IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 October 2024, accepted 14 December 2024, date of publication 26 December 2024, date of current version 2 January 2025.

Digital Object Identifier 10.1109/ACCESS.2024.3522992

RESEARCH ARTICLE

Optimized Identification of Sentence-Level Multiclass Events on Urdu-Language-Text Using Machine Learning Techniques

SOMIA ALI, UZMA JAMIL, (Member, IEEE), MUHAMMAD YOUNAS, BUSHRA ZAFAR, AND MUHAMMAD KASHIF HANIF

Department of Computer Science, Government College University Faisalabad, Faisalabad 38040, Pakistan

Corresponding author: Uzma Jamil (uzma.jamil@gcuf.edu.pk)

ABSTRACT In today's digital world, social media platforms generate a plethora of unstructured information. However, for low-resource languages like Urdu, there is a scarcity of well-structured data for specific tasks such as event classification. Urdu, a language prominent in South Asia, has boasted a complex morphological structure with unique features but has lacked standard linguistic resources like datasets. Long-text classification has demanded more effort than short-text classification due to its expansive vocabulary, information redundancy, and noise. Text processing has been the latest trend in research, with many machine learning and deep learning techniques widely used for it. Multiclass classification has been utilized to classify different languages for various purposes. In this research, a multiclass classification for the Urdu language was performed using a text dataset taken from five different social media platforms including Geo News, Samaa News, Dawn News, Express News, and Urdu Blogs totaling 103,771 sentences. We used sentence-level classification to categorize sentences including terrorist attacks, national news, sports, entertainment, politics, safety, earthquakes, fraud and corruption, sexual assault, weather, accidents, forces, inflation, murder and death, education, and international news. Deep learning, transformer-based and machine learning classifiers are used for event classification. The SMFCNN classifier achieved the greatest accuracy of 88.29%. We incorporated transformer-based models, with the proposed XLM-R+ model demonstrating superior performance with an accuracy of 89.8%. Our results were compared to previously reported techniques that used traditional models, highlighting the significant improvements offered by our approaches. The novelty of this research lies in the inclusion of 16 event categories to broaden coverage and the implementation of the SMFCNN and transformer-based algorithms. This study highlights the potential of deep learning and transformer-based models in enhancing the accuracy and generalizability of multiclass classification in low-resource languages Urdu.

INDEX TERMS Sentence level classification, deep learning, machine learning, Urdu language, event classification.

I. INTRODUCTION

Social media cannot be called just Internet sites, as they have become the necessary means of communication in the modern world, helping their users find information and communicate with others and with content in real-time [1]. However, the use of these platforms provides a massive volume of unstructured textual content making it difficult for users and businesses to relevancy filter information [2]. This has made it even more challenging for any business entity to effectively address the needs of their target clients through the internet given these issues of information overload. This has led to the use of Natural Language Processing (NLP) and Machine Learning (ML) approaches to solve this problem

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

by gaining meaning from the large amount of text data to optimize the provision of valuable content to users [3].

One of the significant applications of NLP on these platforms is to determine the kind of event that occurred or is in the process of happening for instance the event may be a natural disaster, a political event, or sports or entertainment news among others from text content. Despite these advances, the mentioned progress reforms the work on high-resource languages mostly. Urdu is a low-resource language spoken largely in Pakistan and India, with a growing global audience. It is written in a modified form of the Arabic script, with additional characters to represent sounds unique to the language. Despite its importance as a language of culture, literature, and communication, Urdu has long been considered a low-resource language in the context of NLP and ML [4]. This poses a significant problem, especially for researchers who are involved in Urdu text classification research.

Like other Perso-Arabic scripts, Urdu script and its writing system provide its linguistic complexities, including the use of ligatures, diacritical marks, and multiple layers of contextual meanings. Figure 1 presents the basic characters and numerals of the Urdu script, showcasing the unique shape and flow of each character. Due to the inherent complexity and the high number of ligatures (around 24,000), tasks like sentence segmentation and part-of-speech tagging become challenging in Urdu. The script's connected nature complicates natural language processing tasks compared to languages with simpler, disconnected scripts [5].

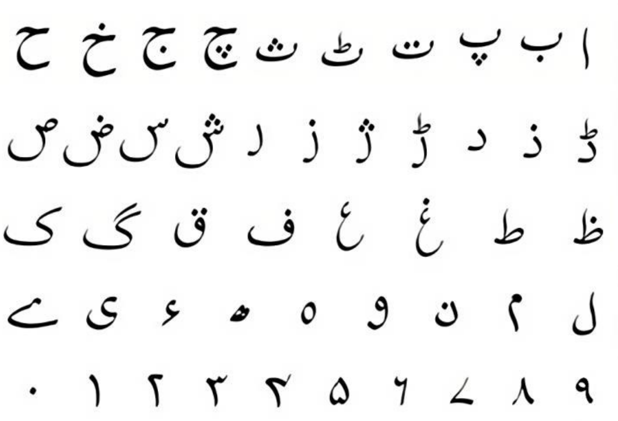

FIGURE 1. Urdu basic characters and numerals.

The joining natures enrich Urdu with almost 24,000 ligatures. This alphabet set is being used as a superset for all Urdu script-based language alphabets like Arabic and Persian having 28 and 32 characters respectively [11]. Table 1 demonstrates examples of Urdu words, their corresponding ligatures, and the individual characters that compose them. It explains how Urdu script, which has a fairly large number of ligatures, words are formed by joining the characters worked out to maintain the flow and style of the language. These ligatures have a particularly important function of the correct representation and proper pronunciation of words, which is why they are important in text classification and linguistic analysis.

TABLE 1. Urdu characters with ligatures and words [10].

Urdu Words	Urdu Ligatures	Urdu Characters
پاکستان	پاک، ستان	پ،ا،ک،س،ت،ا،ن
خوش آمدید	خوش، آمدید	خ، و، ش، آ، م، د، ی، د
زندگی	زند، گی	ز، ن، د، گ، ی
شاعر	شاع،ر	ش، ا، ع، ر
کتاب	کتاب	ک، ت، ا، ب

There are still deficiencies, for instance, for POS markers, titles, entity identifiers, and other annotation instruments for sentence detection and classification in the Urdu language. It is a fact that most people do not understand the meaning and usage of some of the Urdu words. This leads to the creation of linguistically ambiguous content thus making sentence classification an extremely difficult and counter intuitive process [4].

Several studies conducted fairly well recently in the task of building large-scale word embedding models for Urdu [7]. Nonetheless, such big annotated sets were not available; scientists used large-scale word embedding models to improve the results of natural language processing for Urdu by training the models on huge amounts of textual information. Moreover, the other approaches for the enhancement of NLP for Urdu other than word embedding have also been explored by researchers. For example, [6] developed a morphological analysis of Urdu text using both rule-based and statistical systems. Urdu has a rather complex grammar, and this can also create issues for NLP tools. Here are several examples:

1 Ligature
2 Diacritics
3 Context sensitivity
4 Direction of writing
5 Diagonal writing style
6 The dotting problem
7 Script
8 Degrees of the the verb to distinguish between past present, and future actions [8]
9 A complex system of noun declension and case marking

These aspects of the Urdu grammar impede the NLP model implementation due to the lack of large-scale annotated data and inadequate knowledge of Urdu languages [9]. A unique approach for automatically annotating Urdu text is developed e.g. part-of-speech tagging and dependency parsing [8]. These methods involve analyzing the grammatical structure of an Urdu text to determine the syntactic roles of words and their relationships.

Using supervised learning approaches, researchers were able to extract 12 separate event types from the Urdu Language Text dataset with an accuracy of 84% [11]. Furthermore, researchers investigated document-level classification, where they introduced the Convolutional Neural Network

(CNN) for successful text classification [10]. Based on the literature reviewed, proposed research questions are presented in the following section.

A. RESEARCH QUESTIONS

To evaluate and assess the proposed approach and methodology, the following are the research questions:

RQ 1. How unstructured Urdu text datasets will be classified into different event classes?

RQ 2. How do machine-learning classifiers that rely on manual feature engineering perform compared to classifiers that automatically learn hierarchical representations in multiclass event classification on the Urdu Language Text dataset?

RQ 3. What will be the impact of different Urdu language domains of knowledge in event classification?

There exists a considerable gap in text classification based on sentences, which needs to be addressed. Due to the limited size of the dataset, the models have been poorly trained, resulting in low accuracy. To overcome this issue, an existing Urdu Language Text (ULT) dataset has been enhanced from social media sites, and the training has been enhanced to achieve the best and most accurate results. The precise problem statement that highlights the need of the research is as follows:

"The development of an accurate and precise event classification approach that incorporates noise removal, stop words removal, sentence filtering, sentence labeling, and feature selection, vectorization using deep learning and machine learning algorithms to accurately classify events into their appropriate classes has been addressed".

This research aims to use deep learning and machine learning algorithms to classify sentences in the Urdu Language Text (ULT) dataset. With a larger dataset size, the training process will be smoother, and testing will yield better results for the ULT dataset extracted from social media sites.

The motivation for this research is due to limited resources in Urdu text processing, there is a significant gap to fill in different domains such as classification with unstructured datasets, a multiclass classification that helps to easily classify types of events at different levels like document level, sentence level, phrase level, etc. Overcoming these gaps is essential to help predictors easily analyze Urdu text.

B. CONTRIBUTION

This research contributes to the existing literature on natural language processing and helps to solve the problems of event classification in the Urdu language, which belongs to the Group of low-resource languages. To address this problem, we present a novel solution that encompasses older machine learning methodologies as well as newer deep learning and transformer learning models to categorize Urdu sentences into 16 event classes. Our work is methodologically dissimilar to prior studies in two ways: We cover a more extensive range of first-order events, and we achieve a superior level of correctness in the categorization of sentence-level events.

The dataset used in this research was gathered from five major Urdu-language platforms: Samaa News, Dawn News, Geo News, Express News, and Urdu Blogs. These sources provide a broad spectrum of content, including events related to terrorist attacks, national news, sports, entertainment, politics, and several other domains. A web scraper was used to automatically extract the data from these platforms, resulting in a comprehensive raw dataset. By implementing advanced models CNN, RNN, LSTM-RNN, mBERT, RoBERTa, and proposed XLM-R+ and optimizing their hyperparameters, we achieve competitive accuracies, with the SMFCNN model attaining an impressive accuracy of 88.29% and XLM-R+ with 89.8% accuracy. Thus, these results meet a major research need by providing benchmark performance numbers on propriety models applied to Urdu text classification. Our work not only provides a method for increasing classification accuracy for low-resource languages but also lays the groundwork for more research on multilingual natural language processing.

II. RELATED RESEARCH

In this section, we discuss the techniques and methods that have been applied in natural language processing (NLP), with a specific focus on Urdu-language text classification. For sentiment analysis and other problems, much improvement has been seen in NLP but the problem of event classification for low-resource languages such as Urdu is still open. To support this discussion, we present a review of prior work and its drawbacks in studying sentence-level event classification, setting the context for our research.

A. TEXT CLASSIFICATION IN NLP

Text classification is one of the main problems in NLP: it aims to assign categories or predefined tags to a certain text based on its content [9], [12]. This task is pivotal to many applications including event detection, spam filtering, topic classification, and sentiment analysis [13]. Although classification tasks are well explored in high-resource languages like English, their analysis in low-resource languages such as Urdu is very rare and limited in scope. In the case of the Urdu language, the textual intricacy of the script, and the absence of enriched datasets, create further difficulties in the employ of conventional text categorization approaches [4].

The mainstream categorization approaches in the text classification context of NLP are the rule-based approaches, statistical models, and finally machine learning-based methods [10]. There is significant use of supervised learning, where models are built on fully annotated datasets, in most NLP classification tasks [14]. To minimize noise that affects the classification algorithms, basic preprocessing steps such as tokenization, stemming, lemmatization, and stop-word removal mostly have been applied [16]. There are precision, recall, F1-score, accuracy, etc., which are commonly used to measure the performance of the text classification models [17]. Among them, the neural network-based approaches have attracted much attention in recent years, and are essentially considered as the most accurate and powerful methods for text classification [18].

However, text classification in LRLs, comprised of Urdu in our case, has several issues hindering its growth. Some of the challenges offered by the linguistic features of Urdu include: The script used in Writing Urdu is Incomplete compatible with Unicode; the ligatures used in writing Urdu present NLP problems that are hard to handle; and the diacritical marks used in writing Urdu are a cause of difficulty for NLP systems. Further, an Urdu-like Arabic writing system is used in an Arabic-based script and in addition to that includes a few other challenges which are discussed below – Urdu script is rich in ligatures, a feature that needs special consideration while preprocessing Urdu text as well as in designing the model [5].

B. URDU LANGUAGE PROCESSING AND EVENT CLASSIFICATION

The proposed language, Urdu, belongs to the low-resource languages, and in particular, it has a limited number of annotated datasets and limited computational resources allocated for its processing. Although tremendous progress has been made in tasks such as sentiment analysis for the Urdu language [14], event classification has not received much attention in the past, which is still a research void. Event extraction is the task of extracting, from text, different and separate events, including natural disasters, political changes, and sports results. Unlike sentiment analysis which concerns the determination of valence from the text, event classification embraces the factual contents and contextual meanings of the sentences [9], [14].

While there are no English translation media materials for specific works, several papers have been written on unique tasks in Urdu NLP, namely hate speech detection, indicating an increasing uptake of computational models to address the language. For example, [6] suggested that the transfer learning method should be applied to detect hate speech in the Urdu language. They were also able to attain better results on low-resource languages for instance Urdu hence making the case for transfer learning for such languages. But this work was just confined to hate speech detection and was not solved with this approach to classify other possible events in Urdu text. In the same manner, [19] followed the geospatial methods to map hate speech in Roman Urdu, to pinpoint the hot spots of hate speech. While these studies offer much-needed information regarding specific applications, event classification continues to be a somewhat overlooked area.

Another important stream of studies in Urdu NLP is related to the identification of cyberbullying. [20] proposed the use of an ensemble approach for identifying patterns in Roman Urdu micro-text for cyberbullying detection. The generative model for detection employment multiple classifiers and an ensemble learning approach to enhance the detection level of accuracy. While this method worked well to deal with cyberbullying detection it did not delve into event classification. This has shown the need to develop better models especially to categorize multiple types of events in the Urdu text.

Moreover, the existence of approaches to recognize and classify the handwritten Urdu numerals has also been presented through deep learning [21], [22]. These works utilized Convolutional Neural Networks (CNNs) that correctly read Urdu handwritten numerals and therefore established the usability of deep learning for Urdu text. In parallel, [23] used CNN and LSTM networks for the identification of short paraphrases of texts in Urdu. This research affirmed all other findings and supported the utilization of deep learning for paraphrase detection nonetheless, it has not offered a solution to the event classification at the sentence level.

