

Research Article

Multiclass Event Classification from Text

Daler Ali , Malik Muhammad Saad Missen , and Mujtaba Husnain 

The Islamia University Bahawalpur, Bahawalpur, Pakistan

Correspondence should be addressed to Daler Ali; daler.ali@iub.edu.pk and Malik Muhammad Saad Missen; saad.missen@iub.edu.pk

Received 3 November 2020; Accepted 24 December 2020; Published 13 January 2021

Academic Editor: Ligang He

Copyright © 2021 Daler Ali et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Social media has become one of the most popular sources of information. People communicate with each other and share their ideas, commenting on global issues and events in a multilingual environment. While social media has been popular for several years, recently, it has given an exponential rise in online data volumes because of the increasing popularity of local languages on the web. This allows researchers of the NLP community to exploit the richness of different languages while overcoming the challenges posed by these languages. Urdu is also one of the most used local languages being used on social media. In this paper, we presented the first-ever event detection approach for Urdu language text. Multiclass event classification is performed by popular deep learning (DL) models, i.e., Convolution Neural Network (CNN), Recurrence Neural Network (RNN), and Deep Neural Network (DNN). The one-hot-encoding, word embedding, and term-frequency inverse document frequency- (TF-IDF-) based feature vectors are used to evaluate the Deep Learning (DL) models. The dataset that is used for experimental work consists of more than 0.15 million (103965) labeled sentences. DNN classifier has achieved a promising accuracy of 84% in extracting and classifying the events in the Urdu language script.

1. Introduction

In the current digital era, social media dominated other sources of communication, i.e., print and broadcast media [1]. Real-time availability [2] and multilingual support [3] are the key features that boost the usage of social media for communication. The usage of local languages on social media is overwhelming for the last few years. People share ideas, opinions, events, sentiments, and advertisements, etc. [4] in the world via social media using local languages. A considerable amount of heterogeneous data is being generated which causes challenges to extract worthy insights, while this information plays a vital role in developing natural language processing (NLP) application, i.e., sentiment analysis [5], risk factor analysis [6], law and order predictor, timeline constructor, opining mining, decision-making systems [7], monitoring social media [8], spam detection, information retrieval, document classification [9], e-mail categorization [10], and sentence classification [11], topic modeling [12], content labeling, and finding the latest trend.

In South Asia (<https://www.worldometers.info/>), about 24.98% population of the world live in different countries.

Many languages are being spoken in Asia. The most famous among these are Arabic, Hindi, Malay, Persian, and Urdu, etc.

1.1. Features of Urdu Language. The Urdu language is one of the languages in South Asia that is frequently used for communication on social media, namely, Facebook, Twitter, News Channels, and Web Blogs [13]. It is also the national language of Pakistan which is the 6th (<https://www.worldometers.info/world-population/population-by-country/>) most populous country in the world. In other countries, i.e., India, Afghanistan, and Iran, the Urdu language is also spoken and understood. There are 340 million people in the world who use the Urdu language on social media for various purposes [13].

The Urdu language follows the right-to-left writing script. Its grammatical structure is different from other languages.

- (1) Subject-object-verb (SOV) sentence structure [14]
- (2) No letter capitalization

- (3) Diacritics
- (4) Free word order [15]

The Urdu language has 38 basic characters which can be written as joined and non-joined with other characters [16]. The words having joined characters of Urdu alphabet set are called ligature, and this joining feature of the alphabets made possible to enrich the Urdu vocabulary having almost 24,000 ligatures [15, 16]. It is pertinent to mention that this alphabet set is also considered as a superset of all Urdu script-based languages alphabets, namely, the Arabic and Persian, which contain 28 and 32 alphabets, respectively. Furthermore, there are also some additional alphabets in Urdu script that are used to express some Hindi phonemes [15, 16].

1.2. Event Classification. An event can be defined as “specific actions, situations, or happenings occurring in a certain period [17, 18].” The extracted information can represent different types of events, i.e., sports, politics, terrorist attacks, and inflation, etc.; information can be detected and classified at a different level of granularity, i.e., document level [19], sentence level [20], word level, character level, and phrase level [21].

Event classification is an automated way to assign a predefined label to new instances. It is pertinent to describe that the classification can be binary, multiclass, and multilabel [22].

The implementation of a neural network for text classification provided help to handle a complex and large amount of data [23]. Semantically similar words are used to generate feature vectors [24] that eliminate the sparsity of n-grams models. Urdu text classification is performed [25] to assess the quality of the product based on comments and feedback. In [25], an embedded layer of the neural network was used to convert text into numeric values and classification performed at the document level. Contrary to [25], multiclass event classification is performed at the sentence level instead of the document level. We further performed multiple experiments to develop an efficient classification system using TF-IDF, one-hot-encoding, pretrained Urdu word embedding model and by creating custom pretrained Urdu language word embedding models.

1.3. Event Classification Challenges. The lack of processing resources, i.e., part-of-speech (PoS) tagger, name, entity recognizer, and annotation tools, is the other major hurdle to perform the event detection and classification for the Urdu language. Many people are unfamiliar with the meaning and usage of some Urdu words. It creates semantically ambiguous content that makes the event classification process a nontrivial and challenging task. The unavailability of appropriate resources/datasets is another major challenge for data-driven and knowledge-based approaches to extract events and classify events.

Our contributions are given as follows:

- (1) The first-ever large-scale labeled Urdu dataset for event classification that is the biggest in terms of

instances [15] and classes [25] in other Urdu text datasets reported in state of the art [19, 26, 27]

- (2) To our best knowledge, it is the first multiclass event classification task at sentence level for the Urdu language
- (3) Different feature vector generating methods, i.e., one-hot-encoding, word embedding, and TF-IDF, are used to evaluate the performance of DNN, CNN, and RNN deep learning models
- (4) Pretrained and custom word embedding models for the Urdu language are also explored
- (5) Performance comparison of traditional machine learning classifiers and deep learning classifiers

In this paper, we performed a multiclass event classification on an imbalance dataset of Urdu language text. Our framework is a design to classify twelve different types of events, i.e. sports, inflation, politics, casualties, law and order, terrorist attack, sexual assault, fraud (and corruption), showbiz, business, weather, and earthquake. Furthermore, we also presented a detailed comparative analysis of different deep learning algorithms, i.e., long short-term memory (LSTM) and convolutional neural network (CNN) using TF-IDF, one-hot-coding, and word embedding methods. We also compared the results of traditional machine learning classifiers with deep learning classifiers.

2. Related Work

In the past, researchers were impassive in the Urdu language because of limited processing resources, i.e., datasets, annotators, part-of-speech (PoS) taggers, and translators [14], etc. However, now, since the last few years, feature-based classification for Urdu text documents started the use of machine learning models [28–30]. A framework was proposed [31] to classify Chinese short texts into 7 kinds [32] of emotion and product review. The event-level information from the text and conceptual information from the external knowledge base are provided as supplementary input to the neural models.

A fusion of CNN and RNN models is used to classify sentences using a movie review dataset and achieved 93% accuracy [33]. A comparative research study of machine learning (ML) and deep learning (DL) models is presented [25] for Urdu text classification at the document level. CNN and RNN single-layer/multilayer architectures are used to evaluate three different sizes of the dataset [26]. The purpose of their work was to analyze and to predict the quality of products, i.e., valuable, not valuable, relevant, irrelevant, bad, good, or very good [25].

Different datasets reported in state of the art, i.e., Northwestern Polytechnical University Urdu (NPUU), consist of 10K news articles labeled into six classes, Naïve dataset including 5003 news articles consists of five classes [34] and Corpus of Urdu News Text Reuse (COUNTER) having 1200 news articles with five classes [27]. A joint framework consisting of CNN and RNN layers was used for sentiment analysis [35]. Stanford movie review dataset and

Stanford Treebank dataset were used to evaluate the performance of the system. Their proposed system showed 93.3% and 89.2% accuracy, respectively.

