pad_sequences(sequences, maxlen=MAX_SEQUENCE_LENGTH)

x_test = pad_sequences(sequences_test, maxlen=MAX_SEQUENCE_LENGTH)
def model:
#initialize the model

Model1 ← sequential ()
Compile ← model. Compile ('sparse_categorical_crossentropy', 'adam', metrics=['accuracy'])
#Fitting the model
Fit Model ← model1.fit (x_train, y_train, batch_size=batch, epochs=25,
Validation_data= [x_test,y_test], verbose=1)

#Prediction
Predict ← model1.predict_classes(X_test, verbose=1)
END Class

def analysis:
analysis= model1(Reviews)
If analysis.sentiment.value>0:
return 1
elif analysis. sentiment.value==0:
Return 0
else:
return -1
```

---

Algorithm 1 describes the pseudo-code for the CNN model for Urdu SA. After vectorization, the data is divided into four parts, i.e., (X_train, X_test, train_y, test_y) using the train test split function of the sklearn python library. The maximum sequence length is limited to 300, and the model is compiled using the loss function as "'sparse_categorical_crossentropy,'" optimizer as "adam" with an accuracy metric. Then, the model is fitted on "x_train and y_train data and validated on x_test and y_test data set. In the end, the predictions are made by the model based on given data as 1 (Negative), or 0 (Positive), or -1 in case of incompatible data.

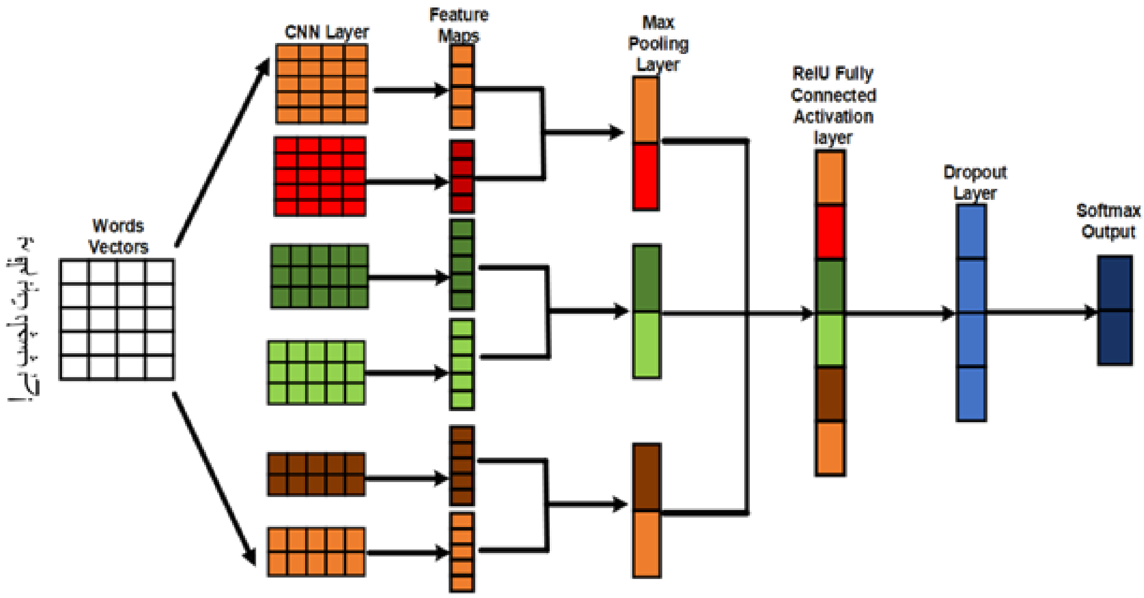


Figure 9 Proposed CNN architecture for Urdu sentiment analysis

*4.2.2 LSTM Model*

Long short term memory (LSTM) is a recurrent neural network (RNN) used to learn long-term dependencies. The RNN computes the next hidden state based on the current word embedding and the previous hidden state [40]. The hidden state of RNN can be described by ht Ɛ Wm, Where m denotes the dimension of the RNN model. In our case, m=300 at "t" time can be computed using the below equation (1).

$$ht=f(Wh*xt+Uh*ht-1+bh) \qquad (1)$$

where f(x) represents a non-linear function, i.e., tanh, "xt" represents the current word embedding, "Wh" and "Uh" are used for weight matrixes. The "ht-1" is used for the previous hidden state. While "bh" denotes the biased term. Unfortunately, this simple RNN has a disadvantage: it suffers from a problem named the "vanishing gradient problem" and cannot handle long-term dependencies in the backpropagation training stage. LSTMs are intended to prevent the long-term dependence issue that traditional RNNs cannot handle. The following equations 2-6 compute the hidden state of an LSTM unit:

$$ft = \sigma(Wf*xt+Uf*ht-1 + bf) \qquad (2)$$

$$it = \sigma (Wi - xt + Ui * ht-1 + bi)\, bf \qquad (3)$$

$$ot = \sigma(Wo*xt + Uo * ht-1+ bo) \qquad (4)$$

$$ct = ft \circ ct-1 + it \circ \tanh (Wc*xt+Uc*ht-1+bc) \qquad (5)$$

$$ht=ot \circ \tanh (ct) \qquad (6)$$

where it represents the input gate, "ft" represents the forget gate, and ct represents the cell state. The ht is used for the hidden state, σ denotes the sigmoid function, and ◦ is the Hadamard product. The LSTM's key drawback is that it does not sufficiently allow for post-word knowledge since the statement can only be interpreted in one way, i.e., forward. The Pseudo-code for the proposed LSTM model is given in Algorithm 2.