C. ADVANCES IN MULTICLASS CLASSIFICATION AND GAPS IN EVENT DETECTION

Multiclass classification is a significant problem in NLP where text to be analyzed is classified into more than one predefined class. This is even harder where limited annotated data, as seen with Urdu, exists and the linguistic patterns are difficult to decipher. Even though multiclass classification has been rather effective when applied to several domains, it has a rather poor representation in terms of event classification especially for the Urdu language.

For example, [4] presented a new multi-target classification technique to address the issue of semantic tagging in Urdu revealing that the proposed work exhibited multi-target tagging to be quite effective. However, this work did not cover event detection. A survey on sentiment analysis and emotion recognition from Khasi text data by [13] highlighted the need for language-specific models in low-resource languages. Although this study provided valuable insights into sentiment analysis, it did not address event classification, which requires distinct approaches due to its focus on factual and contextual event categorization.

Previous research in NLP has also explored semi-supervised learning for text classification. For instance, [24] presented a comprehensive overview of semi-supervised learning approaches, which can be particularly useful when labeled data is scarce. Although these techniques have been applied to tasks like sentiment analysis and text categorization, they have yet to be fully leveraged for event classification in Urdu. As such, our research aims to bridge this gap by applying state-of-the-art deep learning techniques to event classification, specifically addressing the challenges posed by the Urdu language.

D. EVENT CLASSIFICATION TECHNIQUES FOR LOW-RESOURCE LANGUAGES

The development of event classification models for low-resource languages like Urdu has been hampered by the limited availability of high-quality, annotated datasets. Most of the research in NLP for Urdu has focused on sentiment analysis or hate speech detection, with little emphasis on event classification. Word embedding techniques, such as word2vec and GloVe, have been applied successfully in text classification for high-resource languages [3], [28]. These methods, combined with machine learning algorithms like Naive Bayes and Support Vector Machines (SVMs), have demonstrated strong performance in various domains. However, there remains a gap in their application to Urdu event classification.

Some studies have explored hybrid models, combining traditional machine learning and deep learning techniques. For example, in [34], the authors presented a combination of feature extraction using CNNs and classification using Naive Bayes. This method was tested with satisfactory results in text classification however event classification using this method was not attempted in Urdu. Using deep learning models and transformer-based architecture for the classification of events, our study expands these findings to overcome the shortcomings of the current models.

E. DEEP LEARNING AND TRANSFER LEARNING FOR URDU NLP

Bert, RoBERTa, XLM-R the use of transformer-based models has transformed NLP by enhancing cross-lingual language understanding as well as handling low-resource languages more efficiently. Exactly these models are helpful for event classification as they take into account the context and dependencies in the text. Even though there are some works related to sentiment analysis for Urdu using transformer models [56], [61], there is limited work that can be done in the event classification. To fill these gaps, our study builds upon XLM-R+, a transformer model particularly suited for low-resource languages such as Urdu.

[58] did a study on deep learning for detecting sarcasm in the Urdu language; the author used CNNs and LSTMs to increase the classification rate. While obtaining high accuracy in a similar task of sarcasm detection, the study did not generalize the scope of the work to the 14-class event classification problem. Likewise, [59] proposed the detection of Offensive Language in Urdu with the help of Convolutional Neural Structures incorporating attention mechanisms at equally an identical satisfactory rate. Still, the aspect of event detection in Urdu has not been greatly studied and improved, which is what our study aims to address.

F. CHALLENGES IN EVENT DETECTION FOR URDU TEXT

Urdu is a tough language for NLP techniques mainly because of the WASEA properties, Complex Morphology, Context sensitivity of Word forms, Ligatures, and Diacritical marks. Compared to previous work which employs methods such as hate speech detection, as well as sentiment analysis [58], [59], the task of event detection is more complicated since it needs models that can comprehend the various syntactic and semantic contexts of a sentence. For instance, the study triangulating deep and big learning approaches to detect fake news in Urdu found CNNs and RNNs effective for high accurate fake news classification [60]; however, the research was constrained by areas such as the quality and diversity of the dataset. Similarly, in sarcasm detection [66] and offensive language detection [67], deep learning models were used, but these were more of a specific-task approach they did not address the problem of event classification.

To overcome these challenges in our work, we introduce a dataset that is extracted from five social media platforms including information about natural disasters, political events, and sports results. By applying advanced deep learning models and transformer architectures, we aim to significantly improve the performance of event classification for Urdu text, filling the gap left by previous research.

In summary, while there has been significant progress in NLP tasks like sentiment analysis and hate speech detection for Urdu, there remains a considerable gap in event classification. Our research contributes to addressing this gap by introducing advanced DL techniques and a robust dataset specifically focused on event classification for Urdu text, as summarized in Table 2.

III. METHODOLOGY

This research consists of multiple stages, beginning with the collection of an Urdu-language dataset. Existing datasets for Urdu sentence-level classification are limited, with only one dataset containing 0.1 million labeled sentences [11], [40]. Therefore, this study focuses on generating a larger dataset specifically for sentence-level event classification to address this gap. We opted for sentence-level classification to achieve greater precision in identifying events. Sentences often encapsulate distinct events even when similar in structure but occurring in different contexts.

The core of this research is event classification, distinguishing it from sentiment analysis [13] or related NLP tasks. Classifying events at the sentence level requires a specialized approach to accurately identify the nature of the event, ensuring that our model delivers practical applications like improved news categorization and social media analysis.

The transformer-based model XLM-R+ is central to our classification framework. Other models are used for comparison, and better performance is emphasized for XLM-R+. This model was chosen for its adequacy in processing Urdu text data and improving the performance of the event categorization tasks. The overview of the proposed research is provided in the following Fig. 2. The following are outlined in the simple form of the steps. For categorization, each event class is assigned with generally accepted label.

A. DATASET

A comprehensive dataset was developed specifically for this research to classify Urdu-language sentences into multiple event categories. Unlike datasets focused on sentiment analysis or product reviews, this dataset emphasizes multiclass event classification. To gather the data, we implemented a PHP-based web scraper to extract data from popular

TABLE 2. Summary of literature.

Article reference	Dataset	Methodology	Results	Limitations
[4]	Amazon reviews dataset	Word embeddings and CNNs for sentiment classification	High accuracy in sentiment classification of Amazon reviews	Limited to Amazon reviews, may not generalize to other review types
[13]	Khasi text data	Survey of sentiment analysis and emotion detection techniques	A comprehensive review of techniques for Khasi text	The survey lacks implementation details and specific results
[5]	Roman Urdu micro-text	NLP, machine learning, and ensemble techniques for Cyberbullying detection	Successful identification of cyberbullying patterns	Limited to Roman Urdu, requires adaptation for other languages
[19]	Urdu text dataset	Transfer learning for hate speech detection	High accuracy in detecting hate speech in Urdu	Requires large annotated datasets for effective transfer learning
[23]	Handwritten Urdu numerals dataset	Deep learning techniques for numeral recognition and classification	High accuracy in recognizing and classifying handwritten Urdu numerals	Dataset size and diversity limitations
[24]	Various datasets for semi-supervised learning	Introduction and methodologies of semi-supervised learning	Effective strategies for leveraging unlabeled data	The general overview may lack specific application details
[27]	Online Bangla web text corpus	TF-IDF feature extraction for document categorization	Effective categorization of Bangla web text documents	Limited by the use of a single feature extraction method
[29]	Scene text images	Hybrid CNN models for text recognition and classification	High performance in recognizing and classifying scene-based text	Dataset diversity and size limitations
[30]	Various text classification datasets	Comparative study of naive Bayes classifiers with improvements	Enhanced performance of improved naive Bayes classifiers	Results dependent on dataset characteristics
[31]	Urdu news articles dataset	Machine learning techniques for news classification	High accuracy in classifying Urdu news articles	Dataset limitations and the need for a more comprehensive evaluation
[34]	Printed Urdu script dataset	Document analysis and recognition techniques	Effective recognition of printed Urdu script	The older methodology may require updates with modern techniques
[36]	Twitter text	Sentiment and emotion analysis using machine learning techniques	Effective sentiment and emotion classification from Twitter text	Limited to the Twitter dataset, may not generalize to other text types
[37]	Annotated Urdu corpus	Semantic tagging and multi-target classification methods	Successful semantic tagging of Urdu text	Limited by the quality and representativeness of the annotated corpus
[52]	Newly developed benchmark dataset	N-gram features, fastText embeddings, Logistic Regression	Highest F1 score of 82.05% using LR with n-gram features	Limited to Urdu text only, may not generalize
[53]	6,000 labeled sentences	LSTM, BiLSTM-ATT, CNN, Hybrid CNN-LSTM	BiLSTM-ATT achieved 77.9% accuracy and 72.7% F1 score	N/A
[54]	1,372 multimodal expressions	CNNs, LSTMs, Decision-level and Feature-level fusion methods	Accuracy improved from 84.32% to 95.35% with multimodal features	Specific to multimodal data, complex integration
[55]	Unique dataset for cross-domain analysis (cricket and football)	Deep learning and machine learning with n-grams and word embeddings	Accuracy 77%, Precision 83%, Recall 68%, F1 score 75%	Requires extensive manual annotation for new domains

TABLE 2. ***(Continued.)*** **Summary of literature.**

[56]	Multiple datasets	Advanced word embeddings, LSTM, and GRU models	F-scores up to 0.98	Challenges with rare words and complex syntax handling
[57]	Urdu Nastalique Emotions Dataset (UNED)	Deep learning models for emotion classification	F1 score of 85% on sentence-based and 50% on paragraph-based evaluations	Limited resources for Urdu NLP, specific to context-aware models
[58]	Limited dataset of Urdu text	Various deep learning techniques including CNNs and LSTM networks for sentiment classification	Significant improvement in sentiment classification accuracy using cognitive relationship-based features	Limited dataset size and potential overfitting due to complex feature engineering
[59]	Urdu text dataset	Deep learning models integrated with attention mechanisms, combining CNNs and LSTM networks	High accuracy and precision in detecting abusive language	Quality and diversity of the dataset, need for validation on larger and varied data sources
[60]	Annotated dataset of Urdu tweets	Machine learning models including SVMs, random forests, logistic regression, naive Bayes, decision trees, CNNs, and RNNs with data preprocessing (normalization, tokenization, TF-IDF)	CNNs achieved the highest accuracy at 99% in detecting fake news	Dependency on the quality of annotated data and handling nuanced linguistic features of Urdu
[61]	Urdu-translated IMDB dataset	Deep learning models (1D-CNN, LSTM, Multilingual-MiniLM-L12-H384) for sentiment classification	The transformer model showed the highest accuracy at 89.36%	Reliance on a single dataset and the need for more comprehensive data
[62]	Dataset of Urdu tweets	Preprocessing, tokenization, and CNN model for classifying tweets	High accuracy in detecting violent content	Dataset size and potential bias in manually labeled data
[63]	Dataset of threatening and non-threatening texts	Fine-tuning Urdu-BERT model with contextual embeddings	Significant improvement in identifying threatening content	Dependence on dataset quality and representativeness, need for diverse training data
[64]	Labeled Urdu sentiment data	Multilingual BERT model for feature extraction and classification into multiple sentiment categories	High accuracy and robustness in sentiment classification	Limited availability of labeled Urdu sentiment data, high computational resources required
[65]	Large-scale multilingual datasets	Transformer-based multilingual masked language model (XLM-R)	Significant performance gains over mBERT, especially in low-resource languages	High computational cost and complexity in training large-scale models
[66]	Dataset of Urdu tweets	Hybrid approaches combining CNN, LSTM models, and fine-tuning transformer models	High accuracy and F1 scores in detecting sarcasm	Need for more diverse and comprehensive datasets, high computational resources required.
[67]	Various Urdu text datasets	Semantic and embedding models for classification	High precision and recall in detecting offensive language	Need for larger and more diverse datasets, potential bias in manually annotated data

Urdu-language media outlets, including Geo News, Samaa News, Dawn News, Express News, and Urdu Blogs. This raw dataset was stored in a MariaDB database for further processing.

1) DATA ACCUMULATION PROCESS

Data was collected using a custom-built web scraper that targeted the aforementioned platforms in the initial six months of the research. We did not use any API keys since most of

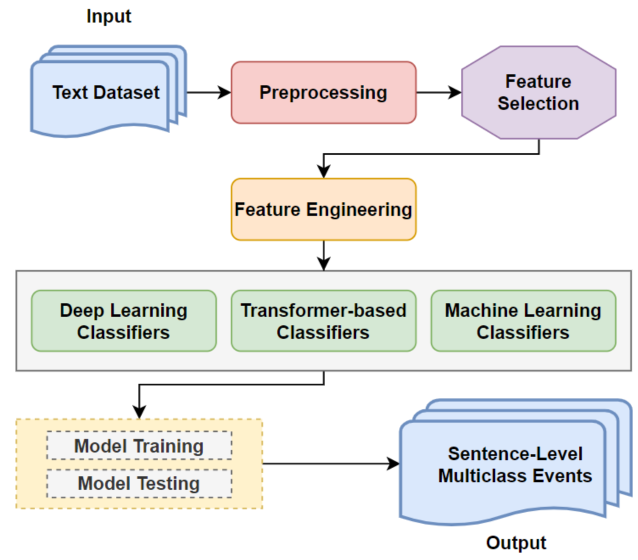

FIGURE 2. Overview of the proposed methodology.

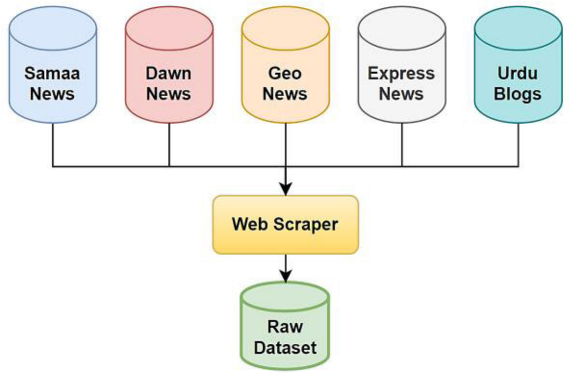

FIGURE 3. Dataset collection from news channels and blogs.

FIGURE 4. (a) News highlights from Samaa, News, (b) News highlights from Express-News.

these websites do not provide open-access APIs. Instead, the scraper crawled the public web pages of these platforms. The web scraper was programmed to extract text that matched predefined search queries related to specific event types. The search queries used were keywords in Urdu related to these domains. Since the data was scraped from public news articles, no API permissions were required. However, we adhered to ethical data scraping practices by respecting the terms of service of each platform. The credibility of the dataset is ensured as the data sources are established and reputable media outlets that report on a wide range of events. A sample of news from these sites is given in Fig. 4 while Fig. 3 shows the data collection from blogs and news channels.