In [35], the authors performed a supervised text classification in the Urdu language by using a statistical approach like Naïve Bayes and support vector machine (SVM). The classification is initiated by applying different preprocessing approaches, namely, stemming, stop word removal, and both stop words elimination and stemming. The experimental results showed that the stemming process has little impact on improving performance. On the other hand, the elimination of stop words showed a positive effect on results. The SVM outperformed the Naïve Bayes by achieving the classification accuracies of 89.53% and 93.34% based on polynomial and radial function, respectively.

Similarly, the SVM is also applied in the news headlines classification [36] in Urdu text showing a very low amount of accuracy improvement of 3.5%. News headlines are a small piece of information that frequently does not describe the contextual meaning of the contents. In [36], the majority voting algorithm used for text classification in the Urdu language showed 94% accuracy. The classification is performed on seven different types of news text. However, the number of instances was very limited. A dynamic neural network [37] was designed to model the sentiment of sentences. It consists of dynamic K-modeling, pooling, and global pooling over a linear sequence that performs multiclass sentiment classification.

A quite different task is performed [38] in which the authors used a hybrid approach of rule-based and machine learning-based techniques to perform the sentiment classification while analyzing the Urdu script [38] at the phrase level. The hybrid approach showed an accuracy of 31.25%, 8.46%, and 21.6% using the performance metrics of recall, precision, and accuracy, respectively. In [39], a variant of recurrent neural network (RNN) called long short-term memory (LSTM) is used to overcome the weakness of bag-of-words and n -grams models and it outperformed these conventional approaches.

A neural network-based system [39] was developed to classify events. The purpose of the system was to help the people in natural disasters like floods by analyzing tweets. The Markov model was used to classify and predict the location that showed 81% accuracy for classification tweets as a request for help and 87% accuracy to locate the location. Research work was conducted on life event detection and classification, i.e., marriage, birthday, and traveling, etc., to anticipate products and services to facilitate the people [40]. The data about life events exist in a very small amount. Linear regression, Naïve Bayes, and nearest neighbor algorithms were evaluated on the original dataset that was very small but did not show favorable results.

A multiple minimal reduct extraction algorithm was designed [41] by improving the quick reduct algorithm. The multiple reducts are used to generate the set of classification rules which represent the rough set classifier. To evaluate the proposed approach, an Arabic corpus of 2700 documents was used to categorize into nine classes. By using multiple and single minimal reducts, the proposed system showed

94% and 86%, respectively. Experimental results also showed that both the K-NN and J48 algorithms outperformed regarding classification accuracy using the dataset on hand.

Table 1 depicts the summary of the related research discussed previously.

3. Dataset

3.1. Data Collection. Contrary to the dataset reported in state of the art [27, 34] in which no datasets were created for event classification, we created a larger dataset specific for event classification. Instead of focusing on a specific product [25] analysis, or phrase-level sentiment analysis [38], we decided to classify sentences into multiple event classes. Instead of using the joint framework of CNN and RNN for sentiment analysis [35], we evaluated the performance of deep learning models for multiclass event classification. To collect data, a PHP-based web scraper is written to crawl data from the popular social media websites, i.e., Geo News Channel (<https://urdu.geo.tv/>) website, BBC Urdu (<https://www.bbc.com/urdu>), and Urdu point (<https://www.urdupoint.com/daily/>). A complete post is retrieved from the website and stored in MariaDB (database). It consists of a title, body, published date, location, and URL. The sample text or tweet of both languages of the South Asian countries, i.e., Urdu language on Twitter and Hindi language on Facebook, is shown in Figure 1.

There are 0.15 million (150,000) Urdu language sentences. The diversity of data collection sources helped us to develop multiclass datasets. It consists of twelve types of events. The subset of datasets can be useful for other researchers.

3.2. Preprocessing. In the first phase of dataset preparation, we performed some preprocessing steps, i.e., noise removing and sentence annotation/labeling. All non-Urdu words, sentences, hyperlinks, URLs, and special symbols were removed. It was necessary to clean out the dataset to annotate/label the sentences properly.

3.2.1. Annotation Guidelines

- (1) Go through each sentence and assign a class label
- (2) Remove ambiguous sentences
- (3) Merge relevant sentences to a single class, i.e., accident, murder, and death
- (4) Assign one of the twelve types of events, i.e., sports, inflation, murder and death, terrorist attack, politics, law and order, earthquake, showbiz, fraud and corruption, weather, sexual assault, and business, to each sentence

To annotate our dataset, two M.Phil. (Urdu) level language experts were engaged. They deeply read and analyzed the dataset sentence by sentence before assigning event labels. They recommended removing 46035 sentences from the dataset because those sentences would not contain information that useful for event classification. Finally, after

TABLE 1: Summary of the related research.

Paper reference	Classifier used	Dataset	Accuracy (%)
[33]	CNN and RNN	Movie reviews	92
[34]	CNN and RNN	1. Stanford movie review dataset 2. Stanford Treebank dataset	93.3 and 89.2
[35]	Naïve Bayes and SVM	Corpus of Urdu documents	89.53 and 93.34
[36]	Dynamic neural network	News articles	96.5
[38]	Rule-based modeling	Urdu corpus of news headlines	31.25
[39]	LSTM	Tweets	81.00
[41]	K-NN and J48	Arabic corpus of 2700 documents	95 and 86



FIGURE 1: Urdu and Hindi language text on social media.

annotation, the dataset size was reduced to 103965 imbalanced instances of twelve different types of events.

The annotation inter-agreement, i.e., Cohen Kappa score, is 0.93, which indicates the strong agreement between the language and expert annotators. The annotated dataset is almost perfect according to the annotation agreement score.

In the second phase of preprocessing, the following steps are performed, i.e., stop words eliminated, word tokenization, and sentence filtering.

All those words which do not semantically contribute to the classification process are removed as stop words, i.e., *وہ*, *اور*, *سے*, *وغیرہ*, *وغیرہ*, *میں*, *اس*, *اور*, *سے*, *وغیرہ*, *وغیرہ*, etc. A list of standard stop words of the Urdu language is available here (<https://www.kaggle.com/rtatman/urdu-stopwords-list>).

After performing data cleaning and stop word removal, every sentence is tokenized into words based on white space. An example of sentence tokenization is given in Table 2.

The previous preprocessing step revealed that many sentences are varying in length. Some sentences were so short, and many were very long. We decided to define a length boundary for tokenized sentences. We observed that many sentences exist in the dataset which have a length range from 5 words to 250 words. We selected sentences that consist of 5 words to 150 words. An integer value is assigned to each type of event for all selected sentences. The detailed description of the different types of events and their corresponding numeric (integer) values that are used in the dataset is also given in Table 3.

In Figure 2, a few instances of the dataset after preprocessing are presented. It is a comma-separated value

(CSV) file that consists of two fields, i.e., sentence and label, i.e., numeric value for each class (1–12).

In our dataset, three types of events have a larger number of instances, i.e., sports (18746), politics (33421), and fraud and corruption (10078), contrary to three other types of events that have a smaller number of instances, i.e., sexual assault (2916), inflation (3196), and earthquake (3238).

The remaining types of events have a smaller difference of instances among them. There are 51814 unique words in our dataset. The visualization in Figure 3 shows that the dataset is imbalanced.

4. Methodology

We analyzed the performance of deep learning, i.e., deep neural network, convolutional neural network, and recurrent neural network, along with other machine learning classifiers, i.e., *K*-nearest neighbor, decision tree, random forest, support vector machine, Naïve Bayes multinomial, and linear regression.