ALGORITHM 2: Proposed LSTM model for Urud SA

```
Procedure: BiLSTM
Class BiLSTM [features, UrSA, Reviews]
Vectorizer ←———— Transform (Reviews)
Ursa ←———— Analysis (Reviews)
X_train, X_test, train_y, test_y ←———— train_test_split(data['review'],data['sentiment'],test size=0.3)
Int MAX_SEQUENCE_LENGTH=300
x_train=pad_sequences(sequences, maxlen=MAX_SEQUENCE_LENGTH)

x_test=pad_sequences(sequences_test, maxlen=MAX_SEQUENCE_LENGTH)
def model:
#initialize the model

Model ←———— sequential ()

Compile←—— model.Compile('sparse_categorical_crossentropy','adam', metrics=['accuracy'])

Fit Model ←———— model.fit(x_train,y_train, batch_size=batch, epochs=25,
Validation_data= [x_test,y_test], verbose=1)
Prediction
model.predict(x_test)
END Class

def analysis:
analysis= Model(Reviews)
If analysis.sentiment.value>0:
return 1
elif:
analysis.sentiment.value==0:
Return 0
else:
return -1
```

Algorithm 2 explains the pseudo-code for the LSTM model for Urdu SA. The first phase preprocesses and vectorizes the data for model training. Then LSTM model is defined and complied using the loss function as "'sparse_categorical_crossentropy,'" optimizer as "adam" with accuracy metric. After that model is trained on "x_train and y_train data and validated on x_test and y_test data set. On successful training and validation of the model, it predicts the sentiment of given data as 1 (Negative), or 0 (Positive), or -1 in case of incompatible data. We employ a bidirectional LSTM, consisting of two LSTMs with their outputs connected to solve this problem. The sentence is read forward by one LSTM and backward by the other. This way, reviews are passed through every LSTM with hidden size "h." After processing, the output is summed up to make a vector of size 2h. This output vector is given to a fully connected layer that uses RELU as an activation function. A dropout layer is added after this layer to prevent overfitting in the network, as shown in Figure 10. Finally, an output layer with a softmax activation function is used for classification output.

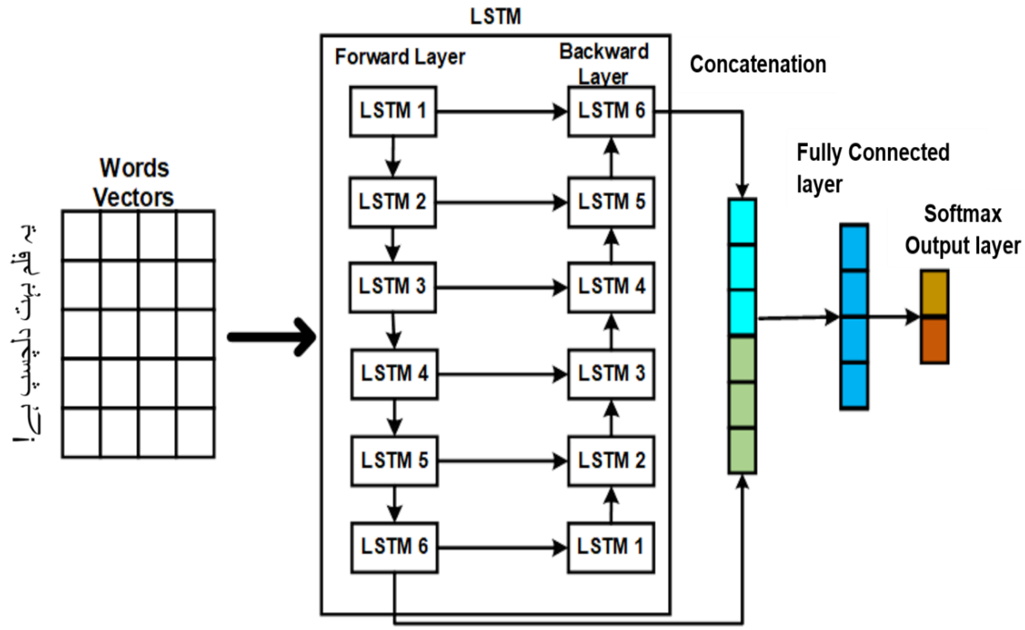


Figure 10 Proposed LSTM architecture for Urdu sentiment analysis

We implemented a thorough strategy for improving the model, which involved both fine-tuning and precise selection of hyperparameters. Following the initial training phase, we performed fine-tuning to optimize the models further. This included iteratively fine-tuning certain parameters to optimize the model's performance on our particular Urdu sentiment analysis task. The hyperparameter selection process was carried out methodically, examining multiple combinations to get the optimal set for both the Convolutional Neural Network (CNN) and Long Short-term Memory (LSTM) models. During this optimization process, we considered factors such as learning rates, batch sizes, and layer configurations during this optimization process The final settings were selected by doing thorough experimentation to attain optimal performance measures, such as accuracy and F1 scores.

## 5 EXPERIMENTS AND RESULTS

In this Section, we provide the results of conducted experiments. In the study's experimental phase, we used a 10-fold cross-validation technique using a balanced data set containing 7 thousand reviews. 85% of data is used for model building in every fold, while 15% is used for validation purposes. The word embedding for our task is set to 300 dimensions. A fully connected dropout layer is used to avoid overfitting. The cross-entropy function is used as a loss function for the proposed models. We used the most common performance metrics, i.e., accuracy and F1-measure, to assess the classification performance of proposed models. Accuracy is the most used measure for machine and deep learning model classification performance, and it can be calculated using equation 7:

$$Accuracy=(TP+TN)/(TP+TN+FP+FN) \quad (7)$$

Where "TP" denotes true positive, and "TN" is used for the true negative. True positive is the measure that belongs to any class "X" and also predicted correctly by the model as the member of class "X." True negative is the number of data points that do not correspond to class "X" and are not predicted to belong to class "X" by the model. False Positive is the number of reviews whose actual labels do not correspond to class "X" but are predicted to belong to class "X" by the classifier. A false negative is the number of reviews whose actual labels belong to class "X" but are predicted by the model to be part of other classes. The harmonic mean of precision and recall is known as the F-measure, and it can be computed by equation 8.