2) DATA CHARACTERISTICS

The dataset includes a total of 103,771 sentences, distributed across 16 event categories. These categories cover a broad spectrum of real-world events. Table 3 outlines the dataset properties and the number of sentences associated with each event category. Some categories, such as politics and inflation, have a significantly higher number of sentences, contributing to an imbalanced dataset.

Each sentence was manually labeled according to the event it describes. For example, sentences reporting on a terrorist attack were labeled as "دہشت گرد حملہ," while those discussing politics were labeled as "سیاست."

3) DATA PREPROCESSING AND STRATIFICATION

To ensure that the dataset was ready for model training, several preprocessing steps were applied and explained in the next section. These included tokenization, stop-word removal, and sentence cleaning to remove any noise in the data. Since the dataset was imbalanced, we employed a stratified random split to ensure that all event categories were proportionally represented in both the training and test sets. This technique helps mitigate bias and ensures that the models are exposed to a balanced distribution of event types during training.

4) DATASET ENHANCEMENTS

In addition to creating a new dataset, we enhanced an existing one by adding more event categories to increase its utility for multiclass event classification tasks. This enriched dataset can be valuable for future research in areas related to Urdu-language event classification. The dataset has already been published is these researches [11], [40].

B. DATASET PRE-PROCESSING

The dataset was preprocessed to ensure it was clean, consistent, and ready for effective classification. This section outlines the key steps undertaken to preprocess the data before model training and testing, ensuring high-quality input

TABLE 3. Number of sentences in a dataset.

Dataset Properties			
Events		**Labels**	**Number of Sentences**
Terrorist attack	دہشت گرد حملہ	1	2726
National News	قومی	2	4098
Sports	کھیل	3	19820
Entertainment	تفریح	4	9460
Politics	سیاست	5	32346
Fraud and corruption	فراڈ اور کرپشن	6	9162
Sexual assault	جنسی حملہ	7	1503
Weather	موسم	8	1440
Accidents	حادثات	9	1520
Forces	افواج	10	8447
Inflation	مہنگائی	11	23863
Murder and death	قتل اور موت	12	6694
Education	تعلیم	13	1482
Law and order	امن و امان	14	14973
Social media	سوشل میڈیا	15	7999
Earthquakes	زلزلے	16	3238

for machine learning, deep learning, and transformer-based models. The preprocessing steps are outlined in Fig. 5.

1) TOKENIZATION

Tokenization is a very important initial stage of text preparation for analysis, which is especially needed for Urdu, a language with a highly extensive system of word formation. To produce the final result each sentence was preprocessed by tokenizing the words using the Natural Language Toolkit (NLTK). Now this helps the classification models to deal with various forms of words that may seem different by context. For instance, it is possible for the Urdu word "بہار" to appear in several different grammatical constructs, and tokenization makes sure to deal with that. This step allows the model to pay attention to the skeletal structure of each given sentence. Samples of tokenized sentences are presented in Table 4 below.

2) STOP WORDS ELIMINATION

To reduce dimensionality and improve the model's ability to focus on significant terms, we removed stop words from the dataset. Stop words are common words (like articles, prepositions, and conjunctions) that do not carry significant meaning for event classification. By eliminating these non-informative words, we reduced the noise in the dataset, which allows the model to focus on the event-specific content. The list of stop words used in this experiment is provided in Table 5.

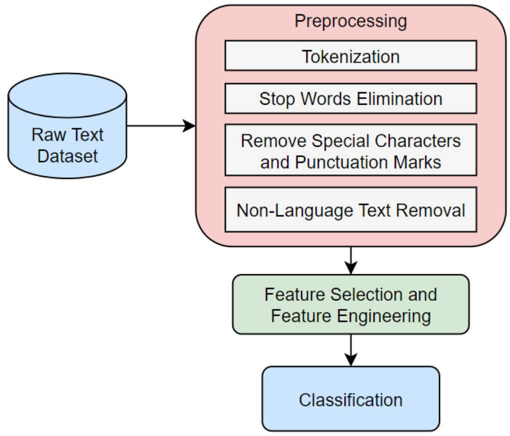

FIGURE 5. Pre-processing techniques.

3) REMOVE SPECIAL CHARACTERS AND PUNCTUATION MARKS

We removed special characters (punctuation, URLs, hashtags, and digits) to further reduce noise in the dataset. The presence of such characters often introduces unnecessary complexity for classification tasks. Removing these elements ensures that the models are trained on clean, meaningful text, improving the overall accuracy of the classification. The punctuation marks removed from the Urdu text are shown in Table 6.

All the special symbols and punctuation marks are removed using regular expressions. An example with and without punctuation is given in Table 7.

4) NON-LANGUAGE TEXT REMOVAL

To ensure the dataset consists solely of Urdu text, we removed any non-language elements (e.g., English text, numerical data, and metadata). This ensures that only relevant textual data is used for classification. Removing non-language text could mislead the classifier, and so leaving it out gives a clearer, more target-oriented data set for use in training. A sample of the use of non-language words in sentences with and without them is given in Table 8.

The data was divided into training and testing sets using a random method. This is the reason why, to have a proportionate distribution of every event type, stratified sampling was used. This method avoids class imbalance and ensures that the models of classification have a balanced type of events essential in the classification of events.

Specifically, preprocessing for the Urdu language, which is tokenization, stop-word elimination, and non-language text removal. Through data cleansing and feature selection, it is possible to prevent the distortion of the resulting classification model with irrelevant arrays, and thus enhance detection of event-related features in classification models. The pseudocode for the text preprocessing function is explained as follows in Table 9.

TABLE 4. Tokenization of urdu text.

Urdu Sentence	ٹرمپ کے بیان کے بعد ریکارڈ کی تصحیح کیلئے										
Tokens	کیلئے	تصحیح	کی	ریکارڈ	کے	بیان	کے	ٹرمپ			
Urdu Sentence	ایشین ہاکی کپ کے فائنل میچ میں بھارت										
Tokens	بھارت	میں	میچ	فائنل	کے	کپ	ہاکی	ایشین			
Urdu Sentence	زلزلے سے پہلے ہمارے پاس اسکے نشان دہی کرنے والے آلات										
Tokens	آلات	والے	کرنے	دہی	نشان	اسکے	پاس	ہمارے	پہلے	سے	زلزلے
Urdu Sentence	زلزلے سے متعلق تحقیقات اہم ہیں										
Tokens	ہیں	اہم	تحقیقات	متعلق	سے	زلزلے					

TABLE 5. List of stop words.

اور	خو	گی	رہی	ھگر	تک	طرف	پر
ایک	ہیں	ہو	رہے	ھیں	کی	در	اچھی
تھے	دی	گے	لگیں	ہے	ہوا	ہر	اچھے
رکھ	کیوں	کوئی	والے	یہ	کے	چلے	تو
پھر	کررہی	لگی	کررہے	رکھتی	چلو	کہ	ہیں

TABLE 6. Punctuation marks in the urdu language.

English	Marks	Urdu
Full Stop or Period	.	ختمہ
Question Mark	?	سوالیہ نشان
Exclamation Mark	!	فجائیہ
Comma	,	سکتہ
Inverted Commas	" "/' '	واوین
Colon	:	رابطہ
Semicolon	;	وقفہ
Dash	_	نشان ربط
Apostrophe	'	علامت حذف
Hyphen	-	نشان الحاق
Round Brackets	( )	ہلالی قوسین
Curly Brackets	{ }	گھونگرا بریکیٹ
Square Brackets	[ ]	مربع بریکیٹس
Ellipsis	...	نقطے
Slash or Oblique	/	علامت تغیر

TABLE 7. Sentence with and without special symbols and punctuation marks.

Sentence before removal of punctuation marks and symbols
#"، معلم نے پوچھا۔"زلزلے کے بعد بچوں کو چھوڑنا درست ہوگا؟
Sentence after removal of punctuation marks and symbols
زلزلے کے بعد بچوں کو چھوڑنا درست ہوگا معلم نے پوچھا

TABLE 8. Sentence with and without non-language words.

Sentence with non-language words
پاکستان اسٹاک ایکسچینج 100انڈیکس میں2 ہزار350پوائنٹس کی تیزی (Samaa News)
Sentence without non-language words
پاکستان اسٹاک ایکسچینج انڈیکس میں ہزارپوائنٹس کی تیزی

C. FEATURE SELECTION AND FEATURE ENGINEERING

The selection and generation of features provide the foundation of any text classification problem since they influence the performance of the classification models. Because of the aspects of the Urdu Language such as Language Type, SOV, morphological complexity, and Syntactic complexity, it is crucial to use special techniques to enhance event classification. This section briefly describes all the methods included in the feature selection and feature engineering process in our proposed system and explains why some features were selected for the processing of the Urdu language, along with the fact that these methods are new to the best of the authors' knowledge. The feature selection and feature engineering techniques are depicted in the following Fig. 6.

TABLE 9. Pseudo code for text pre-processing.

Pseudo code 1: For text pre-processing
def preprocess_text(text):
Text Cleaning
text = remove_punctuations(text)
text = remove_urls(text)
text = remove_special_chars(text)
text = remove_extrahitespaces(text)
Tokenization
tokens = tokenize_text(text)
Stop Words Removal
stop_words = get_stop_words()
tokens = remove_stop_words(tokens, stop_words)

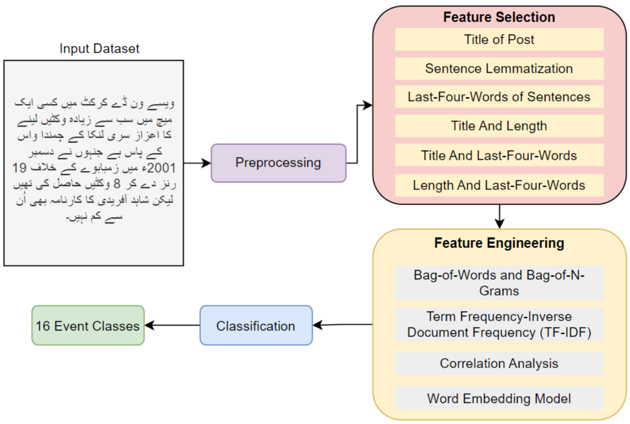

FIGURE 6. Techniques for feature selection and feature engineering.

1) FEATURE SELECTION

The choice of features is well done to improve the performance of the classification models since irrelevant features will affect the model's performance. Fig. 6 shows the details of selected features and feature engineering techniques used in this research. After these techniques, selected features are used successfully for classification. In our study, we focused on selecting features that best represent Urdu sentence structure and assist in identifying the occurrence of specific events. The selected features are based on linguistic and semantic properties of Urdu, and we propose innovative methods for sentence-level classification. The following features were used in our experiment, each contributing uniquely to the model's ability to classify events:

a: TITLE OF POST

The title of a news post typically provides a concise summary of the event, making it a central point of focus. In Urdu, titles are often structured to provide essential information upfront. We chose to use the title as a feature for classification since it encapsulates the main idea of the news event, helping the model to identify relevant categories. For instance, in the dataset sourced from Samaa News and Geo News, titles like "انٹرٹینمنٹ" (Entertainment), "پاکستان" (Pakistan), and "بین الاقوامی" (International) help guide the classifier to relevant categories. This feature plays a significant role in sentence classification by helping the model focus on the primary content of the text.

b: SENTENCE LEMMATIZATION

Lemmatization reduces the various inflected forms of words into their base form. In a morphologically rich language like Urdu, words may have several variations depending on context. We used word probability to determine the importance of sentences in our dataset. The probability of a word $P(w)$ is calculated by dividing its frequency $f(w)$ by the total number of words N in the sentence [38]. Probability is given in equation 1.

$$P(w) = \frac{f(w)}{N} \tag{1}$$

The weight of a sentence is determined by the average probability of its words, ensuring that high-probability words contribute more to the sentence's overall importance given in Equation 8.

$$g(S_j) = \frac{\Sigma W_j \in S_j P(W_j)}{|(w_i|w_i \in S_j)} \tag{2}$$

In equation 2, g (S_j) represents the weight of the sentence. Additionally, the first four words of a sentence are given double weight.

c: LAST-FOUR-WORDS OF SENTENCES

Urdu follows a Subject-Object-Verb (SOV) structure, where verbs that indicate events usually appear at the end of a sentence. Therefore, the last four words are particularly crucial in identifying event-based sentences. This feature improves the classification of events like earthquakes, politics, or sports, where the occurrence is typically signaled by verbs at the end of the sentence. For example, in the sentence "بھارتایشین ہاکی کپ کے فائنل میچ میں" (In the Asian Hockey Cup final match against India), the word "میچ" (match) appears in the last four words, which is a vital feature for classifying sports-related events. The inclusion of this feature is highly relevant to event classification in Urdu due to its syntactic structure [39].

d: TITLE AND LENGTH

In addition to the title, we considered the length of the sentence as a feature. Sentence length often varies depending on the event type; for instance, political events tend to have longer sentences, while weather updates might be shorter. Adding a title and using context given by the length of the sentence gives the model more details to distinguish and separate major and minor events, and brief updates.

e: TITLE AND LAST-FOUR-WORDS

This is provided by joining the title of the work with the final four words of the whole sentence. With an event indicator in the model title and key verb-related information in the last four words, the model improves event context identification. It turns out that this feature is most useful when they are two similar sentences but they refer to two different categories of events.

f: LENGTH AND LAST-FOUR-WORDS

We also merged the first feature group, that is, sentence length with the last four words to improve event classification. The length of a specific sentence relates to the degree of the event's complexity; the final four words denote the event information. Together, this approach is effective at providing an overall classification of complicated and diverse events that may belong to more than one type rather than simple and narrow ones.

2) FEATURE ENGINEERING

Feature engineering is the process that helps to transfer the raw text data into a format that would be understandable by the machine learning algorithms. In this paper, several methodologies were employed to create meaningful features from the raw Urdu text data. Since there were no Urdu BERT or RoBERTa embeddings available, or for the fact that utilizing such complex models involved computational constraints, we decided to initialize with relatively straightforward methods such as BoW or TF-IDF and then go for the transformer architectures.

a: BAG-OF-WORDS AND BAG-OF-N-GRAMS

Of all the text data pre-processing techniques, the BoW technique is easy to implement to convert text data into a numerical form. How does it operate? This method creates a vocabulary of distinct words across the whole phrases and tokenizes them. This frequency count of each word from the sentences is turned into feature vectors and is presented below: The BoW approach is best used as a baseline technique for our classification tasks because of the simple counts of word occurrence in a text that it provides [45].