The Urdu news headlines contain insufficient information, i.e., few numbers of words and lack of contextual information to classify the events [29]. However, comparatively, to news headlines, the sentences written in informal way contain more information. The sentence-level classification is performed using deep learning models instead of only machine learning algorithms. The majority voting algorithm outperforms on a limited number of instances for seven classes. It showed 94% [36] accuracy, but in our work, more than 0.15 million instances which are labeled into twelve classes are used for classification.

There exist several approaches to extract useful information from a large amount of data. Three common approaches are rule-based, a machine learning approach, and hybrid approaches [42]. The selection of methodology is tightly coupled with the research problem. In our problem, we decided to use machine learning (traditional machine learning and deep learning approaches) classifiers. Some traditional machine learning algorithms, i.e., *K*-nearest neighbor (KNN), random forest (RF), support vector machine (SVM), decision tree (DT), and multinomial Naïve Bayes (MNB), are evaluated for multiclass event classification.

Deep learning models, i.e., convolutional neural network (CNN), deep neural network (DNN), and recurrent neural network (RNN), are also evaluated for multiclass event classification.

TABLE 2: Sentence tokenization.

Sentence	Tokenized sentence						
کرونا وائرس نے متعدد لوگوں کی جان لیے لی۔ طوفانی بارش سے کئی گھروں کی چھت گر گئی	لی	کرونا	وائرس	نے	متعدد	لوگوں	کی
	گی	چھت	بارش	سے	کئی	گھروں	کی

TABLE 3: Event label.

Event	Label
Sports	1
Inflation	2
Murder and death	3
Terrorist attack	4
Politics	5
Law and order	6
Earthquake	7
Showbiz	8
Fraud and corruption	9
Rain/weather	10
Sexual assault	11

sentence,label

محمد حسام فرانسسی صدر ملاقات بتایا سڑک گزر نظر پڑی عمارت جمع	3
مقامی میڈیا مظاہرین یورپی یونین کونسل عمارت چوتوں 4500 چوڑے رکھ احتجاج	6
مہم گروپ جانب یورپی ممالک حکومتیں اسرائیلی مظالم خلاف آواز اٹھائیں	6
اسرائیلی فوج 2 ماہ بچوں سمیت 110 زائد فلسطینیوں شہید	6
شدید بارشوں پالی کئی گھروں داخل ہوگیا گھروں سڑکوں کھڑی گاڑیاں ڈوب گئیں	10
محکمہ موسمیات شہر 19 زائد مقامات فٹہ وارننگ جاری	10
شدید بارش اسپارک بل، پورٹنلے گرین ایسٹ، ایچسٹن سٹی ایوک علاقے متاثر ہوئے	10
انتظامیہ ڈرائیورز متاثرہ علاقوں گاڑی اجتباب برتنے ہدایات جاری	10
چینر مین پی بی ٹیم سیٹھی دنوں لندن موجود لارڈز قومی ٹیم ٹیسٹ دیکھا	1
لیب اعلامیے پنجاب لیڈ ٹویٹلمٹ کمپنی 2015 ایل ڈی آشیانہ اقبال پروجیکٹ ٹھیکا	5
5 دیا، کمپنی ٹھیکا ایل ٹھی، ٹھیکا پی پی ایس ایکٹ 2014 خلاف ورزی	5
5 احد چیمہ گرفتاری پنجاب حکومت شدید تحفظات اظہار لیب کارروائی مناسب قرار	5
پنجاب نگران وزیراعلیٰ ناصر محمود کھوسہ نام اتفاق	5

FIGURE 2: The few instances of the dataset.

A collection of Urdu text documents $D = \{d_1, d_2, \dots, d_n\}$ is split into a set of sentences $S = \{s_1, s_2, \dots, s_n\}$. Our purpose is to classify the sentences to a predefined set of events $E = \{e_1, e_2, \dots, e_n\}$.

Various feature generating methods are used to create a feature vector for deep learning and machine learning classifiers, i.e., TF-IDF, one-hot-encoding, and word embedding. Feature vectors generated by all these techniques are fed up as input into the embedding layer of neural networks. The output generated by the embedding layers is fed up into the next fully connected layer (dense layer) of deep learning models, i.e., RNN, CNN, and DNN. A relevant class label out of twelve categories is assigned to each sentence at the end of model processing in the testing/validation phase.

Bag-of-words is a common method to represent text. It ignores the sequence order and semantic of text [43], while the one-hot-coding method maintains the sequence of text. Word embedding methods Word2Vec and Glove (<https://ybbaiigo.gitbooks.io/26/pretrained-word-embeddings.html>) that are used to generate feature vectors for deep learning models are highly recommended for textual data. However, in the case of Urdu text classification, pre-existing wrod2vec and Glove are incompatible.

The framework of our designed system is represented in Figure 4. It shows the structure of our system from taking input to producing output.

5. Experimental Setup

We performed many experiments on our dataset by using various traditional machine learning and deep learning classifiers. The purpose of many experiments is to find the most efficient and accurate classification model for the multiclass event on an imbalance dataset for the Urdu language text. A detailed comparison between traditional classifiers and deep neural classifiers is given in the next section.

5.1. Feature Space. Unigram and bigram tokens of the whole corpus are used as features to create the feature space. TF-IDF vectorization is used to create a dictionary-based model. It consists of 656608 features. The training and testing dataset are converted to TF-IDF dictionary-based feature vectors. A convolutional sequential model (see Figure 5) consists of three layers, i.e., the input layer, hidden layer, and output layer, which are used to evaluate our dataset. Similarly, word embedding and one-hot-encoding are also included in our feature space to enlarge the scope of our research problem.

5.2. Feature Vector Generating Techniques. Feature vectors are the numerical representation of text. They are an actual form of input that can be processed by the machine learning classifier. There are several feature generating techniques used for text processing. We used the following feature vector generating techniques.

5.2.1. Word Embedding. A numerical representation of the text is that each word is considered as a feature vector. It creates a dense vector of real values that captures the contextual, semantical, and syntactical meaning of the word. It also ensures that similar words should have a related weighted value [29].

5.2.2. Pretrained Word Embedding Models. Usage of a pretrained word embedding model for the small amount of data is highly recommended by researchers in state of the art. Glove and Word2Vec are famous word embedding models that are developed by using a big amount of data. Word embedding models for text classification, especially in the English language, showed promising results. It has emerged as a powerful feature vector generating technique among others, i.e., TF, TF-IDF, and one-hot encoding, etc.

In our research case, sentence classification for different events in the Urdu language using the word embedding technique is potentially preferable. Unfortunately, the Urdu language is lacking in processing resources. We found only three word embedding models, a word embedding model