$$F\text{-}measure=2\times\ (Precision\times Recall)/(Precision+Recall) \quad (8)$$

Precision is a metric used to assess the exactness of any classifier's output, while recall determines the completeness of the classification model. For the CNN model, Various experiments are performed by varying hyper-parameters. The default configuration used for the CNN model is shown in Table 5. The best CNN model configuration is performed on the test data using a fully connected layer of 100 sizes. A fully connected layer of size 100 got the best results with other parameters like the default configuration. The CNN model got the highest accuracy of 88% and 86% F1-score during the testing phase of the study.

Table 5: CNN configurations

| **Hyper-Parameter** | **Value** |
|---|---|
| Number of neurons in fully connected layer | 40 |
| Number of filters | 300 |
| Dropout rate | 0.5 |
| Learning rate | 0.01 |
| Number of epochs | 25 |
| Batch size | 40 |

The LSTM configuration used in the proposed study is listed in Table 6. We measured the accuracy and F1-score after varying each hyper-parameter value separately. The optimum results of the LSTM model are obtained by using a dropout rate of 0.2 while leaving the other parameters at their default settings.

Table 6: LSTM configurations

| Hyper-Parameter | Value |
|---|---|
| Hidden state dimensions | 300 |
| Number of neurons in fully connected layers | 40 |
| Dropout rate | 0.5 |
| Learning rate | 0.001 |
| Number of epochs | 25 |
| Batch size | 40 |

The proposed LSTM model achieves 96% and 91% F1 score and accuracy using the configuration mentioned above. The accuracy and F1 scores of the proposed models are illustrated in Figure 10. As from Figure 11, the LSTM model outperforms the CNN model by achieving higher accuracy.

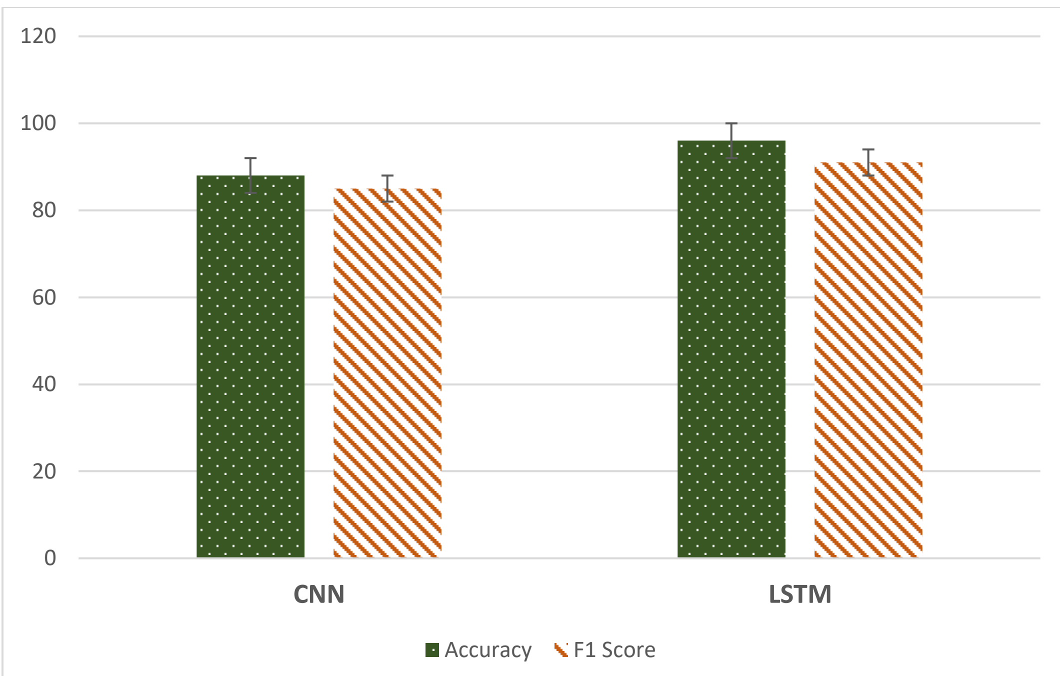


Figure 11 Accuracy and F1 score of proposed Models

The CNN model achieves a handsome accuracy level in the training phase, while the highest accuracy of this model is 88% on the validation data set. The LSTM model outperforms the CNN model in both the training and validation phases. The accuracy of the CNN model during the training and Validation phase is illustrated in Figure 12.