As per the BoW technique, we can calculate the BoW value of a scale factor 'w' in a given sentence, wherein Count (w, s) represents the frequency of the particular word 'w' by using equation 3.

$$BoW\ (w, s) = Count(w, s) \tag{3}$$

Besides the implementation of BoW, Bag-of-N-Grams are also incorporated for collecting context information. N-grams are the continuous combination of words up to N in numbers. We extend the representation beyond individual words by considering these N-gram sequences N-grams are denied as contiguous sequences of N words. To compute the Bag-of-N-Grams value for an N-gram 'w' in a sentence; use the BoN equation, where Count (w, s) reflects the frequency of the N-gram 'w' in the sentence' given in equation 4.

$$BoN\ (w, s) = Count(w, s) \tag{4}$$

We use feature engineering techniques to convert Urdu sentences into numerical representations, allowing us to execute effective sentence-level classification tasks.

b: TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF)

While BoW simply counts word occurrences, TF-IDF assigns weights to words based on their importance in the document relative to their frequency across all documents. This helps reduce the impact of commonly occurring words like stop words, giving more weight to less frequent but more informative words. We also use this method to calculate the weight based on the frequency of the word in the text and the number of times the word appears in the data [27]. In this research, the irrelevant terms were eliminated using TF–IDF [3]:

$$TD - IDF\left(t_i, d_j\right) = TF\left(t_i, d_j\right) \times \log\frac{N}{DF\ (t_i)} \tag{5}$$

The given equation 5 represents the calculation of the TF-IDF of a term. t_i in a sentence d_j. TF measures how frequently a term appears in a document, while $DF\ (t_i)$ Counts the number of sentences in the corpus that contain the term t_i. If we multiply the two terms above we get the value of TF-IDF, which shows the significance of a term about a sentence. There is N, which is the total number of sentences in the collection and is used to scale the TF-IDF score across the corpus.

c: CORRELATION ANALYSIS

Spearman correlation coefficient test used for comparing two variables having continuous type of data. This coefficient used in computing dependence between two variables where both are in continuous form. The value posted for coefficient ranges from -1 through $+1$, meaning negative correlation equals -1 while positive correlation is $+1$ and a score of 0 shows no linear relationship.

We use the following formula to obtain Pearson's coefficient r [41] given in equation 6:

$$r = \frac{(x_1 - \bar{x}_1 L)\,(x_2 - \bar{x}_2 L)}{(|x_1|\,|x_2|)} \tag{6}$$

In equation 6, $\bar{x}_1$ and $\bar{x}_2$ signifies the mean of the vector x_1 and x_2 correspondingly. Vector of 1s is represented by L and $|x|$ represents the magnitudes of vectors x. We use this correlation coefficient to quantitatively investigate the linear dependency between variables and get insights into their relationship.

d: WORD EMBEDDING MODEL

We use the Word2Vec model to represent each word in the numerical text and generate feature vectors for classifiers. This method generates dense vectors with real values that capture the word's semantic, contextual, and syntactical meaning. It also assigns weighted values to similar terms.

The Word2Vec model has two learning algorithms: continuous bag-of-words and continuous skip-gram. The training complexity architecture of the skip-gram model, which we employ as the default model, is described by equation 7 [42]:

$$Q = C_x(O + Ox \log_2 (V)) \tag{7}$$

In equation 7, C stands for the greatest distance between words, O stands for word representations and V stands for dimensionality. Given the current word, we can anticipate the adjacent words using this architecture. For all word positions, the weight matrix between the input and projection layers is shared. The vocabulary for our task is defined as vocab $= \{t_i \mid i \in 1 \ldots N\}$, where N is the vector size, and sentences are represented as $s_i = <w_i \ldots w_j>$. The vector representation of our technique denoted as $w2v\,(t_i)$. We use TF-IDF weighting with Word2Vec to integrate weights given in equations 8, 9, 10, and 11.

$$R\,(d_i) = \sum\nolimits_t w2v\,(t_i) \quad where\ t \in d_i \tag{8}$$

$$w_R\,(d_i) = \lambda w_t w2v\,(t_i) \tag{9}$$

Here

$$w_t = tf - idfweightoft \tag{10}$$

$$C\,(d_i) = concatenate(tf - idf\,(d_i w\_R\,(d_i)) \tag{11}$$

We sum the Word2Vec representations, apply TF-IDF weights, and then concatenate the TF-IDF vector with the Word2Vec representation to form a combined feature representation. An example of feature vectors generated with this method is given in Table 10.

TABLE 10. Similarity scores of the Word2Vec model.

Word	Most Similar Words and Similarity Score
مرد	حضرات (0.8628), مردوں (0.8199), عورت (0.7592), مخنث (0.7465), خواتین (0.7432), حجاب (0.7368), لیےبہت (0.7259), ماڈلز (0.7218), مالدار (0.7104), کھو 0.6958))
عورت	لڑکیاں (0.8762), سیکس (0.8697), وراثت (0.8684), لڑکا (0.8666), ہنس (0.8632), چھاڑ (0.8606), بڑھاتا (0.8606), ریکٹ (0.8601), ساونت (0.8572), چوٹ 0.8562))
فلم	فلموں (0.7724), فلمیں (0.7627), ٹریلر (0.7001), بینر (0.6798), ویڈنگ (0.6757), دیوگن (0.6709), جوانی (0.6597), طیفا (0.6589), اکشے (0.6565), کھنہ 0.6548))

Table 11 shows the Word2Vec model used to generate word embeddings with a vector size of 128 and a window size of 5, creating a vocabulary size of 48,484. The tokenized sentences were converted into sequences using Keras's Tokenizer and subsequently padded to ensure uniform input length for the neural network. An embedding matrix was constructed where each word index was mapped to its corresponding Word2Vec vector, and the labels were encoded into a categorical format using Keras utilities.

TABLE 11. Implementation of the Word2Vec model for classification.

Parameter/Step	Value/Description
Word2Vec Model	vector_size=128 window=5 workers=10 min_count=1
Vocabulary Size	48484
Embedding Dimensions	128
Maximum Sentence Length	99
Tokenization	Keras Tokenizer
Label Encoding	to_categorical(list(df_w2v.label), dtype="int64")

D. EVENT CLASSIFICATION

Deep learning is a part of Machine learning in which multiple layers of neural networks can learn hierarchical representations from data on their own. Finally, deep learning models such as CNN, RNN, and LSTM-RNN do not need feature engineering as conventional machine learning models do since they can deal with big data with unstructured text data. These models are implemented to parse and learn about the raw text so they are especially useful in the NLP process, for example, in event classification. In this research deep learning models were used for the same reason they are likely used in many applications: to classify Urdu sentences given their complex patterns. For instance, LSTM-RNN networks were applied because the models are capable of remembering the context of a sentence over an extended sequence, this comes in handy when capturing event-related information scattered across sentences.

1) SINGLE-LAYER MULTISIZE FILTERS CONVOLUTIONAL NEURAL NETWORK

For the research also, shown in Fig. 8, we employ a Single-layer Multisize Filters Convolutional Neural Network (SMFCNN). In the SMFCNN, we employ filters of different sizes to produce feature maps for each filter size [10]. The next step involves 1-max pooling which is done to each of the feature maps. The feature map results are concatenated to come up with a Feature vector, which is the input for the 'penultimate layer'. The last layer of the network is softmax layer is used to classify the sentences into one out of sixteen events classes. To enhance the performance levels of the SMFCNN model, the hyperparameters regarding the number of filters, filter size, dropout rate, and batch sizes given in table 9 were determined. We employed three convolutional layers with 64, 128, and 256 filters, using ReLU activation functions. The model was trained using the Adam optimizer with a learning rate of 0.001, a batch size of 32, and a dropout rate of 0.5 over 20 epochs. The data was split into training and testing sets with an 80-20% split, and a softmax activation function was used in the output layer. Table 12 shows the details of each hyperparameter.

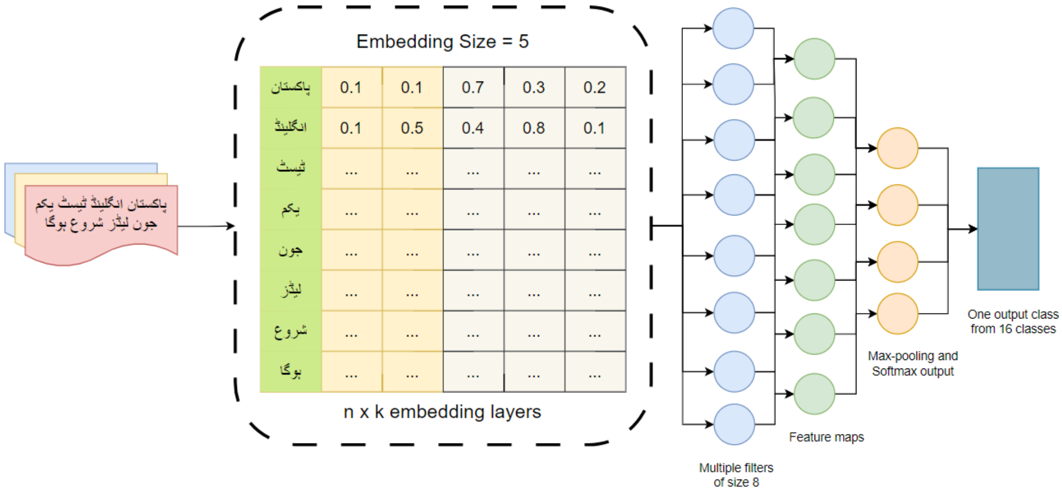

FIGURE 7. Layered architecture of Single-layer Multisize Filters Convolutional Neural Network (SMFCNN) [10].

Fig. 7 illustrates the architecture of the Single-layer Multisize Filters Convolutional Neural Network (SMFCNN), where input sentences are passed through embedding layers, multisize convolutional filters, and max-pooling to extract relevant features. The final output layer uses a softmax function to classify sentences into one of 16 event categories, supporting the experimental setup by demonstrating the model's capability to capture features for accurate event classification.

2) RECURRENT NEURAL NETWORK

We use Recurrent Neural Network (RNN) models within the neural network-based language model in our research.

The hidden layer state stores information from prior time steps and updates based on the current input and the previous hidden layer state, this architecture allows the model to incorporate short-term memory. The RNN's hidden state is indicated as h_t, is determined mathematically by the current input x_t and the prior hidden state h_{t-1}:

$$h_t = f(x_t, h_{t-1}) \tag{12}$$

In equation 12, x_t represents the input at a specific time step in the sequence (x_1, x_2 …x_T), and h0 is the initial hidden state which can be all zeros [43]. Based on this capability to remember information from the past, the RNN can therefore classify the current sentences under the context of Urdu language events. Below is the architecture of the model presented in the following Fig. 8.

The structure of the Recurrent Neural Network (RNN) is depicted in figure 8 below where the holding state depends on the present input and the previous holding state. This sequential structure enables it to capture information across time steps and this makes RNN suitable for capturing temporal dependencies in sentence-level event classification.

Table 12 shown below, is the setting of hyperparameters of our RNN model having two layers, with 128 units each and with tanh as the activation function. It used the Adam optimizer, with a learning rate of 0.003 and batch size of 32, including a dropout rate of 0.4. This model was trained using 20 epochs, with 80 percent train data and 20 percent test data; the output layer used softmax activation.

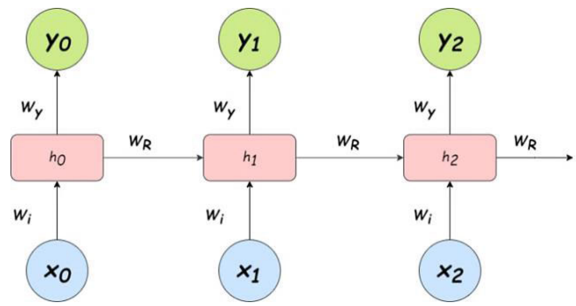

FIGURE 8. Recurrent Neural Network (RNN) architecture.

3) LONG SHORT-TERM MEMORY- RECURRENT NEURAL NETWORK

We used LSTM-RNN networks, which a recurrent neural networks with a powerful capability of handling and predicting data as well as data classification. In our experiment, we first construct an input layer in which brief text data is converted to an embedding matrix for our algorithm. According to Hassan and Mahmood, the number of inputs in the classification model defines the size of the input layer [43]. We compute the input gate value. i_t and the candidate value for the memory cell states at time t ($\tilde{c}_t$) given in equations 13 and 14:

$$i_t = \sigma(W_i x_t + U_t h_{t-1} + b_i) \tag{13}$$

$$\tilde{c}_t = \tanh(W_i x_t + U_i h_{t-1} + b_c) \tag{14}$$

Second, we compute the forget gate f_t value for the memory cells at time t given in equation 15:

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1}\right) + b_f \tag{15}$$

The new state of the memory cells c_t at time t using the derived values of the input gate i_t, forget gate f_t, and candidate state value c_t is intended as presented in equation 16:

$$c_t = i_t * \tilde{c}_t + f_t * c_{t-1} \tag{16}$$

Finally, using the new memory cell state, we compute the value of the output gate. o_t and, later, the output h_t Given in equations 17 and 18:

$$o_t = \sigma(W_o x_t + U_i h_{t-1}) \tag{17}$$

$$h_t = o_t * \tanh(c_t) \tag{18}$$

In this paper, LSTM-RNN has been designed with 4 interconnected layers as depicted in Fig. 9. This strategy enables us to predict the Urdu language events with the highest level of certainty because sequence data is processed rationally and the specific characteristics of text information are revealed.

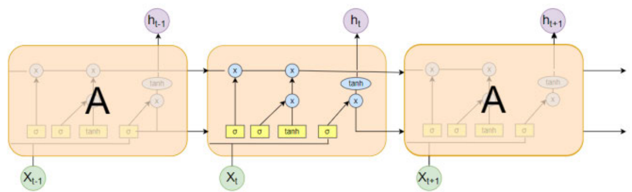

FIGURE 9. Long Short-Term Memory (LSTM-RNN) architecture with four interacting layers.

For hyperparameters, we had two layers with 128 nodes in each of them, and the tanh function as an activation function. The Adam optimizer with a learning rate of 0.001 was used, alongside a batch size of 32 and a dropout rate of 0.2. The training was conducted over 20 epochs with an 80-20% train-test split, and a softmax activation function was applied in the output layer given in Table 12.

TABLE 12. Hyperparameters of SMFCNN, RNN, AND LSTM-RNN.