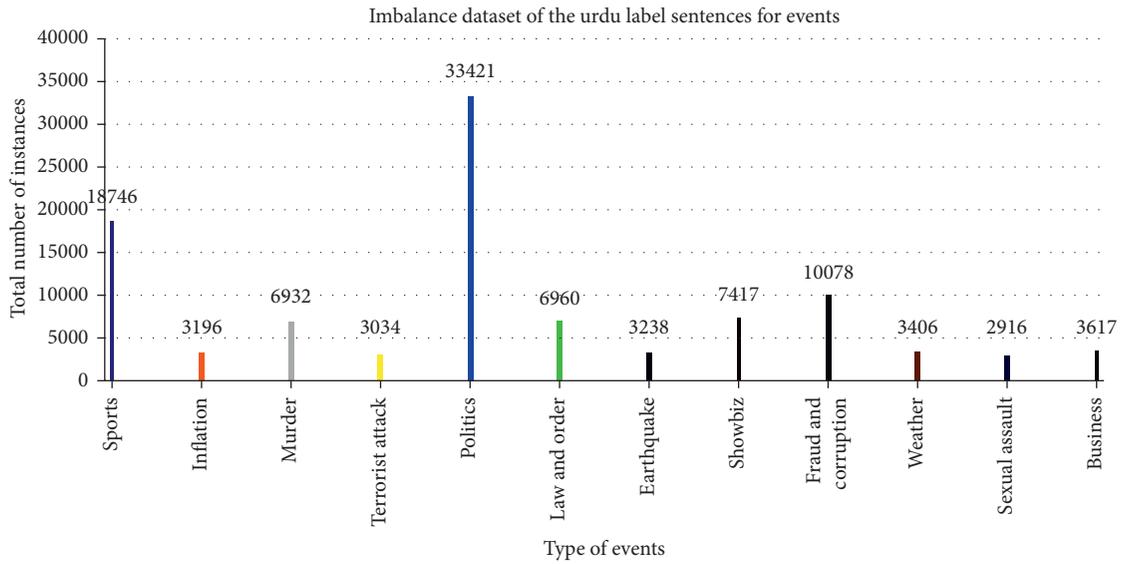


FIGURE 3: Imbalance instances of the dataset.

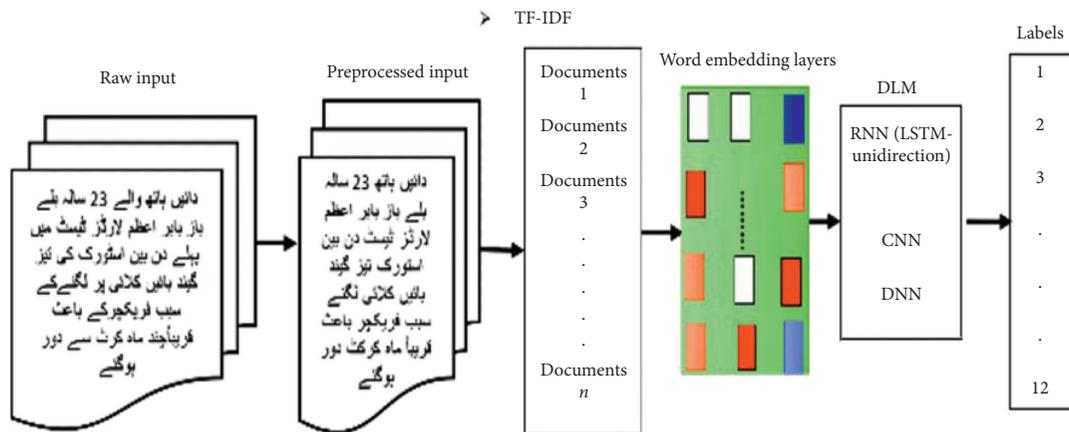


FIGURE 4: Methodology.

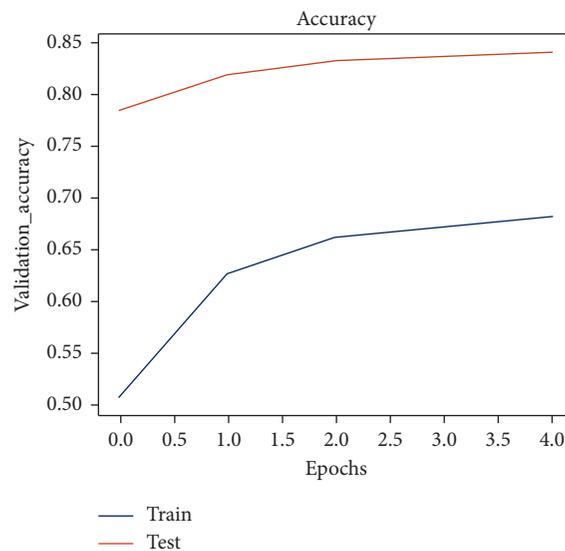


FIGURE 5: DDN's accuracy.

[44] that is developed by using three publicly available Urdu datasets, Wikipedia's Urdu text, another corpus having 90 million tokens [45], and 35 million tokens [46]. It has 102214 unique tokens. Each token comprises 300-dimensional real values. Another model publicly available for research purposes consists of 25925 unique words of Urdu language [47]. Every word has a 400-dimensional value. A word embedding model comprises web-based text, created to classify text. It consists of 64653 unique Urdu words and 300 dimensions for each word.

The journey of research is not over here; to expand our research scope and find the most efficient word embedding model for sentence classification, we decided to develop custom word embedding models. We developed four word embedding models that contain 57251 unique words.

The results of pretrained existing word embedding models are good at the initial level but very low, i.e., 60.26% accuracy. We explored the contents of these models, which revealed that many words are irrelevant and borrowed from other languages, i.e., Arabic and Persian. The contents of Wikipedia are entirely different than news websites that also affect the performance of embedding models. Another major factor, i.e., low amount of data, affected the feature vector generation quality. Stop words in the pretrained word embedding model are not eliminated and considered as a token, while in our dataset all the stop words are removed. It also reduces the size of the vocabulary of the model while generating a feature vector. Therefore, we decided to develop a custom word embedding model on our preprocessed dataset. To postulate the enlargement of the research task, three different word embedding models are developed. The details of all used pretrained word embedding models are given in Table 4.

5.2.3. One-Hot-Encoding. Text cannot be processed directly by machine learning classifiers; therefore, we need to convert the text into a real value. We used one-hot-encoding to convert text to numeric features. For example, the sentences given in Table 5 can be converted into a numeric feature vector using one-hot-encoding as shown in Table 6.

5.2.4. TF-IDF. TF and TF-IDF are feature engineering techniques that transform the text into the numerical format. It is one of the most highly used feature vectors for creating a method for text data. Three deep learning models were evaluated on our corpus. The sequential model with embedding layers outperformed other pretrained word embedding models [44] reported in state of the art [48]. The detailed summary of the evaluation results of CNN, RNN, and DNN is discussed in Section 7.

5.3. Deep Learning Models

5.3.1. Deep Neural Network Architecture. Our DNN architecture consists of three layers, i.e., n -input layer, 150 hidden (dense) layers, and 12 output layers. Feature vector is given as input into a dense layer that is fully connected. The SoftMax activation function is used in the output layer to classify sentences into multiple classes.

5.3.2. Recurrence Neural Network. The recurrence neural network is evaluated using a long short-term memory (LSTM) classifier. RNN consists of embedding, dropout, LSTM, and dense layers. A dictionary of 30000 unique most frequent tokens is made. The sentences are standardized to the same length by using a padding sequence. The dimension of the feature vector is set as 250. RNN showed an overall 81% accuracy that is the second highest in our work.

5.3.3. Convolutional Neural Network (CNN). CNN is a class of deep neural networks that are highly recommended for image processing [49]. It consists of the input layer (embedding layer), multiple hidden layers, and an output layer. There are a series of convolutional layers that convolve with a multiplication. The embedded sequence layer and average layer (GlobalAveragePooling1D) are also part of the hidden layer. The common activation of CNN is RELU Layer. The details of the hypermeters that are used in our problem to train the CNN model are given in Table 7.

5.3.4. Hyperparameters. In this section, all the hyperparameters that are used in our experiments are given in the tabular format. Only those hyperparameters are being discussed here which have achieved the highest accuracy of DNN, RNN, and CNN models. The hyperparameters of DNN that are fine-tuned in our work are given in Table 8.

The RNN model showed the highest accuracy (80.3% and 81%) on two sets of hyperparameters that are given in Table 9. Similarly, Table 7 provides the details of the hyperparameters of the convolutional neural network.