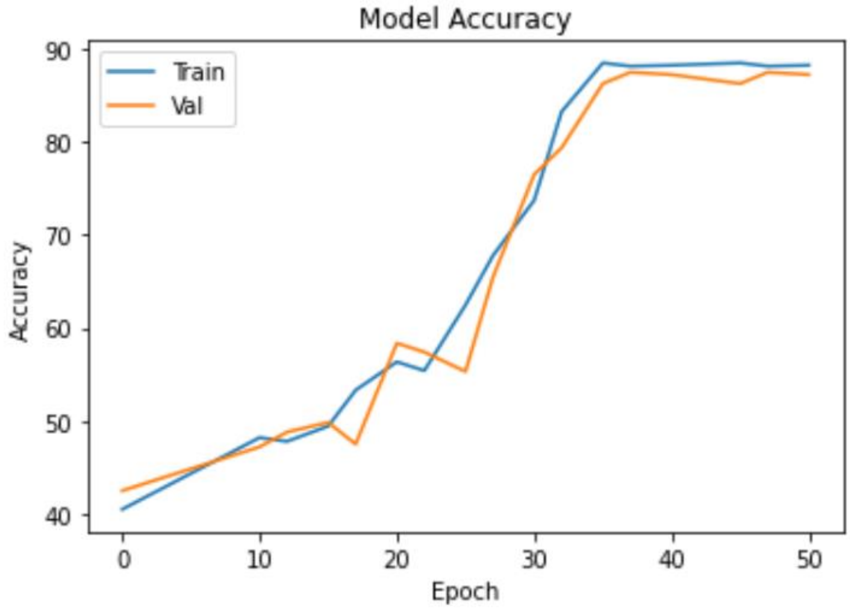


Figure 12 CNN training and validation accuracy

As illustrated in Figure 12, the CNN model performance is low at the start. However, as epochs increase, the validation accuracy becomes higher than training; in the end, it got 88% validation and 87% training accuracy. The LSTM model performs better than CNN because it deals with long-term dependencies and has higher accuracy levels in the training and validation phases. Figure 13 shows the training and validation accuracy of the LSTM model.

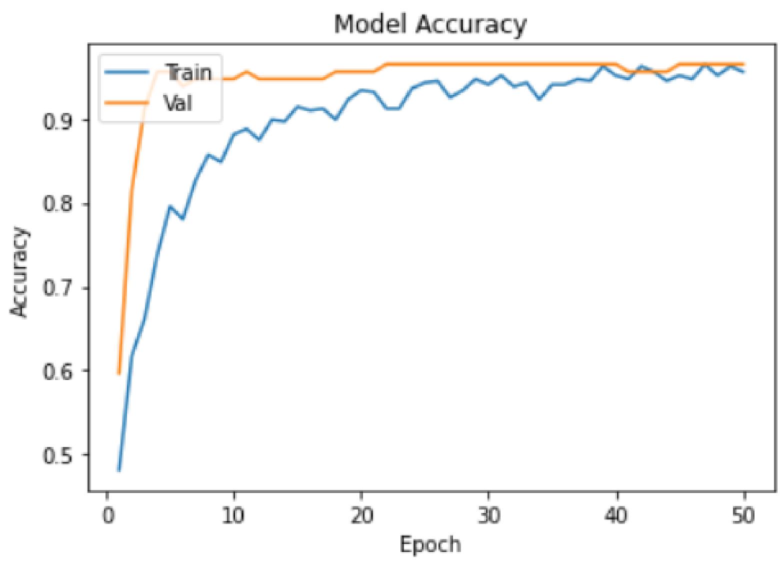


Figure 13 LSTM training and validation accuracy

We also perform some other tests to find the effectiveness of the proposed models. We varied the dataset sizes, ranging from small subsets to the entire dataset, to assess the scalability of our models. The experiments involved measuring the performance metrics as the dataset size increased, allowing us to evaluate the models' efficiency with larger datasets. We performed time measurements by assessing the training time taken by the models to process varying amounts of data. We also access models by giving varied inputs in real time. The models predict the output efficiently on new data. Figure 14 presents the scalability analysis of the proposed model on varied dataset sizes.

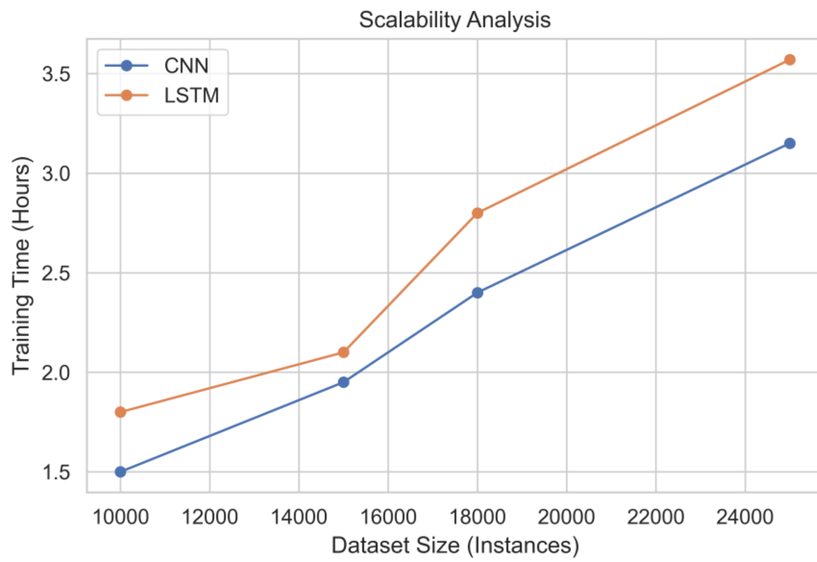


Figure 14 Scalability Analysis of Proposed Models

To demonstrate the significance of our experimental results, we performed statistical analyses, including ANOVA tests and t-tests. These analyses allowed us to quantitatively assess the observed differences in performance metrics

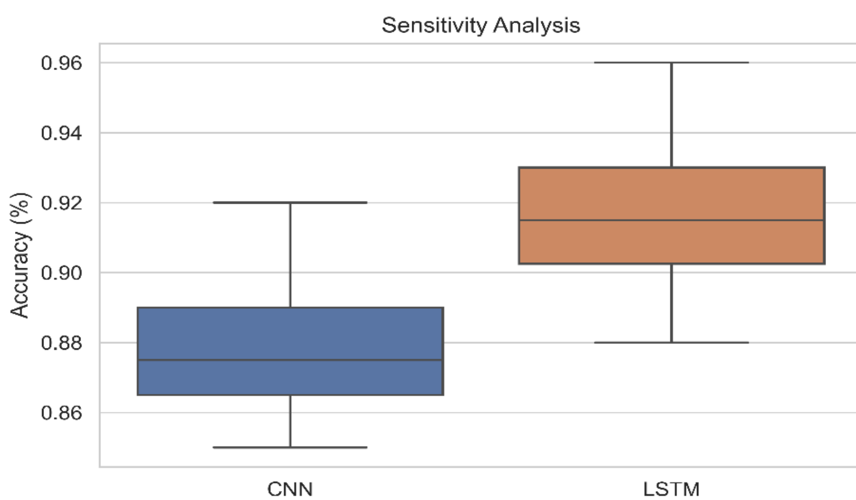


Figure 15: Sensitivity Analysis of Proposed Models

and establish the statistical significance of our findings. The values of T-statistics and P-value are -1.6947980485980951 and 0.1410488553056692, respectively. The values of F-statistics and P-value in ANOVA are 2.872340425531916 and 0.14104885530566907, respectively. Figure 15 presents the sensitivity analysis of the proposed model. We performed a well-known statistical test named the Wilcoxon signed-rank test to compare the performance of the proposed models. The Wilcoxon signed-rank test detects differences in the distributions of paired samples. To perform this test we split the dataset into different sizes, i.e., from 10,000 to 24,000 with an increment of 2,000 in every iteration. With a p-value of 0.023 and a test statistic of 2.00 from the Wilcoxon test, there were significant differences between the models. These statistical evaluations verify that, in terms of accuracy and F1 score, the LSTM model performs much better than the CNN model. The performance of the proposed models on different ratios of the dataset for the Wilcoxon test is illustrated in Figure 16.