Hyperparameters	CNN	RNN	LSTM-RNN
Number of Layers	3 convolutional layers	2 layers	2 layers
Number of Units/Filters	64, 128, 256 filters	128 units each	128 units each
Activation Function	ReLU	tanh	Tanh
Optimizer	Adam	Adam	Adam
Learning Rate	0.001	0.003	0.001
Batch Size	32	32	32
Dropout Rate	0.5	0.4	0.2
Epoch Size	20	20	20
Train-Test Split	80 - 20%	80 - 20%	80 - 20%
Output Activation	Softmax	Softmax	Softmax

4) MBERT (MULTILINGUAL BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS)

For the Urdu to various event classes, we use mBERT which has revolutionized natural language processing tasks as it provides deep bidirectional features where mBERT is conditioning on both left and right context in all layers giving the best performance [64].

The mBERT model is pre-trained on a massive amount of content and then fine-tuned in our study. mBERT model consists of 12 layers of 768 hidden and 12 self-attention heads by default. To avoid overfitting, the model employs a dropout rate of 0.1 The code below gives the complete model configuration: These hyperparameters include the AdamW optimizer with a learning rate of 2e-5, the batch size of 32, the number of epochs being 3 and the model we choose is the mBERT model. Hyperparameters are described in detail in Table 13.

5) ROBERTA (ROBUSTLY OPTIMIZED BERT APPROACH)

RoBERTa, an optimized version of BERT, is utilized to enhance the classification accuracy even better. RoBERTa improves upon BERT in terms of data used in training, larger batch size, and duration of training. This yields a stronger model that has better capability in acts of generalization [64].

Further still similar to the BERT model, the RoBERTa model consists of 12 layers with 12 heads and 768 hidden dimensions. We apply dropout with a dropout rate of 0.1 and finally finetune the facilitating model with AdamW optimizer at a learning rate of 1e-5. The training is done with a batch size of 32, across three epochs. The hyperparameters are summarized in Table 13.

6) XLM-R+ (CROSS-LINGUAL LANGUAGE MODEL-ROBERTA PLUS)

To leverage multilingual capabilities, we propose an advanced version of the XLM-R model, named XLM-R+. This model extends RoBERTa to handle multiple languages by training on a large multilingual corpus, making it particularly suitable for low-resource languages like Urdu, benefiting from cross-lingual transfer learning. XLM-R+ retains the robust architecture of the original XLM-R, with 12 layers, each containing 768 hidden units and 12 attention heads [65]. However, several enhancements have been incorporated to further improve its performance. These include a dynamic attention mechanism that adjusts focus based on context, enhanced dropout regularization with a variable rate to prevent overfitting more effectively, and an adaptive learning rate schedule that fine-tunes the learning rate during training to optimize convergence speed and model accuracy. Additionally, layer-wise fine-tuning is applied, with different learning rates for different layers, allowing lower layers to be fine-tuned less aggressively, thereby retaining pre-trained knowledge better. The model is also trained with an increased batch size of 64 and extended epochs to 5, which helps in better model generalization. Table 13 presents all the hyperparameters used in our research, demonstrating the advanced features of mBERT, RoBERTa, and our proposed XLM-R+ model. These enhancements are aimed at significantly improving the performance of event classification in low-resource languages like Urdu.

TABLE 13. Hyperparameters of MBERT, ROBERTA, AND XLM-R+.

Hyperparameter	mBERT	RoBERTa	XLM-R+
Number of Layers	12	12	12
Hidden Units	768	768	768
Attention Heads	12	12	12
Dropout Rate	0.1	0.1	Variable (0.1 - 0.3)
Learning Rate	2e-5	1e-5	Adaptive (start at 3e-5)
Batch Size	32	32	64
Epochs	3	3	5
Optimizer	AdamW	AdamW	AdamW
Dynamic Attention	N/A	N/A	Enabled
Layer-wise Fine-Tuning	N/A	N/A	Enabled

7) K-NEAREST NEIGHBORS (KNN)

To improve our estimation of the similarity between those sentences in the current study, we opt to employ the KNN classification technique. The Euclidean distance is used to calculate similarity, which is calculated as given in equation 19 [44]:

$$dist\,(p, q) = \sqrt{\sum\nolimits_{k=1}^{k} (p_k - q_k)^2} \tag{19}$$

In this case, p and q are two samples, and 'K' is the number of feature attributes. The KNN classification, which is based on the nearest neighbor approach, assigns a label to a testing sample based on the labels of its closest neighbors. The following equation 20 is a description of the KNN classification representation:

$$q'_x = \arg max \sum\nolimits_{p_k \in \propto(p_k)} f\,(q_x = L)\frac{1}{dist\,(p_x, p_k)} \tag{20}$$

$where\ L \in \{L_1, L_2$

The distance between the testing and training samples is represented by $dist\,(p_x, p_k)$ in Equation (2). The effectiveness of the algorithm relies on the choice of the value k and the similarity measure employed.

8) ADAPTIVE BOOSTING (ADABOOST)

We utilized AdaBoost, an ensemble learning algorithm widely employed for multiclass classification. In our experiment, AdaBoost combines multiple weak classifiers to create a robust and accurate classifier. AdaBoost's weak classifiers are decision trees with only one split, known as decision stumps.

The AdaBoost algorithm works by giving more weight to cases that are difficult to correctly classify while giving less weight to instances that have already been handled efficiently.

$$h\,(x) = \begin{cases} c & if\ x^j = 1 \\ -c & else \end{cases} \tag{21}$$

In equation 21, c ∈ {−1, 1} and in each iteration, the algorithm assigns greater weights to misclassified examples to enhance the classifier's performance in subsequent iterations [45].

9) SUPPORT VECTOR MACHINE (SVM)

Support vector machine is a strong machine learning technique that learns a hyperplane with n dimensions to properly classify data. We employ, in particular, the SVM with a polynomial kernel, which allows for the examination of polynomial degree transformations of the described input data and enhances classification performance in the context of events underlying the Urdu language.

$$K\,(xi, xj\,) = \left\{x_i^T xj + c\right\}^d \tag{22}$$

From equation 22, *xi and xj* are the input space vectors, c is a tuning factor that decides the trade-off between the highest and lowest order of the polynomial, and d is the order of the polynomial. Another kernel used by SVM is the Radial Basis Function (RBF) kernel [46]. It is an external function with real values changing as the distance from the origin increases. It can be defined as:

$$K\,(xi, xj\,) = \exp(-\gamma\,(xi - xj)^2)\,for\ \gamma > 0 \tag{23}$$

In equation 23, the value of γ set as $1/2\sigma^2$, where σ^2 is the variance of the input data. SVM also uses the sigmoid kernel function, which is defined as:

$$K\,(xi, xj\,) = \tanh(ax_i^T xj + b) \tag{24}$$

In equation 24, ax_i^T is the scaling parameter for the input data, while b is the shifting value that controls the mapping threshold.

10) LOGISTIC REGRESSION

LR is a widely used approach for binary classification, but it can also be extended to handle multiclass classification problems, which is the focus of our experiment. In our research, LR learns a set of weights for each class and predicts the likelihood of each class for a given input in the context of multiclass LR. We believe that the characteristics are affected not just by the observation x, but also by the potential output class c. As a result, instead of f_i or $f_{i(cx)}$, we designate the features as $f_{i(cx)}$, where f_i (c, x), represents feature i from class c assigned to the input x [47] given in equation 25.

$$P\,(c\,|\,x) = \frac{\exp \sum_{i=1}^{N} w_i.f_i(c, x))}{\sum_{c' \in C} \exp(\sum_{i=1}^{N} w_i.f_{i(cx)})} \tag{25}$$

The class with the highest probability is selected as the predicted class. LR assumes a linear relationship between the features and the output variable to make accurate classifications for Urdu language events.

11) DECISION TREE

The decision tree employs nodes to represent attributes of an example and their significance in the classification process, while the leaf nodes represent different classes [48]. It is a flowchart-type structure with an error rate and accuracy rate calculated with equations 26 and 27:

$$ER = \frac{Total\ number\ of\ multiclassified\ events}{Total\ number\ of\ data\ points} \tag{26}$$

$$Accuracy\ Rate = Error\ Rate - 1 \tag{27}$$

One of the advantages of using DT for our experiment is its ease of interpretation, allowing us to gain valuable insights into the classification process for Urdu language events.

12) EXTREME GRADIENT BOOSTING (XGBOOST)

XGBoost is an ensemble algorithm that is based on decision trees and utilizes the gradient-boosting framework. To create predictions in our research, we use XGBoost, each tree learns from the residuals or errors of the preceding trees. The predicted output of XGBoost is calculated by aggregating the outputs of all individual trees. This iterative strategy allows XGBoost to continuously improve its predictions by building on the knowledge of earlier trees [49].

$$y_i = \sum_{k=1}^{n} fk\ (x_i), \quad fk \in F \tag{28}$$

In equation 28, F represents the tree space, fk represents the tree, $fk\ (x_i)$ represents the outcome of a tree k, and x_i represents the projected value of the instance. We acquired accurate classification results for Urdu language events using this method.

13) RANDOM FOREST

Random Forest is an ensemble approach that is built using decision trees. The decision tree, in this study, uses data attributes to split the data objects accurately. In our research, decisions are taken at each node of the decision tree to generate diverse groups, but the individuals within each group are similar. Mathematically the representation of a random forest is given in equation 29:

$$n_{ij} = w_l C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)} \tag{29}$$

where n_{ij} shows the importance of the j node in the tree, w is the weight and C is the impurity value [50]. The ultimate classification result is established by a majority vote among the ensemble of decision trees' decisions. This method allows us to effectively classify Urdu language occurrences by using the strength of several decision trees.

14) MULTI-LAYER PERCEPTRON (MLP)

The Multi-Layer Perceptron (MLP) classifier is a form of neural network. Backpropagation is used by the MLP classifier to optimize the loss when predicting samples. A sparse input vector is coupled to a fully connected layer of 100 neurons in our research. The basic architecture of MLP is given in Fig. 10. This layer is followed by another fully linked layer with 16 output neurons reflecting the various types of Urdu language events. To train the model for multi-class classification, we utilize the Adam optimizer [37] with a learning rate of 0.001, minimizing the softmax loss function given in equations 30 and 31.

$$\theta_{n+1} = \theta_n - (\eta \frac{\beta_1}{\sqrt{\hat{v}_n + \epsilon}} \hat{m}_n + d) \tag{30}$$

$$d = \Delta\theta_{n-1} * sign\left(\nabla_{\theta_n} J\ (\theta)\right) * (1 - \beta_1) \tag{31}$$

where $\hat{m}_n$ and $\hat{v}_n$ are Adam parameters. $\Delta\theta_{n-1}$ shows the last updated step [51].

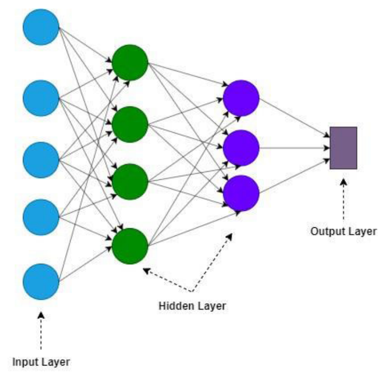

FIGURE 10. Basic architecture of multi-layer perceptron.

This configuration allows us to effectively classify Urdu language events using the MLP classifier. Table 16 shows the details of hyperparameters used for ML models in our research.

Machine learning models, such as SVM and Random Forest, provided a baseline for performance, particularly in terms of interpretability. Deep learning models, like LSTM-RNN and CNN, demonstrated superior performance on more complex patterns within the data, especially when sentence structure played a critical role. Additionally, transformer-based models such as mBERT, RoBERTa, and XLM-R+ built on these approaches by incorporating attention mechanisms to better handle the complexity of language, improving classification performance through a contextualized understanding of the text. The utilization of machine learning, deep learning, and transformer models for this study gave a unique attempt, taking the best of each model to achieve better classification accuracy for the Urdu event at the sentence level.

IV. RESULTS AND DISCUSSIONS

This section presents the deep learning, transformer-based, and machine learning classifier results for the classification of 16 events. The performance of the experiments is described comprehensively with the training and testing characteristics of all the models applied in this work.

TABLE 14. Hyperparameters of machine learning models used.

Hyperparameters	KNN	AdaBoost	SVM	Logistic Regression	Decision Tree	XGBoost	Random Forest	MLP
Number of Layers	1	1	1	1	1	1	1	2 layers
Number of Units	5 neighbors	50 estimators	N/A	N/A	N/A	N/A	100 trees	100, 16 neurons
Activation Function	Euclidean Distance	Decision Stumps	Polynomial, RBF, Sigmoid	Sigmoid	N/A	N/A	N/A	ReLU, Softmax
Optimizer	N/A	N/A	N/A	Gradient Descent	N/A	N/A	N/A	Adam
Learning Rate	N/A	0.1	0.01	0.01	0.01	0.1	0.01	0.001
Batch Size	32	32	32	32	32	32	32	32
Dropout Rate	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.2
Epoch Size	100	50	50	50	50	50	50	20
Train-Test Split	80-20%	80-20%	80-20%	80-20%	80-20%	80-20%	80-20%	80-20%
Output Activation	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Softmax

A. EVALUATION METRICS

The findings of this research, the efficiency of the research is measured by employing the performance measuring metrics of accuracy, F1-score, precision, and recall [11]. Precision is used to calculate the proportion of accurately predicted events in a positive class out of all anticipated events given in Equation 32.

$$Precision = \frac{TP}{TP + FP} \tag{32}$$

The fraction of correctly classified events is measured by recall given in equation 33

$$Recall = \frac{TP}{TP + FN} \tag{33}$$

F1 score is the harmonic mean of the recall and precision values given in equation 34.

$$F1 = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \tag{34}$$

Accuracy compares the proportion of true predictions made by a model to the total number of predictions made.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{35}$$

where TP denotes True Positive, FP denotes False Positive, TN is True Negative, and FN denotes False Negative.

B. RESULTS OF CLASSIFIERS

In terms of deep learning classifiers, SMFCNN achieves an accuracy of 88.29%, while RNN and LSTM-RNN achieve 87.16% and 86.91%, respectively. Fig. 11 shows the accuracy of all three deep learning models.

Table 15 provides the precision, recall, F1 score, and accuracy values for each model. In our research, for our dataset, SMFCNN shows the highest accuracy of 88.29% with a precision of 95%, recall of 91%, and F1-score of 98%. The graphical representation of precision, recall, and F1-score is in Fig. 12.