6. Performance Measuring Parameters

The most common performance measuring [41] parameters, i.e., precision, recall, and $F1$ -measure, are used to evaluate the proposed framework. The selection of these parameters was decided because of the multiclass classification and imbalance dataset.

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}, \quad (1)$$

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})}, \quad (2)$$

$$F1 = \frac{2 * (\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}, \quad (3)$$

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})}, \quad (4)$$

where TP, TN, FP, and FN represent total positive, total negative, false positive, and false negative values, respectively. Precision is defined as the closeness of the measurements to each other and recall is the ratio of the total amount of relevant (i.e., TP values) instances that were actually retrieved during the experimental work. It is

TABLE 4: Pretrained word embedding model and custom word embedding model.

Sr. no.	Unique words	Dimension	Window size
<i>Existing pretrained word embedding models</i>			
1 [11]	64653	300	—
2 [19]	102214	100	—
3	53454	300	—
<i>Custom pretrained word embedding models</i>			
1	57251	50	2
2	57251	100	2
3	57251	100	3
4	57251	350	1

TABLE 5: Event sentence.

Urdu sentence	English sentence
علی فٹ بال کھیلتا ہے	Ali plays football
کرونا وائرس نے لاکھوں لوگوں کی جان لے لی۔	Corona virus killed millions of people

TABLE 6: Event sentence converted using one-hot-encoding.

Sentence	کھیلتا	بال	فٹ	علی	جان	لوگوں	لاکھوں	وائرس	کرونا
1	1	1	1	1	0	0	0	0	0
2	0	0	0	0	1	1	1	1	1

TABLE 7: CNN’s hyperparameters.

CNN (79.28%)	
Parameter	Value
Max_words	20000
Batch size	128
Embedding_dim	50
Activation function	SoftMax
Dense_node	256
Training/testing	70%–30%
No. of epochs	20
Loss function	Categorical cross-entropy

TABLE 8: DNN’s hyperparameters.

DNN (79.28%)	
Parameter	Value
Max_words	5000
Batch size	128
Embedding_dim	512
Activation function	SoftMax
Layers	04
Training/testing	70%–30%
No. of epochs	15
Loss function	Sparse categorical cross-entropy

noteworthy that both precision and recall are the relative values of measure of relevance.

7. Results

7.1. Deep Learning Classifiers. The feature vector can be generated using different techniques. The details of feature

TABLE 9: RNN’s hyperparameters.

Parameter	Value
<i>RNN (LSTM) (80.3%)</i>	
Max_words	50000
Batch size	64
Embedding_dim	100
Activation function	SoftMax
Recurrent dropout	0.2
Training/testing	90%–10%
No. of epochs	05
Loss function	Sparse categorical cross-entropy
<i>RNN (LSTM) (81%)</i>	
Max_words	30000
Batch size	128
Embedding_dim	100
Activation function	SoftMax
Recurrent dropout	0.2
Training/testing	80%–20%
No. of epochs	05
Loss function	Sparse categorical cross-entropy

vector generating techniques were discussed in Section 5. The results of feature vector generating techniques that were used in our work, i.e., “multiclass event classification for the Urdu language text,” are given in the proceeding subsections.

7.1.1. Pretrained Word Embedding Models. The convolutional neural network model is evaluated on the features vectors that were generated by all pretrained word embedding models. The summary of all results generated by

TABLE 10: Classification accuracy of the CNN model.

Sr. no.	Existing pretrained model's validation_accuracy	Custom pretrained model's validation_accuracy
1	58.00	36.85
2	60.26	38.04
3	56.68	37.38
4	—	38.68

pretrained [44] and custom pretrained word embedding models is given in Table 10. Our custom pretrained word embedding model that contains 57251 unique tokens, larger dimension size 350, and 1 as the size of a window, showed 38.68% accuracy. The purpose of developing a different custom pretrained word embedding model was to develop a domain-specific model and achieve the highest accuracy. However, the results of both pre-existing pretrained word embedding models and domain-specific custom word embedding models are very low. The detail summary of results can be seen in Table 10.

7.1.2. TF-IDF Feature Vector. DNN architecture consists of an input layer, a dense layer, and a max pool layer. The dense layer is also called a fully connected layer comprised of 150 nodes. SoftMax activation function and sparse_categorical_crossentropy are used to compile the model on the dataset.

25991 instances are used to validate the accuracy of the DNN model. The DNN with connected layer architecture showed 84% overall accuracy for all event classes. The details of the performance measuring parameters for each class of event are given in Table 11. Law and order, the 6th type of event in our dataset, consists of 2000 instances that are used for validation. It showed 66% accuracy that is comparatively low to the accuracy of other types of events. It affected the overall performance of the DNN model. The main reason behind these results is that the sentence of law and order overlaps with the sentences of politics. Generally, sometimes, humans hardly distinguish between law and order and political statements.

For example,

حکومتی وزیر کی غیور ذمہ دارانہ گفتگو خطے کے امن کے لیے خطرہ ہے۔

“The irresponsible talk of state minister is a threat to peace in the region”

The performance of the DNN model is given in Table 11 that showed 84% accuracy for multiple classes of events. All the other performance measuring parameters, i.e., precision, recall, and F1-score, of each class of events are given in Table 11.

The accuracy of the DNN model can be viewed in Figure 5, where the y -axis represents the accuracy and the x -axis represents the number of epochs. RNN achieved 84% accuracy for multiclass event classification.

The expected solution to tackle the sentence overlapping problem with multiple classes is to use a “pre-trained word embedding” model like W2Vec and Glove. However, unfortunately, like the English language, still, there is no open/close domain pretrained word

TABLE 11: Performance measuring parameters for the DNN model.

Class	Precision	Recall	F1-score	Support
1	0.96	0.95	0.96	4604
2	0.91	0.91	0.91	776
3	0.75	0.75	0.75	1697
4	0.78	0.70	0.74	770
5	0.81	0.85	0.83	8424
6	0.71	0.63	0.67	2000
7	1.00	1.00	1.00	817
8	0.92	0.90	0.91	1839
9	0.70	0.70	0.71	2524
10	0.95	0.99	0.97	856
11	0.95	0.99	0.97	741
12	0.82	0.73	0.77	943
Accuracy			0.84	25991
Macro avg	0.84	0.84	0.85	25991
Weighted avg	0.84	0.84	0.84	25991

embedding model that is developed by a large corpus of the Urdu language text.

The RNN sequential model architecture of deep learning is used in our experiments. The recurrent deep learning model architecture consists of a sequence of the following layers, i.e., embedding layer having 100 dimensions, SpatialDropout1D, LSTM, and dense layers. Sparse_categorical_crossentropy loss function has been used for the compilation of the model. Multiclass categorical classification is handled by a sparse categorical cross-entropy loss function instead of categorical cross-entropy. A SoftMax activation function is used at a dense layer instead of the sigmoid function. SoftMax can handle nonlinear classification, i.e., multiple classes, while sigmoid is limited to linear classification and handles binary classification.

A bag-of-words consisting of 30000 unique Urdu language words is used to generate a feature vector. The maximum length of the feature vector is 250 tokens.

The overall accuracy of the RNN model is presented in Table 12 that achieved 81% validation accuracy for our problem by using TF-IDF feature vectors. Other performance evaluation parameters of each class are also given in Table 12.

The accuracy of the RNN model can be viewed in Figure 6, where the y -axis represents the accuracy and the x -axis represents the number of epochs. RNN achieved 81% accuracy for multiclass event classification.

Although CNN is highly recommended for image processing, it showed considerable results for multiclass event classification on textual data. The performance measuring parameters of the CNN classifier are given in Table 13.

The distributed accuracy of the CNN classifier for the twelve classes can be viewed in Figure 7. There is more than one peak (higher accuracies) in Figure 7 that showed datasets are imbalanced.