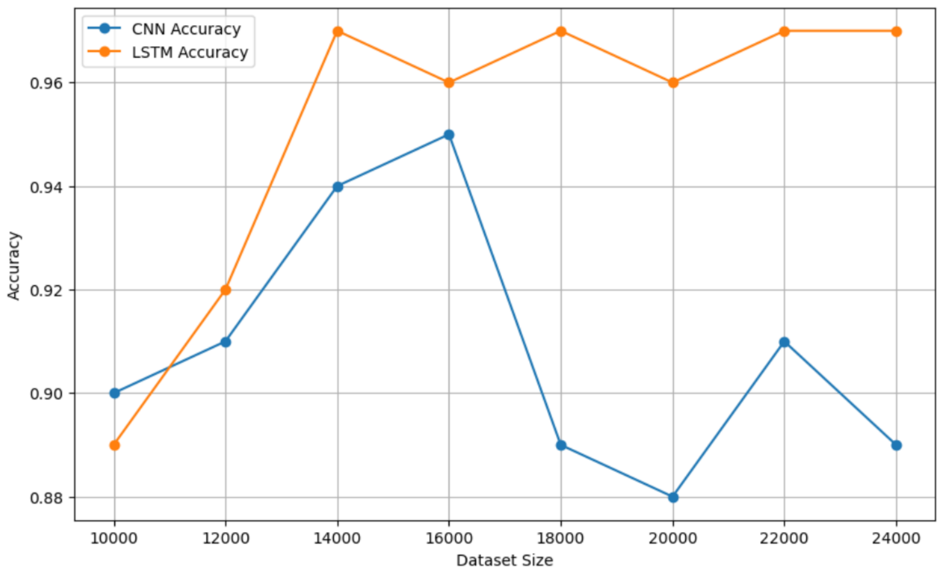


Figure 16: Performance of proposed model in Wilcoxon signed-rank test

We compared proposed techniques with previously proposed methods, as discussed in Section 2, to find the effectiveness of our proposed techniques. Khan et al. [25] proposed a transformer-based approach using various ML and DL models on an Urdu dataset of 9312 reviews. They achieved 77.61% accuracy using the mBert model. Sehar et al. [41] created a custom dataset for Urdu reviews and applied a deep neural network model for Urdu text classification. They used language rules for optimizing the polarity detection from the Urdu text. Their approach achieved 93.05% accuracy and 92% F1 score. Meta-Learning was used for Urdu text classification by Ahmed et al. [22]. They used various ML models in their approach and achieved an accuracy of 86.42%. Another novel technique was proposed by shabbir et al. [42] using various DL models for Urdu text classification. Their best model achieved 89.36% accuracy using the IMDB movies review dataset. The comparison of the proposed models with previous models is illustrated in Figure 2. Figure 17 compares previously proposed approaches and our proposed CNN and LSTM models

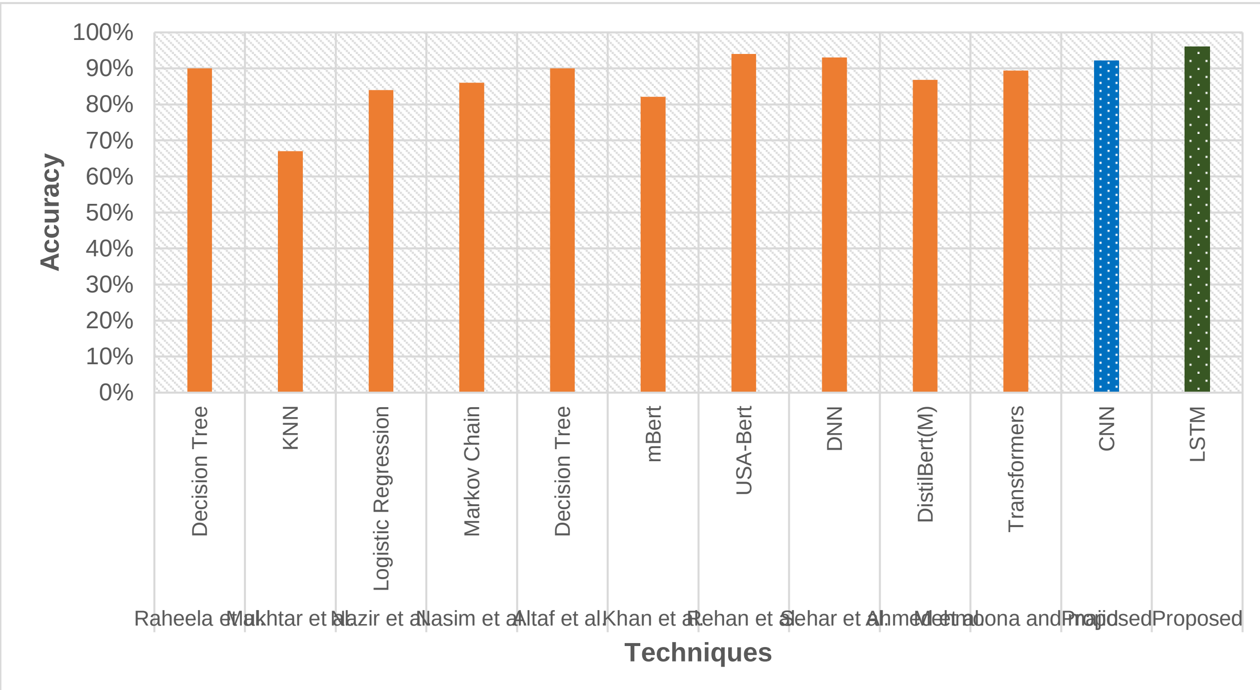


Figure 17: Comparison with previous techniques

As shown in Figure 17, the proposed techniques achieve a higher accuracy than previously proposed techniques. Raheela et al. [18], and Mukhar et al. [1] used traditional machine learning techniques and got 90% and 67% accuracy, respectively. Naesim et al. [25] used the Markov chains model that achieves 86% accuracy, while Amjad et al. [29] used machine learning techniques that achieve 85% accuracy on Urdu data. The reason for getting the higher level of accuracy of proposed models is the use of deep learning models with manually annotated data. Before that, no work had been done on Urdu SA manually annotated data using a deep learning model. As deep learning models can capture essential features from data, they perform better than traditional machine learning and lexicon-based approaches in sentiment analysis.

## 6 DISCUSSION

Sentiment analysis has various uses like analyzing customer feedback, brand monitoring, stock market analysis, campaign monitoring, etc. The motivation behind this research stems from the increasing significance of sentiment analysis in the contemporary landscape of information exchange, particularly in the context of Urdu, a language spoken by over 100 million people worldwide. As the internet becomes a ubiquitous platform for expressing opinions and sharing reviews, the need for effective sentiment analysis tools in Urdu has become evident. Existing sentiment analysis techniques, largely