Fig. 13 shows the training and validation accuracies of deep learning models. This also depicts the training and validation loss of deep learning models. The red line in plots shows the validation accuracy of each model at 20 epochs while the blue line shows the training accuracy of our dataset with 20 epochs. SMFCNN shows a lower training accuracy as compared to other models' training accuracy but gives the best validation accuracy. This model labels the sentences with the most accurate event labels. The second highest accuracy in deep learning models is achieved by the RNN classifier with an accuracy of 87%. LSTM-RNN shows the minimum validation loss with an accuracy of 86%. SMFCNN predicts positive sentence labels, it has a 0.9529 precision of being correct, demonstrating the highest precision among the three.

This model also has an impressive recall of 0.9106, which means it properly detects 0.9106 of all real positive sentence labels. The RNN classifier slightly outperforms SMFCNN with a recall of 0.9291. The F1 score is an informative metric when precision and recall are balanced. SMFCNN once again achieves the highest F1 score of 0.9830.

The behavior of SMFCNN in terms of training and validation accuracies is a captivating finding from the research. Despite having the best validation accuracy, SMFCNN has a lower training accuracy than its competitors [10]. This indicates a strong ability to generalize on previously unseen data, thereby lowering the risk of over-fitting. RNN ranks second in terms of validation accuracy. LSTM-RNN, on the other hand, despite being inaccurate, has the lowest validation loss. This means that its errors are, on average, less severe or confident than those of the other models.

The model's loss on the training data is known as the training loss, and its loss on the validation data is known as the validation loss. The model has been trained on the training data with 20 epochs. Fig. 13 (a) shows that as the number of epochs increases, the training accuracy increases

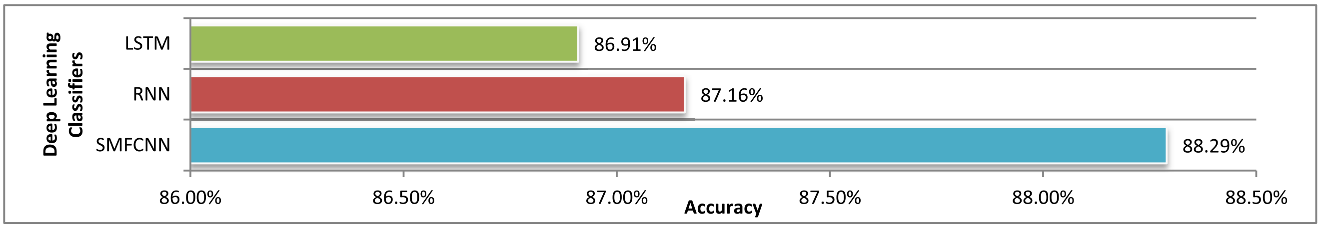

FIGURE 11. Accuracy of deep learning models.

TABLE 15. Results of deep learning models.

DL Classifiers	Precision	Recall	F1 Scores	Accuracy
SMFCNN	0.9529	0.9106	0.9830	0.8829
RNN	0.9021	0.9291	0.9267	0.8716
LSTM-RNN	0.8991	0.9011	0.8982	0.8691

as well. This is a result of the model gradually improving its ability to fit the training set of data. However, after a given 10 epochs, the validation accuracy begins to plateau. The model is beginning to overfit the training set of data. When a model learns the specifics of the training data too thoroughly, over-fitting occurs, and as a result, the model struggles to generalize to new data. The same is the case of the RNN model's accuracy given in Fig. 13 (c). The ideal number of epochs in this situation to train the model is 7. This is because, at this stage, both the training accuracy and the validation accuracy are continuously increasing. The accuracy of LSTM-RNN also increases given in Fig. 13 (e). The image also demonstrates that the training accuracy constantly outperforms the validation accuracy. This is because the validation accuracy is a more precise indicator of how well the model will function with new data. Fig 13 (b), (d), and (f) show the training and validation loss of SMFCNN, RNN, and LSTM-RNN respectively. As the number of epochs increases, the train loss decreases. This is a result of the model gradually improving its ability to fit the training set of data. However, after 7 to 10 epochs, the validation loss starts to increase. The model is beginning to overfit the training set of data. When a model learns the specifics of the training data too thoroughly, over-fitting occurs, and as a result, the model struggles to generalize to new data.

The results of our transformer-based models: mBERT, RoBERTa, and the proposed XLM-R+ are given in Table 16.

TABLE 16. Results of transformer-based models.

Model	Accuracy	Precision	Recall	F1-Score
mBERT	88.50%	87.80%	88.00%	87.90%
RoBERTa	89.20%	88.50%	88.70%	88.60%
XLM-R+	89.80%	88.60%	88.80%	88.70%

These results indicate that all three models perform well in our research of Urdu sentence classification, with XLM-R+ achieving the highest overall performance metrics with 89.80% accuracy. The RoBERTa model also shows strong performance of 89.20% accuracy, benefiting from the additional enhancements incorporated into its architecture. mBERT, while slightly trailing behind RoBERTa and XLM-R+ with an accuracy of 88.50%, still demonstrates robust performance, highlighting its effectiveness in capturing bidirectional contextual information. Fig. 14 shows the results of these models.

A random forest classifier is made up of several decision trees. However, when applied to our dataset, it achieved just 77.76% accuracy with a depth of 3 and a random state value of 1. With a leaf size of 32 and a neighbor value of 10, K-Neighbors achieved an accuracy of 84.41%. We used the Euclidean metric with no threshold in AdaBoost and achieved an accuracy of 78.35%. The model achieves the second-greatest accuracy of 87.12% by setting the random state value to zero in logistic regression.

The Decision Tree (DT) model represents attributes of an example and their value in classifying it, whereas the leaf nodes represent classes. We used grid search for classification and achieved an accuracy of 76.71%. Our model has 75.02% accuracy while utilizing XGBoost. The accuracy of the MLP classifier is the lowest at 69.22%. The SVM model has the highest accuracy of 87.92%. Fig. 15 depicts the accuracy of all machine learning models employed in this research work. Fig. 16 shows the accuracies of these ML models in our research.

Precision, recall, F1-scores, and accuracies are given in Table 17 and these values are shown graphically in Fig. 15, 16, and 17.

The graphical representation of precision, recall, and f1score of all the machine learning models is given in Fig. 17. The confusion matrix of precision, recall, f1score, and accuracy of all the models are shown in Fig.18. The proposed research gives an in-depth comparison of eight machine learning classifiers. It classifies events based on accuracy, precision, recall, and the F1 score. When it comes

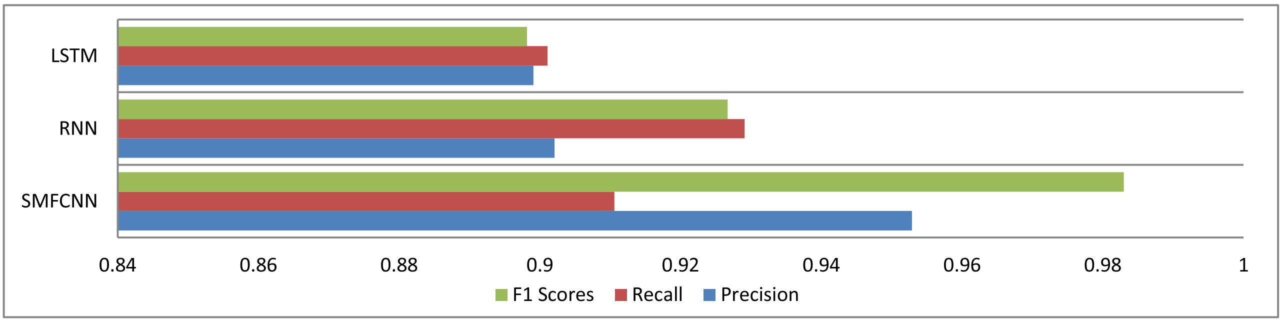

FIGURE 12. Precision, recall, and F1 score of deep learning models.

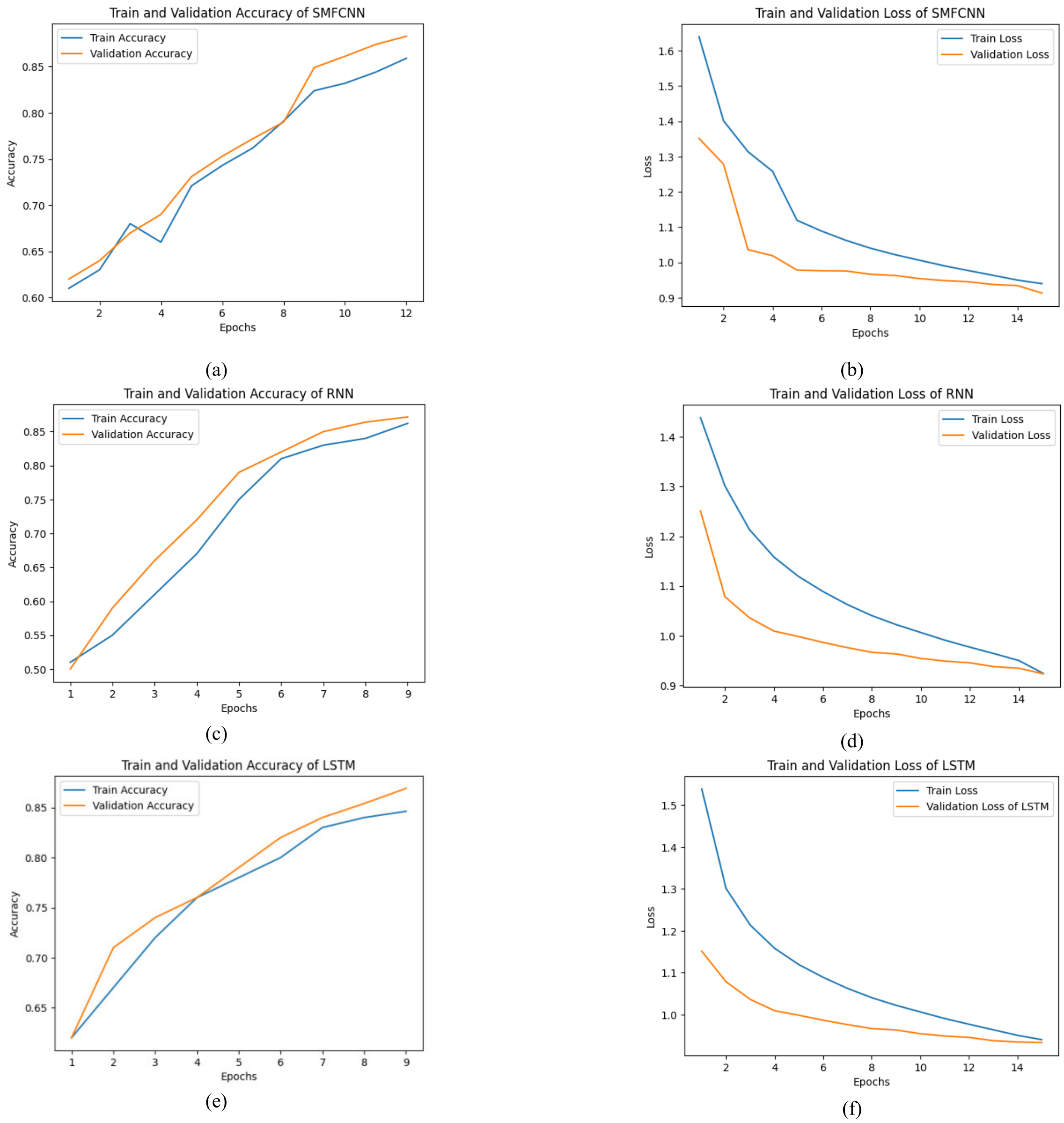

FIGURE 13. Training and validation results of deep learning models. (a) shows the train and validation accuracy of the SMFCNN model and (b) shows the train and validation loss of the SMFCNN model. (c) shows the train and validation accuracy of the RNN model and (d) shows the train and validation loss of the RNN model. (e) shows the train and validation accuracy of the LSTM-RNN model and (f) shows the train and validation loss of the LSTM-RNN model.

to precision, which assesses how many of the indicated positive instances are indeed positive, LR achieves an incredible precision value of 0.9358. The SVM precision value is lower than LR with a 0.9191 score. The Random Forest model has a

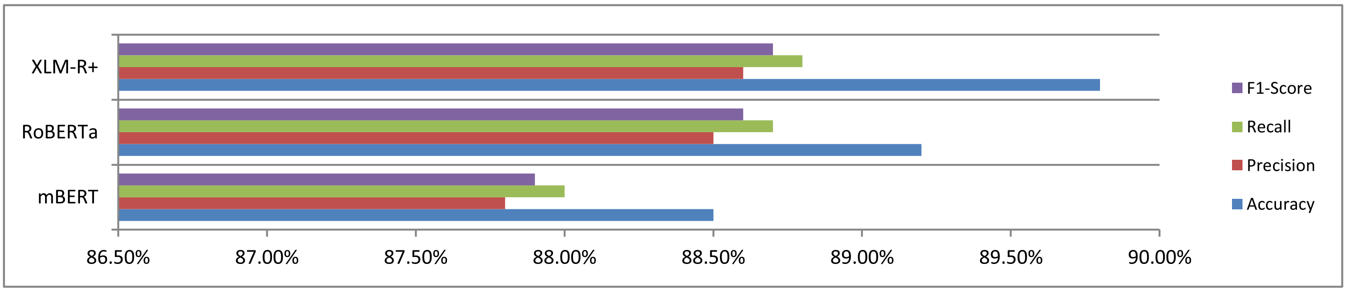

FIGURE 14. Precision, recall, F1 score, and accuracy of transformer-based models.

TABLE 17. Precision, Recall, F1-score, and accuracy of machine learning models.

ML Classifiers	Precision	Recall	F1 Scores	Accuracy
RF	0.5888	0.7394	0.6535	0.7776
KNN	0.7425	0.8395	0.7876	0.8441
AdaBoost	0.7380	0.7165	0.7271	0.7835
LR	0.9358	0.9399	0.9347	0.8712
DT	0.892	0.875	0.891	0.7671
XGBoost	0.7303	0.7362	0.7192	0.7502
MLP	0.5201	0.7277	0.6051	0.6922
SVM	0.9191	0.920	0.9152	0.8792

low precision of 0.5888, it outperforms in recall (the capacity to capture all genuine positives) with a score of 0.7394.

FIGURE 15. Accuracy of machine learning models.

With the highest recall of 0.9399, LR exhibits proficiency, while SVM demonstrates consistency by scoring well. With scores of 0.9347 and 0.9152, respectively, the F1 score, which perfectly balances precision and recall, demonstrates the prowess of LR and SVM.