7.1.3. One-Hot-Encoding. The results of deep learning classifiers are used in our researcher work, and their performance on one-hot-encoding features is presented in Figure 8. The one-hot-encoded feature vectors are given as input to CNN, DNN, and RNN deep learning classifiers. RNN showed better accuracy as compared to CNN while the DNN outperformed

TABLE 12: Performance measuring parameters for the RNN model.

Class	Precision	Recall	F1-score	Support
1	0.95	0.95	0.95	4604
2	0.78	0.77	0.78	776
3	0.70	0.72	0.71	1697
4	0.78	0.64	0.70	770
5	0.78	0.84	0.81	8424
6	0.67	0.57	0.62	2000
7	1.00	1.00	1.00	817
8	0.91	0.87	0.89	1839
9	0.70	0.63	0.66	2524
10	0.93	0.98	0.95	856
11	0.86	0.94	0.90	741
12	0.76	0.67	0.71	943
Accuracy			0.81	25991
Macro avg	0.82	0.80	0.81	25991
Weighted avg	0.81	0.81	0.81	25991

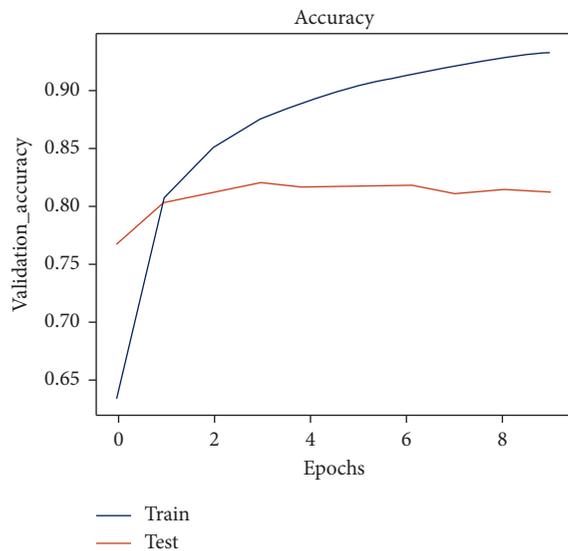


FIGURE 6: RNN's accuracy.

TABLE 13: Performance measuring parameters for the CNN model.

Class	Precision	Recall	F1-score	Support
1	0.96	0.93	0.95	5661
2	0.81	0.65	0.72	967
3	0.72	0.68	0.70	2115
4	0.78	0.54	0.64	878
5	0.73	0.88	0.80	10030
6	0.64	0.51	0.57	2293
7	0.99	0.99	0.99	970
8	0.91	0.86	0.88	2259
9	0.71	0.61	0.66	3044
10	0.93	0.94	0.93	1031
11	0.91	0.82	0.86	889
12	0.77	0.63	0.7	1052
Accuracy			0.80	31189
Macro avg	0.82	0.75	0.78	31189
Weighted avg	0.80	0.80	0.80	31189

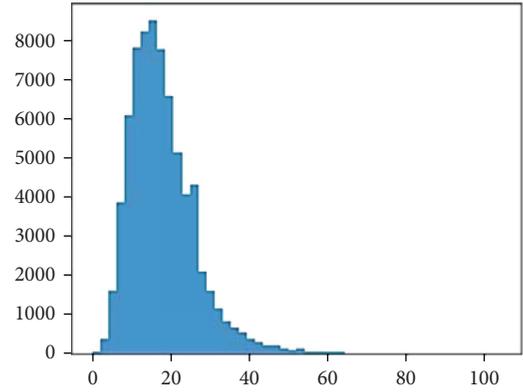


FIGURE 7: CNN's accuracy distribution.

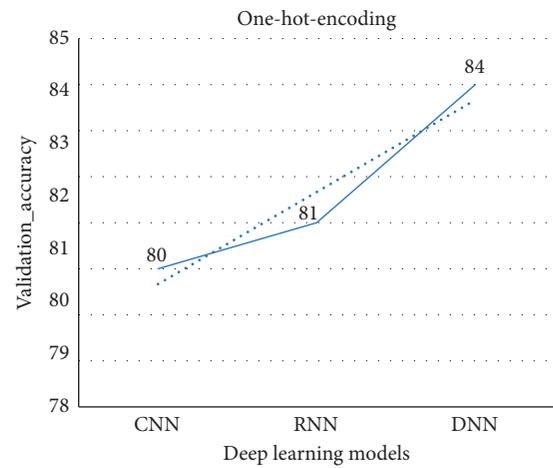


FIGURE 8: CNN, RNN, and DNN accuracy using one-hot-encoding.

among them. RNN and DNN achieved 81% and 84% accuracy, respectively, for multiclass event classification.

7.2. Traditional Machine Learning Classifiers. We also performed a multiclass event classifier by using traditional machine learning algorithms: K-nearest neighbor (KNN), decision tree (DT), Naïve Bayes multinomial (NBM), random forest (RF), linear regression (LR) and support vector machine (SVM). All these models are evaluated using TF-IDF and one-hot encoding features, as feature vectors. It was observed that the results produced using TF-IDF features were better than the results generated using one-hot-encoding features. A detailed summary of the results of the above-mentioned machine learning classifiers is given in the next section.

7.2.1. K-Nearest Neighbor (KNN). KNN performs the classification of a new data point by measuring the similarity distance between the nearest neighbors. In our experiments, we set the value of $k = 5$ that measures the similarity distance among five existing data points [50].

TABLE 14: Performance measuring parameters for the KNN model.

Class	Precision	Recall	F1-score	Support
1	0.91	0.93	0.92	5661
2	0.62	0.83	0.71	967
3	0.67	0.71	0.69	2115
4	0.64	0.60	0.62	878
5	0.78	0.82	0.80	10030
6	0.66	0.50	0.57	2293
7	0.93	1.00	0.96	970
8	0.91	0.80	0.85	2259
9	0.71	0.62	0.66	3044
10	0.85	0.93	0.89	1031
11	0.72	0.85	0.78	889
12	0.75	0.61	0.67	1052
Accuracy			0.78	31189
Macro avg	0.76	0.77	0.76	31189
Weighted avg	0.78	0.78	0.78	31189

TABLE 15: Performance measuring parameters for the DT model.

Class	Precision	Recall	F1-score	Support
1	0.91	0.89	0.90	5661
2	0.83	0.97	0.89	967
3	0.57	0.52	0.54	2115
4	0.58	0.54	0.56	878
5	0.72	0.75	0.73	10030
6	0.44	0.41	0.42	2293
7	0.99	1.00	1.00	970
8	0.79	0.77	0.78	2259
9	0.57	0.55	0.56	3044
10	0.98	0.98	0.93	1031
11	0.86	0.98	0.92	889
12	0.61	0.56	0.58	1031
Accuracy			0.73	31189
Macro avg	0.73	0.74	0.74	31189
Weighted avg	0.73	0.73	0.73	31189

Although the performance of traditional machine learning classifiers is considerable, it must be noted that it is lower than deep learning classifiers. The main performance degrading factor of the classifiers is the imbalanced number of instances and sentences overlapping. The performance of the KNN machine learning model is given in Table 14. It showed 78% accuracy.

7.2.2. Decision Tree (DT). Decision Tree (DT) Decision tree (DT) is a type of supervised machine learning algorithm [51] where the data input is split according to certain parameters. The overall accuracy achieved by DT is 73%, while another performance detail of classes and DT model is given in Table 15.

7.2.3. Naive Bayes Multinomial (NBM). Naïve Bayes multinomial is one of the computational [52] efficient classifiers for text classification but it showed only 70% accuracy that is very low as compared to KNN, DT, and RF. The performance details of all twelve types of classes are given in Table 16.