developed for English, face challenges when applied to Urdu due to its complex morphological structure and script differences. This motivates our focus on deep learning-based Urdu sentiment analysis using deep earning models. The pivotal role of sentiment analysis in decision-making processes for producers, service providers, and organizational leaders is another driving force. User-generated content on social media platforms and blogs serves as a valuable resource for understanding public sentiment. For instance, in election scenarios, sentiment expressed on social media provides insights into the popularity status of political parties and leaders, guiding strategic planning. The proposed study aims to utilize this valuable resource by conducting sentiment analysis on Urdu, thus contributing to informed decision-making processes.As discussed above, Urdu is now widely used in social media platforms and other blogging websites so that Urdu SA can be used for the above-discussed purposes. The experiments of the proposed deep learning models show the effectiveness of models to traditional machine learning models, i.e., SVM, Decision tree, etc., for Urdu SA. These models can effectively capture semantic elements in the text. Furthermore, because these models rely upon word embeddings, the meanings of the words represented by word vectors in the embedding space are also incorporated in a specific form.

The data for the study is collected from various sources, as discussed in Section 4. Annotation plays a vital role in the efficiency of models, so three annotators do manual annotation of the data set. For better annotation and removing conflicts between annotators, Kappa statistics is used. The CNN and LSTM model is trained on Urdu text data, and models are analyzed based on accuracy. Furthermore, the results demonstrate that the best model, such as LSTMs, can capture the Urdu text's context. The proposed LSTM model outperforms traditional machine learning models, so it can be used to analyze Urdu sentiment for various tasks, i.e., customer feedback, brand monitoring, campaign monitoring, etc. To enhance the accuracy of Urdu SA, more research is needed using intelligent techniques as several issues like Sarcasm can affect the accuracy of automated models. Other methods like ensemble and hybrid models, can be used for better accuracy with larger data sets. For various services/products, data is gathered from many websites. Each site has a reasonable number of reviews. However, not all individuals express their opinions on these platforms. It would be necessary to gather responses from all users who use various services at various sites for better results. Moreover, the proposed CNN and LSTM models can be combined to get better results.