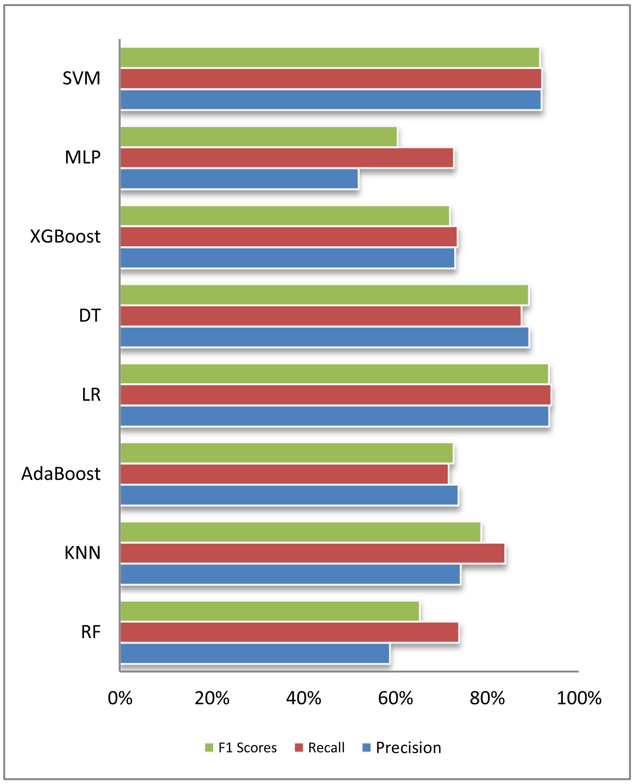

FIGURE 16. Precision, recall, and F1 scores of machine learning models.

C. COMPARISON WITH EXISTING RESEARCH CONCERNING THE NUMBER OF CLASSES

We investigated a wide range of models, including machine learning, deep learning, and transformer-based classifiers with 16 event classes. Our research outperforms other research [11], [40]. Notably, our K-Neighbors method is 84.41% accurate, compared to 78% in the other research. Similarly, our logistic regression model had an accuracy of 87.12% versus 80% in the previous research. Furthermore, our SVM model outperformed the other research with an accuracy of 87.92% versus 73%. Table 18 compares the achieved class-wised results of our proposed solution with the other state-of-the-art reported research [11], [40].

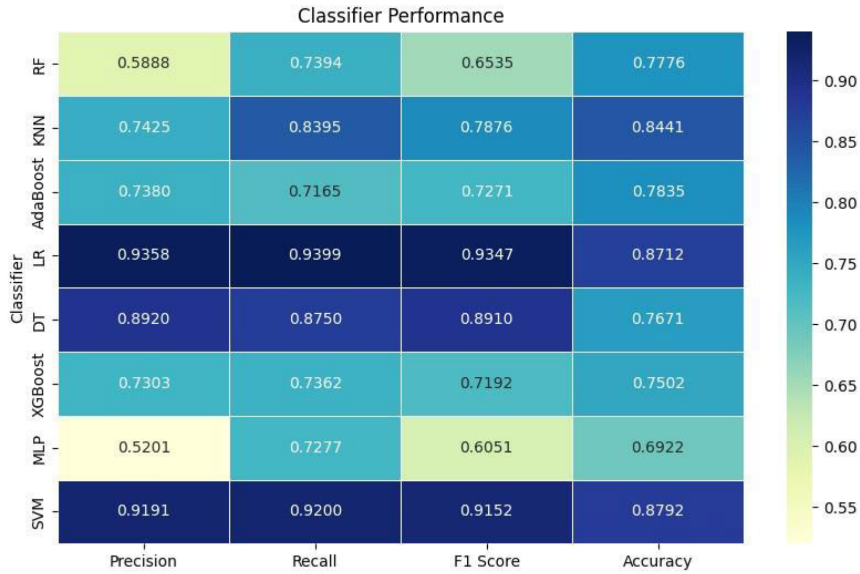

FIGURE 17. Performance of machine learning models.

TABLE 18. Comparison with existing research.

Model	16 Classes	14 Classes	12 Classes
Random Forest	77.76	80	80
K-Neighbors	84.41	78	78
AdaBoost	78.35	-	-
Logistic Regression	87.12	80	80
Decision Tree	76.71	73	73
XGBoost	75.02	-	-
MLP Classifier	69.22	-	-
SVM	**87.92**	73	73
SMFCNN	**88.29**	-	-
RNN	87.16	-	81
LSTM-RNN	86.91	-	84
CNN	-	-	80
DNN	-	-	84
NBM	-	70	70

Our deep learning classifiers performed well, with SMFCNN attaining an accuracy of 88.29% and RNN and LSTM-RNN achieving 87.16% and 86.91%, respectively. These findings highlight the significance of this research, demonstrating improved accuracies in the multiclass classification of Urdu events compared to previous research.

D. COMPARISON WITH EXISTING TRANSFORMER-BASED TECHNIQUES

In this section, we compare the performance of our proposed transformer-based model with existing techniques used for Urdu language processing. We particularly focus on the models including, Fine-tuned Urdu-BERT [63], Multilingual BERT [64], and XLM-R as mentioned in the prior work.

This efficiency over shadow state-of-the-art such as accuracy, precision, recall, and F1 score is already proved by our proposed model XLM-R+. Below, we present a detailed comparison of the performance metrics. A comparison of these techniques is given in Table 19.

TABLE 19. Comparison with other transformer-based models.

Model	Accuracy	Precision	Recall	F1-Score
Fine-tuned Urdu-BERT [63]	82.50%	85.25%	81.25%	83.20%
Multilingual BERT [64]	77.61%	76.15%	78.25%	77.18%
XLM-R [65]	88.00%	87.30%	87.50%	87.40%
mBERT	88.5%	87.8%	88.0%	87.9%
RoBERTa	89.2%	88.5%	88.7%	88.6%
XLM-R+	89.8%	88.6%	88.8%	88.7%

The comparison table clearly shows that our results compared with the results of other studies. The Fine-tuned Urdu-BERT model [63] achieves an accuracy of 82.50%, which is lower than our mBERT model's accuracy of 88.50%. Similarly, the Multilingual BERT [64] for sentiment analysis attains an accuracy of 77.61%, also lower than our RoBERTa and XLM-R+ models, which achieve 89.80% and 89.20%, respectively.

The proposed XLM-R+ model has an accuracy of 89.80%, precision of 88.60%, recall ratio of 88.80% and F1 score of 88.70%. This indicates that the features added into XLM-R+ including dynamic attention mechanism, adaptive learning rate, and extended epoch have a positive effect on the performance of Urdu sentence classification.

These results demonstrate that transformer-based models, particularly when accompanied by certain enrichments, can provide better performance in low-resource languages instead of Urdu.

Based on the multiclass classification of events in Urdu research, we got a good insight into the performance of different models. By comparing our findings between the two types of research, we find that our research obtains a considerably higher accuracy. Analyzing our results we can see that significant enhancements in K-Nearest Neighbor, logistic regression, and SVM models compared to traditional machine learning models are achieved. These findings therefore imply that it's high time stakeholders tried other models and strategies to get better outcomes. Some of the importance of our research is as follows demonstrating the added accuracy with 16 event classes in multiclass learning in Urdu.

V. CONCLUSION

Deep learning, transformer-based, and straightforward machine learning classifiers are popular in multiclass classification and have become the focus of this research by assessing the accuracy level they can achieve for classifying 16 different events from Urdu sentences. Therefore, while using performance measures such as precision, recall, F1-score, and accuracy, we have presented an evaluation of every built model's performance. The SMFCNN classifier also again stood out among the deep learning models with an accuracy of 88.29 percent for remarkable precision (95 percent), recall (91 percent), and F1-score (98 percent). The demonstrated difference between the training and validation accuracies in the SMFCNN model is a strong argument regarding the model's high generalization potential and minimal probability of overfitting. As for the group of transformer-based classifiers, it is worth noting that the highest average accuracy was observed when using the proposed XLM-R+ model and reached the value of 89.80%. The extended changes introduced in XLM-R+ such as the dynamic attention mechanism and adaptive learning rate increase its performance beyond the other models on Urdu sentence classification tasks. Other machine learning classifier benchmarks were equally good with traditional and non-ensemble SVM exposure achieving a peak of 87.92%. The accuracy of Logistic regression and K-NN classifiers was also increased significantly from the previous studies again confirming the novelty of the present work, which lies in the demonstration of exposure to a variety of models and approaches to get them tested to derive the best outcomes. All in all, this work demonstrates the possibility of using advanced deep learning and transformer-based approaches for low-resource languages such as Urdu. The approach seems to be highly beneficial to the classification of Urdu events in multiclass analysis and appears to outcompete previous methods.

Although this research is strong methodologically, several limitations exist. First, the dataset employed for the task is large but the variations in event descriptions in Urdu language may be beyond the scope of this dataset which may constrain the coverage of the proposed models in unseen data with such characteristics. Moreover, there is the issue of the relative difficulty of deep learning and transformer-based models even when they boast high accuracy, which demands a great amount of computational power when training – an especially important factor in low-resource environments. The generalization ability of the proposed SMFCNN was also very good, while the other models such as RNN and LSTM-RNN also showed signs of overfitting, especially as the number of training epochs where increased. However, Urdu as a low-resource language with complex textual features and inadequate annotated data introduces other challenges that the models may or may not solve and consequently will be the limitation towards its efficiency.

More event types should be included in the study as well as more different linguistic expressions to improve modelability in future studies. More enhancements and refinements are possible by fine-tuning hyperparameters and applying sophisticated methods of regularization which also can be used to minimize overfitting. Furthermore, showing that these methodologies can be naturally applied to real-time applications like content moderation and sentiment analysis, and showing that the same methodology applied to other low-resource languages is also effective, and practical. Integrating these classifiers with other AI technologies, such as natural language processing tools and knowledge graphs, can further enhance their capability to process complex sentences accurately.

ACKNOWLEDGMENT

Somia Ali would like to express her heartfelt gratitude to her supervisor, department professors, and the Department Head for their guidance, support, and significant insights into the research.

REFERENCES

[1] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval: The Concepts and Technology Behind Search*. Harlow, U.K.: Pearson, 2011.

[2] B. Liu, *Sentiment Analysis and Opinion Mining*. San Rafael, CA, USA: Morgan and Claypool, 2012.

[3] S. Nazir, M. Asif, S. A. Sahi, S. Ahmad, Y. Y. Ghadi, and M. H. Aziz, "Toward the development of large-scale word embedding for low-resourced language," *IEEE Access*, vol. 10, pp. 54091–54097, 2022, doi: 10.1109/ACCESS.2022.3173259.

[4] S. Deerwester, S. Dumais, G. W. Furnas, T. K. Landauer, and R. A. Harshman, "Indexing by latent semantic analysis," *J. Amer. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, Sep. 1990, doi: 10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.

[5] M. Qorich and R. El Ouazzani, "Text sentiment classification of Amazon reviews using word embeddings and convolutional neural networks," *J. Supercomput.*, vol. 79, no. 10, pp. 11029–11054, Jul. 2023, doi: 10.1007/s11227-023-05094-6.

[6] A. Dewani, M. A. Memon, S. Bhatti, A. Sulaiman, M. Hamdi, H. Alshahrani, A. Alghamdi, and A. Shaikh, "Detection of cyberbullying patterns in low resource colloquial Roman Urdu microtext using natural language processing, machine learning, and ensemble techniques," *Appl. Sci.*, vol. 13, no. 4, p. 2062, Feb. 2023, doi: 10.3390/app13042062.

[7] S. Kanwal, M. K. Malik, Z. Nawaz, and K. Mehmood, "Urdu wikification and its application in Urdu news recommendation system," *IEEE Access*, vol. 10, pp. 103655–103668, 2022, doi: 10.1109/ACCESS.2022.3208666.

[8] S. H. Kumhar, M. M. Kirmani, J. Sheetlani, and M. Hassan, "WITHDRAWN: Word embedding generation for Urdu language using Word2vec model," *Mater. Today, Proc.*, Jan. 2021, doi: 10.1016/j.matpr.2020.11.766.

[9] F. Benites and E. Sapozhnikova, "HARAM: A hierarchical ARAM neural network for large-scale text classification," in *Proc. IEEE Int. Conf. Data Mining Workshop (ICDMW)*, Nov. 2015, pp. 847–854, doi: 10.1109/ICDMW.2015.14.

[10] M. P. Akhter, Z. Jiangbin, I. R. Naqvi, M. Abdelmajeed, A. Mehmood, and M. T. Sadiq, "Document-level text classification using single-layer multisize filters convolutional neural network," *IEEE Access*, vol. 8, pp. 42689–42707, 2020, doi: 10.1109/ACCESS.2020.2976744.

[11] D. Ali, M. M. S. Missen, and M. Husnain, "Multiclass event classification from text," *Scientific Program.*, vol. 2021, pp. 1–15, Jan. 2021, doi: 10.1155/2021/6660651.

[12] C. C. Aggarwal, "Mining text data," in *Data Mining*, 2015, pp. 429–455, doi: 10.1007/978-3-319-14142-8_13.

[13] B. Mukhim and S. Das, "Sentiment analysis and emotion detection from Khasi text data—A survey," in *Proc. Int. Conf. Inf. Commun. Technol. Develop.*, 2023, pp. 171–180, doi: 10.1007/978-981-19-7528-8_14.

[14] K. Ahmed, M. I. Nadeem, D. Li, Z. Zheng, N. Al-Kahtani, H. K. Alkahtani, S. M. Mostafa, and O. Mamyrbayev, "Contextually enriched meta-learning ensemble model for Urdu sentiment analysis," *Symmetry*, vol. 15, no. 3, p. 645, Mar. 2023, doi: 10.3390/sym15030645.

[15] K. Ahmed, M. Ali, S. Khalid, and M. Kamran, "Framework for Urdu news headlines classification," *J. Appl. Comput. Sci. Math.*, vol. 10, no. 1, pp. 17–21, 2016, doi: 10.4316/jacsm.201601002.

[16] M. P. Akhter, Z. Jiangbin, I. R. Naqvi, M. Abdelmajeed, and M. Fayyaz, "Exploring deep learning approaches for Urdu text classification in product manufacturing," *Enterprise Inf. Syst.*, vol. 16, no. 2, pp. 223–248, Feb. 2022, doi: 10.1080/17517575.2020.1755455.

[17] M. P. Akhter, J. Zheng, F. Afzal, H. Lin, S. Riaz, and A. Mehmood, "Supervised ensemble learning methods towards automatically filtering Urdu fake news within social media," *PeerJ Comput. Sci.*, vol. 7, p. e425, Mar. 2021, doi: 10.7717/peerj-cs.425.

[18] S. Latif, F. Shafait, and R. Latif, "Analyzing LDA and NMF topic models for Urdu tweets via automatic labeling," *IEEE Access*, vol. 9, pp. 127531–127547, 2021, doi: 10.1109/ACCESS.2021.3112620.