TABLE 16: Performance measuring parameters for the NB Multinomial model.

Class	Precision	Recall	F1-score	Support
1	0.94	0.91	0.93	5683
2	0.82	0.34	0.48	956
3	0.66	0.47	0.55	2121
4	0.91	0.20	0.32	919
5	0.56	0.95	0.70	10013
6	0.70	0.22	0.34	2387
7	0.98	0.95	0.97	959
8	0.94	0.75	0.83	2188
9	0.75	0.40	0.52	3031
10	0.96	0.78	0.86	998
11	0.96	0.32	0.48	863
12	0.84	0.25	0.39	1071
Accuracy			0.70	31189
Macro avg	0.84	0.54	0.61	31189
Weighted avg	0.76	0.70	0.67	31189

TABLE 17: Performance measuring parameters for the LR model.

Class	Precision	Recall	F1-score	Support
1	0.95	0.94	0.94	5661
2	0.83	0.64	0.72	967
3	0.72	0.69	0.70	2115
4	0.77	0.55	0.64	878
5	0.73	0.88	0.80	10030
6	0.64	0.53	0.58	2293
7	1.00	1.00	1.00	970
8	0.91	0.84	0.88	2259
9	0.73	0.62	0.67	3044
10	0.94	0.92	0.93	1031
11	0.90	0.80	0.85	889
12	0.77	0.66	0.71	1052
Accuracy			0.80	31189
Macro avg	0.82	0.76	0.79	31189
Weighted avg	0.80	0.80	0.80	31189

TABLE 18: Performance measuring parameters for the RF model.

Class	Precision	Recall	F1-score	Support
1	0.94	0.93	0.94	5661
2	0.94	0.96	0.95	967
3	0.72	0.63	0.67	2115
4	0.80	0.58	0.67	878
5	0.71	0.90	0.79	10030
6	0.67	0.41	0.51	2293
7	1.00	1.00	1.00	970
8	0.93	0.80	0.86	2259
9	0.75	0.58	0.65	3044
10	0.94	0.98	0.96	1031
11	0.96	0.98	0.97	889
12	0.84	0.63	0.72	1052
Accuracy			0.80	31189
Macro avg	0.85	0.78	0.81	31189
Weighted avg	0.81	0.80	0.80	31189

7.2.4. Linear Regression (LR). Linear regression is highly recommended for the prediction of continuous output instead of categorical classification [53]. Table 17 shows the

TABLE 19: Performance measuring parameters for the SVM model.

Class	Precision	Recall	F1-score	Support
1	0.84	0.94	0.89	5683
2	0.72	0.43	0.54	956
3	0.72	0.49	0.58	2121
4	0.73	0.43	0.54	919
5	0.64	0.90	0.75	10013
6	0.74	0.24	0.36	2387
7	0.90	0.99	0.94	959
8	0.86	0.78	0.82	2188
9	0.65	0.47	0.57	3031
10	0.85	0.87	0.82	998
11	0.81	0.62	0.70	863
12	0.77	0.63	0.67	1071
Accuracy			0.73	31189
Macro avg	0.77	0.63	0.67	31189
Weighted avg	0.77	0.73	0.71	31189

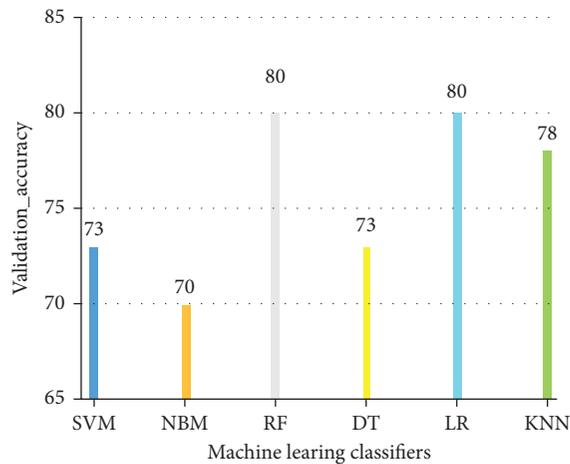


FIGURE 9: Machine learning algorithms' accuracy using TF-IDF.

performance of the LR model, i.e., 84% overall accuracy for multiclass event classification.

7.2.5. Random Forest (RF). It comprises many decision trees [54]. Its results showed the highest accuracy among all evaluated machine learning classifiers. A detailed summary of the results is given in Table 18.

7.2.6. Support Vector Machine (SVM). The support vector machine (SVM) is one of the highly recommended models for binary classification. It is based on statistical theory [55]. Its performance details are given in Table 19.

A comparative depiction of results obtained by the traditional machine learning classifiers is given in Figure 9.

8. Discussion and Conclusion

Lack of resources is a major hurdle in research for Urdu language texts. We explored many feature vectors generating techniques. Different classification algorithms of traditional

machine learning and deep learning approaches are evaluated on these feature vectors. The purpose of performing many experiments on various feature vector generating techniques was to develop the most efficient and generic model of multiclass event classification for Urdu language text.

Word embedding feature generating technique is considered an efficient and powerful technique for text analysis. Word2Vector (W2Vec) feature vectors can be generated by pretrained word embedding models or using dynamic parameters in embedding layers of deep neural networks. We performed sentence classification using pretrained word embedding models, one-hot-encoding, TF, TF-IDF, and dynamic embeddings. The results of the rest of the feature vector generating techniques are better than pretrained word embedding models.

Another argument in support of this conclusion is that only a few pretrained word embedding models exist for Urdu language texts. These models are trained on considerable tokens and domain-specific Urdu text. There is a need to develop generic word embedding models for the Urdu language on a large corpus. CNN and RNN (LSTM) single-layer architecture and multilayer architecture do not affect the performance of the proposed system.

Experimental results are the vivid depiction that the one-hot-encoding method is better than the word embedding model and pretrained word embedding model. However, among all mentioned (see Section 5.2) feature generating techniques, TF-IDF outperformed. It showed the highest accuracy (84%) by using DNN deep learning classifier, while event classification on an imbalance dataset of multiclass events for Urdu language using traditional machine learning classifiers showed considerable performance but lower than deep learning models. Deep learning algorithms, i.e., CNN, DNN, and RNN, are preferable over traditional machine learning algorithms, because there is no need for a domain expert to find relevant features in deep learning like traditional machine learning. DNN and RNN outperformed among all other classifiers and showed overall 84% and 81% accuracy, respectively, for the twelve classes of events. Comparatively, the performance of CNN and RNN is better than Naïve Bayes and SVM.

Multiclass event classification at the sentence level performed on an imbalance dataset; events that are having a low number of instances for a specific class affect the overall performance of the classifiers. We can improve the performance by balancing the instances of each class. The following can be concluded:

- (1) Pretrained word embedding models are suitable only for sentence classification if pretrained models are developed by an immense amount of textual data
- (2) Existing word embedding models Word2Vec and Glove that were developed for the English language text are incompatible for Urdu language text
- (3) In our case, TF-IDF, one-hot-encoding, and dynamic embedding layer are better feature generating

techniques as compared to pre-existing Urdu language text word embedding models

- (4) The TF-IDF-based feature vectors showed the highest results, as compared to one-hot-encoding-and dynamic word embedding-based feature vectors
- (5) Imbalance number of instances in the dataset affected the overall accuracy

9. Future Work

In a comprehensive review of Urdu literature, we found only a few numbers of referential works related to Urdu text processing. The main hurdle in Urdu exploration is the unavailability of the processing resources, i.e., event dataset, close-domain part-of-speech tagger, lexicons, annotators, and other supporting tools.