# 7 CONCLUSION

Deep learning approaches have recently proven significant in various applications, including machine translation, image recognition, object detection, and natural language processing (NLP) and Multimedia Internet of Things (MIoT). These approaches recently gained popularity for sentiment classification, as they usually outperform traditional machine learning algorithms when the data set is huge. Sentiment analysis of the Urdu language is an emerging research domain in the current era. Two deep learning models, CNN, and LSTM, are used for this purpose. A data set of 25 thousand reviews collected from various Urdu platforms is used to train proposed models. The preprocessing step is done using the Urdu Hack library, and models are trained on multiple combinations of hyper-parameters. In the evaluation phase of models, the LSTM model outperforms the CNN model by achieving higher accuracy and F1 score. The proposed models are also compared with previous Urdu SA techniques, showing enhanced results in this study. In the future, we will explore various CNN architectures and ensemble models with larger data sets to improve the results.

## ACKNOWLEDGMENTS

## REFERENCES


[1] Mukhtar, N. and M.A. Khan, Urdu sentiment analysis using supervised machine learning approach. International Journal of Pattern Recognition and Artificial Intelligence, 2018. 32(02): p. 1851001.

[2] Rehman, Z.U. and I.S. Bajwa. Lexicon-based sentiment analysis for Urdu language. in 2016 sixth international conference on innovative computing technology (INTECH). 2016. IEEE.

[3] Iqbal, Amjad, Rashid Amin, Javed Iqbal, Roobaea Alroobaea, Ahmed Binmahfoudh, and Mudassar Hussain. "Sentiment analysis of consumer reviews using deep learning." Sustainability 14, no. 17 (2022): 10844.

[4] Dang, W., Cai, L., Liu, M., Li, X., Yin, Z., Liu, X.,... Zheng, W. (2024). Increasing Text Filtering Accuracy with Improved LSTM. COMPUTING AND INFORMATICS, 42(6), 1491-1517. doi: https://doi.org/10.31577/cai_2023_6_1491

[5] Saeed, Muhammad, Naeem Ahmed, Abid Mehmood, Muhammad Aftab, Rashid Amin, and Shahid Kamal. "Sentiment Analysis for COVID-19 Vaccine Popularity." KSII Transactions on Internet & Information Systems 17, no. 5 (2023).

[6] Yin, L., Wang, L., Cai, Z., Lu, S., Wang, R., AlSanad, A.,... Zheng, W. (2024). DPAL-BERT: A Faster and Lighter Question Answering Model. Computer Modeling in Engineering & Sciences, 141(1), 771-786. doi: https://doi.org/10.32604/cmes.2024.052622

[7] Zheng, W., Lu, S., Cai, Z., Wang, R., Wang, L.,... Yin, L. (2024). PAL-BERT: An Improved Question Answering Model. Computer Modeling in Engineering & Sciences, 139(3), 2729-2745. doi: https://doi.org/10.32604/cmes.2023.046692

[8] Tayir, T., & Li, L. (2024). Unsupervised Multimodal Machine Translation for Low-resource Distant Language Pairs. ACM Trans. Asian Low-Resour. Lang. Inf. Process., 23(4), 1-22. doi: https://doi.org/10.1145/3652161

[9] Sarim, M. Urdu Natural Language Processing Issues and Challenges: A Review Study. in Intelligent Technologies and Applications: Second International Conference, INTAP 2019, Bahawalpur, Pakistan, November 6–8, 2019, Revised Selected Papers. 2020. Springer Nature.

[10] Liu, Z., Xiong, X., Li, Y., Yu, Y., Lu, J., Zhang, S.,... Xiong, F. (2024). HyGloadAttack: Hard-label black-box textual adversarial attacks via hybrid optimization. Neural Networks, 178, 106461. doi: https://doi.org/10.1016/j.neunet.2024.106461

[11] Song, L., Chen, S., Meng, Z., Sun, M., & Shang, X. (2024). FMSA-SC: A Fine-Grained Multimodal Sentiment Analysis Dataset Based on Stock Comment Videos. IEEE Transactions on Multimedia, 26, 7294-7306. doi: 10.1109/TMM.2024.3363641

[12] Lal, M., et al., A Systematic Study of Urdu Language Processing its Tools and Techniques: A Review.

[13] He, S., Luo, H., Jiang, W., Jiang, X., & Ding, H. (2024). VGSG: Vision-Guided Semantic-Group Network for Text-Based Person Search. IEEE Transactions on Image Processing, 33, 163-176. doi: 10.1109/TIP.2023.3337653

[14] Khalid, U., et al., Co-occurrences using Fasttext embeddings for word similarity tasks in Urdu. arXiv preprint arXiv:2102.10957, 2021.

[15] Rasheed, I., H. Banka, and H.M. Khan, A hybrid feature selection approach based on LSI for classification of Urdu text, in Machine Learning Algorithms for Industrial Applications. 2021, Springer. p. 3-18.

[16] Li, T., Li, Y., Zhang, M., Tarkoma, S., & Hui, P. (2023). You Are How You Use Apps: User Profiling Based on Spatiotemporal App Usage Behavior. ACM Trans. Intell. Syst. Technol., 14(4). doi: 10.1145/3597212

[17] Huang, C., Tu, Y., Han, Z., Jiang, F., Wu, F.,... Jiang, Y. (2023). Examining the relationship between peer feedback classified by deep learning and online learning burnout. Computers & Education, 207, 104910. doi: https://doi.org/10.1016/j.compedu.2023.104910

[18] Bibi, Raheela, Usman Qamar, Munazza Ansar, and Asma Shaheen. "Sentiment analysis for Urdu news tweets using decision tree." In 2019 IEEE 17th international conference on software engineering research, management and applications (SERA), pp. 66-70. IEEE, 2019.

[19] Mukund, S., R. Srihari, and E. Peterson, *An Information-Extraction System for Urdu---A Resource-Poor Language.* ACM Transactions on Asian Language Information Processing (TALIP), 2010. **9**(4): p. 1-43.