[19] M. U. Arshad, R. Ali, M. O. Beg, and W. Shahzad, "UHated: Hate speech detection in Urdu language using transfer learning," *Lang. Resour. Eval.*, vol. 57, no. 2, pp. 713–732, Jun. 2023, doi: 10.1007/s10579-023-09642-7.

[20] S. Saleem, N. F. Khan, and S. Zafar, "Prevalence of cyberbullying victimization among Pakistani youth," *Technol. Soc.*, vol. 65, May 2021, Art. no. 101577, doi: 10.1016/j.techsoc.2021.101577.

[21] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Noida, India: Pearson, 2022.

[22] W. Yin and L. Shen, "A short text classification approach with event detection and conceptual information," in *Proc. 5th Int. Conf. Mach. Learn. Technol.*, Jun. 2020, pp. 129–135, doi: 10.1145/3409073.3409091.

[23] A. Bhatti, A. Arif, W. Khalid, B. Khan, A. Ali, S. Khalid, and A. U. Rehman, "Recognition and classification of handwritten Urdu numerals using deep learning techniques," *Appl. Sci.*, vol. 13, no. 3, p. 1624, Jan. 2023, doi: 10.3390/app13031624.

[24] Z. Xu, *Introduction to Semi-Supervised Learning*. New York, NY, USA: Taylor & Francis, 2013.

[25] T. A. Rana, K. Shahzadi, T. Rana, A. Arshad, and M. Tubishat, "An unsupervised approach for sentiment analysis on social media short text classification in Roman Urdu," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 21, no. 2, pp. 1–16, Mar. 2022, doi: 10.1145/3474119.

[26] M. Z. Ali, Ehsan-Ul-Haq, S. Rauf, K. Javed, and S. Hussain, "Improving hate speech detection of Urdu tweets using sentiment analysis," *IEEE Access*, vol. 9, pp. 84296–84305, 2021, doi: 10.1109/ACCESS.2021.3087827.

[27] A. Dhar, N. S. Dash, and K. Roy, "Application of TF-IDF feature for categorizing documents of online Bangla web text corpus," in *Intelligent Engineering Informatics*, 2018, pp. 51–59, doi: 10.1007/978-981-10-7566-7_6.

[28] Y. H. Li and A. K. Jain, "Classification of text documents," *Comput. J.*, vol. 41, no. 8, pp. 537–546, Jan. 1998, doi: 10.1093/comjnl/41.8.537.

[29] S. K. Dasari and S. Mehta, "Scene based text recognition and classification based on hybrid CNN models with performance evaluation," *SSRN Electron. J.*, 2022, doi: 10.2139/ssrn.4174796.

[30] C. Naulak, "A comparative study of naive Bayes classifiers with improved technique on text classification," *Authorea Preprints*, Oct. 2022, doi: 10.36227/techrxiv.19918360.

[31] A. Farooq, Z. Noreen, S. Batool, and F. Naz, "Urdu news classification: An empirical study using machine learning techniques," in *Proc. Mohammad Ali Jinnah Univ. Int. Conf. Comput. (MAJICC)*, Oct. 2022, pp. 1–7, doi: 10.1109/MAJICC56935.2022.9994152.

[32] W. H. Bangyal, R. Qasim, N. U. Rehman, Z. Ahmad, H. Dar, L. Rukhsar, Z. Aman, and J. Ahmad, "Detection of fake news text classification on COVID-19 using deep learning approaches," *Comput. Math. Methods Med.*, vol. 2021, pp. 1–14, Nov. 2021, doi: 10.1155/2021/5514220.

[33] R. Alghamdi and K. Alfalqi, "A survey of topic modeling in text mining," *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 1, 2015, doi: 10.14569/ijacsa.2015.060121.

[34] U. Pal and A. Sarkar, "Recognition of printed Urdu script," in *Proc. 7th Int. Conf. Document Anal. Recognit.*, vol. 1, Aug. 2003, pp. 1183–1187, doi: 10.1109/icdar.2003.1227844.

[35] H. Hafeez, I. Muneer, M. Sharjeel, M. A. Ashraf, and R. M. Adeel Nawab, "Urdu short paraphrase detection at sentence level," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 22, no. 4, pp. 1–20, Apr. 2023, doi: 10.1145/3586009.

[36] K. Sailunaz and R. Alhajj, "Emotion and sentiment analysis from Twitter text," *J. Comput. Sci.*, vol. 36, Sep. 2019, Art. no. 101003, doi: 10.1016/j.jocs.2019.05.009.

[37] J. Shafi, R. M. A. Nawab, and P. Rayson, "Semantic tagging for the Urdu language: Annotated corpus and multi-target classification methods," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 22, no. 6, pp. 1–32, Jun. 2023, doi: 10.1145/3582496.

[38] A. Muhammad, N. Jazeb, A. M. Martinez-Enriquez, and A. Sikander, "EUTS: Extractive Urdu text summarizer," in *Proc. 17th Mex. Int. Conf. Artif. Intell. (MICAI)*, Oct. 2018, pp. 39–44, doi: 10.1109/MICAI46078.2018.00014.

[39] M. D. A. Awan, N. I. Kajla, A. Firdous, M. Husnain, and M. M. S. Missen, "Event classification from the Urdu language text on social media," *PeerJ Comput. Sci.*, vol. 7, p. e775, Nov. 2021, doi: 10.7717/peerj-cs.775.

[40] S. Ali, U. Jamil, M. Jabbar, and M. A. Jabbar, "Sentence-level classification of web-extracted data in Urdu langugare (ULT)," *J. Xi'an Shiyou Univ., Natural Sci. Ed.*

[41] D. Roobaert, G. Karakoulas, and N. V. Chawla, "Information gain, correlation and support vector machines," in *Feature Extraction*, pp. 463–470, doi: 10.1007/978-3-540-35488-8_23.

[42] J. Lilleberg, Y. Zhu, and Y. Zhang, "Support vector machines and Word2vec for text classification with semantic features," in *Proc. IEEE 14th Int. Conf. Cognit. Informat. Cognit. Comput. (ICCI*CC)*, Jul. 2015, pp. 136–140, doi: 10.1109/ICCI-CC.2015.7259377.

[43] A. Hassan and A. Mahmood, "Deep learning for sentence classification," in *Proc. IEEE Long Island Syst., Appl. Technol. Conf. (LISAT)*, May 2017, pp. 1–5, doi: 10.1109/LISAT.2017.8001979.

[44] P. Nakov, "Semantic sentiment analysis of Twitter data," in *Encyclopedia of Social Network Analysis and Mining*, 2018, pp. 2339–2350, doi: 10.1007/978-1-4939-7131-2_110167.

[45] S. Bloehdorn and A. Hotho, "Text classification by boosting weak learners based on terms and concepts," in *Proc. 4th IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2004, pp. 331–334, doi: 10.1109/ICDM.2004.10077.

[46] N. Kalcheva, M. Karova, and I. Penev, "Comparison of the accuracy of SVM kemel functions in text classification," in *Proc. Int. Conf. Biomed. Innov. Appl. (BIA)*, Sep. 2020, pp. 141–145, doi: 10.1109/BIA50171.2020.9244278.

[47] S. T. Indra, L. Wikarsa, and R. Turang, "Using logistic regression method to classify tweets into the selected topics," in *Proc. Int. Conf. Adv. Comput. Sci. Inf. Syst. (ICACSIS)*, Oct. 2016, pp. 385–390, doi: 10.1109/ICACSIS.2016.7872727.

[48] K. Raychaudhuri, M. Kumar, and S. Bhanu, "A comparative study and performance analysis of classification techniques: Support Vector Machine, neural networks and decision trees," in *Advances in Computing and Data Sciences*, 2017, pp. 13–21, doi: 10.1007/978-981-10-5427-3_2.

[49] A. Samih, A. Ghadi, and A. Fennan, "Enhanced sentiment analysis based on improved word embeddings and XGboost," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 13, no. 2, p. 1827, Apr. 2023, doi: 10.11591/ijece.v13i2.pp1827-1836.

[50] K. Shah, H. Patel, D. Sanghvi, and M. Shah, "A comparative analysis of logistic regression, random forest and KNN models for the text classification," *Augmented Hum. Res.*, vol. 5, no. 1, Dec. 2020, doi: 10.1007/s41133-020-00032-0.

[51] Z. Zhang, "Improved Adam optimizer for deep neural networks," in *Proc. IEEE/ACM 26th Int. Symp. Quality Service (IWQoS)*, Jun. 2018, pp. 1–2, doi: 10.1109/iwqos.2018.8624183.

[52] 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, doi: 10.1109/ACCESS.2021.3093078.

[53] 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, doi: 10.1109/ACCESS.2021.3104308.

[54] 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, doi: 10.1109/ACCESS.2021.3122025.

[55] A. Altaf, M. W. Anwar, M. H. Jamal, S. Hassan, U. I. Bajwa, G. S. Choi, and I. Ashraf, "Deep learning based cross domain sentiment classification for Urdu language," *IEEE Access*, vol. 10, pp. 102135–102147, 2022, doi: 10.1109/ACCESS.2022.3208164.

[56] R. Anam, M. W. Anwar, M. H. Jamal, U. I. Bajwa, I. D. L. T. Diez, E. S. Alvarado, E. S. Flores, and I. Ashraf, "A deep learning approach for named entity recognition in Urdu language," *PLoS ONE*, vol. 19, no. 3, Mar. 2024, Art. no. e0300725, doi: 10.1371/journal.pone.0300725.

[57] M. F. Bashir, A. R. Javed, M. U. Arshad, T. R. Gadekallu, W. Shahzad, and M. O. Beg, "Context-aware emotion detection from low-resource Urdu language using deep neural network," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 22, no. 5, pp. 1–30, May 2023, doi: 10.1145/3528576.

[58] M. Y. Khan, T. Ahmed, M. S. Siddiqui, and S. Wasi, "Cognitive relationship-based approach for Urdu sarcasm and sentiment classification," *IEEE Access*, vol. 11, pp. 126661–126690, 2023, doi: 10.1109/ACCESS.2023.3325048.

[59] A. Khan, A. Ahmed, S. Jan, M. Bilal, and M. F. Zuhairi, "Abusive language detection in Urdu text: Leveraging deep learning and attention mechanism," *IEEE Access*, vol. 12, pp. 37418–37431, 2024, doi: 10.1109/ACCESS.2024.3370232.

[60] Z. Iqbal, F. M. Khan, I. U. Khan, and I. U. Khan, "Fake news identification in Urdu tweets using machine learning models," *Asian Bull. Big Data Manage.*, vol. 4, no. 1, Feb. 2024, doi: 10.62019/abbdm.v4i1.105.

[61] M. Shabbir and M. Majid, "Sentiment analysis from Urdu language-based text using deep learning techniques," in *Proc. 5th Int. Conf. Advancements Comput. Sci. (ICACS)*, Feb. 2024, pp. 1–5, doi: 10.1109/ICACS60934.2024.10473232.

[62] M. S. Khan, M. S. I. Malik, and A. Nadeem, "Detection of violence incitation expressions in Urdu tweets using convolutional neural network," *Expert Syst. Appl.*, vol. 245, Jul. 2024, Art. no. 123174, doi: 10.1016/j.eswa.2024.123174.

[63] M. S. I. Malik, U. Cheema, and D. I. Ignatov, "Contextual embeddings based on fine-tuned Urdu-BERT for Urdu threatening content and target identification," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 35, no. 7, Jul. 2023, Art. no. 101606, doi: 10.1016/j.jksuci.2023.101606.

[64] 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, Mar. 2022, doi: 10.1038/s41598-022-09381-9.

[65] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," 2019, *arXiv:1911.02116*.

[66] M. E. Hassan, M. Hussain, I. Maab, U. Habib, M. A. Khan, and A. Masood, "Detection of sarcasm in Urdu tweets using deep learning and transformer based hybrid approaches," *IEEE Access*, vol. 12, pp. 61542–61555, 2024, doi: 10.1109/ACCESS.2024.3393856.

[67] S. Hussain, M. S. I. Malik, and N. Masood, "Identification of offensive language in Urdu using semantic and embedding models," *PeerJ Comput. Sci.*, vol. 8, p. e1169, Dec. 2022, doi: 10.7717/peerj-cs.1169.

SOMIA ALI is currently pursuing the Ph.D. degree with the Computer Science Department, Government College University Faisalabad. Her research area of focus is natural language processing. Her work is marked by a commitment to innovation, a meticulous approach to experimentation, and a passion for translating theoretical insights into practical solutions. Her research interests include feature engineering, feature extraction, and classification.

UZMA JAMIL (Member, IEEE) received the Ph.D. degree in computer engineering from Bahria University, Islamabad, Pakistan. She has been an Execution Manager with the Business Incubation Center, GCUF. She has been granted a Ph.D. scholarship and travel grants from the Higher Education Commission of Pakistan for research and international conference participation. She is currently an Assistant Professor with the Computer Science Department, Government College University Faisalabad, Pakistan, and has been doing other administrative tasks, since April 2005. Her research interests include computing, biomedical imaging, image processing, computer vision, and pattern recognition. She was one of the founding members of the IEEE Faisalabad Sub-Section. In 2023, she was elected as the Chairperson of the IEEE Faisalabad Sub-Section. Being part of the IEEE Faisalabad Sub-Section, she organized, conducted, and attended many national and international events representing IEEE. Since 2019, she has been a General Secretary of IEEE EX-COM (IEEE Faisalabad Sub-Section). She is serving as an international journal and conference reviewer for IEEE and Springer.

MUHAMMAD YOUNAS received the Ph.D. degree from the Faculty of Engineering, School of Computing, Universiti Teknologi Malaysia (UTM). He is currently an Assistant Professor with the Computer Science Department, Government College University Faisalabad, Pakistan. His research interests include software engineering, agile software development, cloud computing, and code clone detection.

BUSHRA ZAFAR received the Ph.D. degree in computer science from National Textile University (NTU), Faisalabad, Pakistan. She is currently an Assistant Professor with the Department of Computer Science, Government College University Faisalabad (GCUF), Pakistan. She looks forward to collaborating with esteemed researchers, addressing pressing challenges, and sharing knowledge with the next generation of scholars. Her research interests include computer vision, digital image processing, and artificial intelligence.

MUHAMMAD KASHIF HANIF is currently an Associate Professor and the Chairperson of the Department of Computer Science, Government College University Faisalabad. His approach to research and mentorship is characterized by a deep-seated passion for fostering intellectual curiosity, encouraging critical thinking, and nurturing the development of independent researchers. His research interests include machine learning, bioinformatics, and GPU computing.