[20] Mukhtar, N., et al., *Recognition and Effective Handling of Negations in Enhancing the Accuracy of Urdu Sentiment Analyzer.* Mehran University

Research Journal of Engineering and Technology, 2020. **39**(4): p. 759-771.

[21] Altaf, A., Anwar, M. W., Jamal, M. H., & Bajwa, U. I. (2023). Exploiting Linguistic Features for Effective Sentence-Level Sentiment Analysis in Urdu Language. *Multimedia Tools and Applications*, 1-27.

[22] Ahmed, K., Nadeem, M. I., Li, D., Zheng, Z., Al-Kahtani, N., Alkahtani, H. K., ... & Mamyrbayev, O. (2023). Contextually Enriched Meta-Learning Ensemble Model for Urdu Sentiment Analysis. *Symmetry*, *15*(3), 645.

[23] Muhammad, K. B., & Burney, S. A. (2023). Innovations in Urdu Sentiment Analysis Using Machine and Deep Learning Techniques for Two-Class Classification of Symmetric Datasets. *Symmetry*, *15*(5), 1027.

[24] Mukhtar, N., M.A. Khan, and N. Chiragh, Lexicon-based approach outperforms Supervised Machine Learning approach for Urdu Sentiment Analysis in multiple domains. Telematics and Informatics, 2018. 35(8): p. 2173-2183

[25] Khan, L., Amjad, A., Ashraf, N., & Chang, H. T. (2022). Multi-class sentiment analysis of urdu text using multilingual BERT. Scientific Reports, 12(1), 5436.

[26] Dong, J., Hu, J., Zhao, Y., & Peng, Y. (2024). Opinion formation analysis for Expressed and Private Opinions (EPOs) models: Reasoning private opinions from behaviors in group decision-making systems. Expert Systems with Applications, 236, 121292. doi: https://doi.org/10.1016/j.eswa.2023.121292

[27] Zhenfeng Liu, Jian Feng, Lorna Uden, 2023. Technology opportunity analysis using hierarchical semantic networks and dual link prediction. Technovation 128, 102872. https://doi.org/10.1016/j.technovation.2023.102872

[28] Nazir, M.K., et al. Sentiment Analysis of User Reviews about Hotel in Roman Urdu. in 2020 14th International Conference on Open Source Systems and Technologies (ICOSST). 2020. IEEE.

[29] Amjad, K., et al. Exploring Twitter news biases using urdu-based sentiment lexicon. in 2017 International Conference on Open Source Systems & Technologies (ICOSST). 2017. IEEE.

[30] Hashim, F. and M. Khan, *Sentence level sentiment analysis using urdu nouns.* Department of Computer Science, University of Peshawar, Pakistan, 2016: p. 101-108.

[31] Mukund, S. and R.K. Srihari. Analyzing Urdu social media for sentiments using transfer learning with controlled translations. in Proceedings of the Second Workshop on Language in Social Media. 2012.

[32] Nasim, Z. and S. Ghani, *Sentiment Analysis on Urdu Tweets Using Markov Chains.* SN Computer Science, 2020. **1**(5): p. 1-13.

[33] Asghar, M.Z., et al., Creating sentiment lexicon for sentiment analysis in Urdu: The case of a resource-poor language. Expert Systems, 2019. **36**(3): p. e12397.

[34] Yang, D., Zhu, T., Wang, S., Wang, S., & Xiong, Z. LFRSNet: A Robust Light Field Semantic Segmentation Network Combining Contextual and Geometric Features. Frontiers in Environmental Science, 1443. doi: 10.3389/fenvs.2022.996513

[35] Zheng, W., Lin, L., Wu, X., & Chen, X. (2024). An Empirical Study on Correlations Between Deep Neural Network Fairness and Neuron Coverage Criteria. IEEE Transactions on Software Engineering, 50(3), 391-412. doi: 10.1109/TSE.2023.3349001

[36] Pan, S., Xu, G. J. W., Guo, K., Park, S. H., & Ding, H. (2023). Video-Based Engagement Estimation of Game Streamers: An Interpretable Multimodal Neural Network Approach. IEEE Transactions on Games. doi: 10.1109/TG.2023.3348230

[37] Anwar, W., et al., An empirical study on forensic analysis of urdu text using LDA-based authorship attribution. IEEE Access, 2018. **7**: p. 3224-3234.

[38] Al-Thubaity, A., et al. A saudi dialect twitter corpus for sentiment and emotion analysis. in 2018 21st Saudi Computer Society National Computer Conference (NCC). 2018. IEEE.

[39] Ding, J., Chen, X., Lu, P., Yang, Z., Li, X.,... Du, Y. (2023). DialogueINAB: an interaction neural network based on attitudes and behaviors of interlocutors for dialogue emotion recognition. The Journal of Supercomputing, 79(18), 20481-20514. doi: 10.1007/s11227-023-05439-1

[40] Ahmed, S.B., et al., Handwritten Urdu character recognition using one-dimensional BLSTM classifier. Neural Computing and Applications, 2019. 31(4): p. 1143-1151.

[41] Sehar, U., Kanwal, S., Allheeib, N. I., Almari, S., Khan, F., Dashtipur, K., ... & Khashan, O. A. (2023). A hybrid dependency-based approach for Urdu sentiment analysis. *Scientific Reports*, *13*(1), 22075.

[42] Shabbir, M., & Majid, M. (2024, February). Sentiment Analysis From Urdu Language-based Text using Deep Learning Techniques. In *2024 5th International Conference on Advancements in Computational Sciences (ICACS)* (pp. 1-5). IEEE.