There are a lot of tasks that can be accomplished for Urdu language text in the future. Some of those are mentioned as follows:

- (1) Generic word embedding models can be developed for a large corpus of Urdu language text
- (2) Different deep learning classifiers can be evaluated, i.e., BERT and ANN
- (3) Event classification can be performed at the document level
- (4) A balance dataset can be used for better results
- (5) Multilabel event classification can be performed in the future
- (6) Unstructured data of Urdu text can be classified into different event classes
- (7) Classification of events for the Urdu language can be further performed for other domains of knowledge, i.e., literacy ratio, top trends, famous foods, and a religious event like Eid
- (8) Contextual information of sentence, i.e., presence and postsentence information, certainly plays a vital role in enhancing the performance accuracy of the classification model
- (9) Event classification can be performed on a balanced dataset
- (10) Unstructured Urdu data can be used for event classification
- (11) Classification can be performed at a document and phrase level

Data Availability

The data used to support this study are available at <https://github.com/unique-world/Multiclass-Event-Classification-Dataset>.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

- [1] A. Lenhart, R. Ling, S. Campbell, and K. Purcell, *Teens and Mobile Phones: Text Messaging Explodes as Teens Embrace it as the Centerpiece of Their Communication Strategies with Friends*, Pew Internet & American Life Project, Washington, DC, USA, 2010.
- [2] M. Motoyama, B. Meeder, K. Levchenko, G. M. Voelker, and S. Savage, "Measuring online service availability using twitter," *WOSN*, vol. 10, p. 13, 2010.
- [3] J. Rogstadius, M. Vukovic, C. A. Teixeira, V. Kostakos, E. Karapanos, and J. A. Laredo, "CrisisTracker: crowdsourced social media curation for disaster awareness," *IBM Journal of Research and Development*, vol. 57, no. 5, pp. 4–1, 2013.
- [4] T. Reuter and P. Cimiano, "Event-based classification of social media streams," in *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, pp. 1–8, Bielefeld, Germany, June 2012.
- [5] K. Sailunaz and R. Alhaji, "Emotion and sentiment analysis from Twitter text," *Journal of Computational Science*, vol. 36, Article ID 101003, 2019.
- [6] P. Capet, T. Delavallade, T. Nakamura, A. Sandor, C. Tarsitano, and S. Voyatzi, "A risk assessment system with automatic extraction of event types," in *Proceedings of the International Conference on Intelligent Information Processing*, pp. 220–229, Springer, Beijing, China, October 2008.
- [7] F. Hogenboom, F. Frasinca, U. Kaymak, F. De Jong, and E. Caron, "A survey of event extraction methods from text for decision support systems," *Decision Support Systems*, vol. 85, pp. 12–22, 2016.
- [8] S. Jiang, H. Chen, J. F. Nunamaker, and D. Zimbra, "Analyzing firm-specific social media and market: a stakeholder-based event analysis framework," *Decision Support Systems*, vol. 67, pp. 30–39, 2014.
- [9] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.
- [10] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.
- [11] T. Mikolov, Q. V. Le, and I. Sutskever, "Exploiting similarities among languages for machine translation," 2013, <http://arxiv.org/abs/1309.4168>.
- [12] R. Alghamdi and K. Alfalqi, "A survey of topic modeling in text mining," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 6, no. 1, 2015.
- [13] D. M. Eberhard, S. F. Gary, and C. D. Fennig, *Ethnologue: Languages of the World*, SIL International, Dallas, TX, USA, 2019.
- [14] A. Daud, W. Khan, and D. Che, "Urdu language processing: a survey," *Artificial Intelligence Review*, vol. 47, no. 3, pp. 279–311, 2017.
- [15] 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, 2021.
- [16] U. Pal and A. Sarkar, "Recognition of printed Urdu script," in *Proceedings of the 2003 Seventh International Conference on Document Analysis and Recognition*, pp. 1183–1187, IEEE, Edinburgh, Scotland, August 2003.
- [17] Y. Yang, T. Pierce, and J. Carbonell, "A study of retrospective and on-line event detection," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and*

- development in information retrieval*, pp. 28–36, Melbourne, Australia, August 1998.
- [18] T. Kala, “Event detection from text data,” *Computational Intelligence*, vol. 31, pp. 132–164, 2015.
- [19] M. Naughton, N. Stokes, and J. Carthy, “Sentence-level event classification in unstructured texts,” *Information Retrieval*, vol. 13, no. 2, pp. 132–156, 2010.
- [20] G. Jacobs, E. Lefever, and V. Hoste, “Economic event detection in company-specific news text,” in *Proceedings of the First Workshop on Economics and Natural Language Processing*, pp. 1–10, Melbourne, Australia, July 2018.
- [21] E. D’Andrea, P. Ducange, A. Bechini, A. Renda, and F. Marcelloni, “Monitoring the public opinion about the vaccination topic from tweets analysis,” *Expert Systems with Applications*, vol. 116, pp. 209–226, 2019.
- [22] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.
- [23] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Proceedings Advances Neural Information Processing Systems*, vol. 26, pp. 3111–3119, Lake Tahoe, NV, USA, December 2013.
- [24] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.
- [25] 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 Information Systems*, pp. 1–26, 2020.
- [26] G. Liu and J. Guo, “Bidirectional LSTM with attention mechanism and convolutional layer for text classification,” *Neurocomputing*, vol. 337, pp. 325–338, 2019.
- [27] M. Sharjeel, R. M. A. Nawab, and P. Rayson, “COUNTER: corpus of Urdu news text reuse,” *Language Resources and Evaluation*, vol. 51, no. 3, pp. 777–803, 2017.
- [28] K. Mehmood, D. Essam, and K. Shafi, “Sentiment analysis system for Roman Urdu,” in *Proceedings of the 2018 Science and Information Conference*, pp. 29–42, Springer, Casablanca, Morocco, July 2018.
- [29] K. Ahmed, M. Ali, S. Khalid, and M. Kamran, “Framework for Urdu news headlines classification,” *Journal of Applied Computer Science & Mathematics*, vol. 10, no. 1, pp. 17–21, 2016.
- [30] Z. Tehseen, M. P. Akhter, and Q. Abbas, “Comparative study of feature selection approaches for Urdu text categorization,” *Malaysian Journal Computer Science*, vol. 28, no. 2, pp. 93–109, 2015.
- [31] W. Yin and L. Shen, “A short text classification approach with event detection and conceptual information,” in *Proceedings of the 2020 5th International Conference on Machine Learning Technologies*, pp. 129–135, Beijing, China, June 2020.
- [32] H. Zhou, M. Huang, T. Zhang et al., “Emotional chatting machine: emotional conversation generation with internal and external memory,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA, February 2018.
- [33] A. Hassan and A. Mahmood, “Convolutional recurrent deep learning model for sentence classification,” *IEEE Access*, vol. 6, pp. 13949–13957, 2018.
- [34] T. Zia, M. P. Akhter, and Q. Abbas, “Comparative study of feature selection approaches for Urdu text categorization,” *Malaysian Journal of Computer Science*, vol. 28, no. 2, pp. 93–109, 2015.
- [35] A. R. Ali and M. Ijaz, “Urdu text classification,” in *Proceedings of the 7th international conference on frontiers of information technology*, pp. 1–7, Abbottabad, Pakistan, December 2009.
- [36] M. Usman, Z. Shafique, S. Ayub, and K. Malik, “Urdu text classification using majority voting,” *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 8, pp. 265–273, 2016.
- [37] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A convolutional neural network for modelling sentences,” 2014, <http://arxiv.org/abs/1404.2188>.
- [38] D. M. Awais and D. M. Shoaib, “Role of discourse information in Urdu sentiment classification,” *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 18, no. 4, pp. 1–37, 2019.
- [39] J. P. Singh, Y. K. Dwivedi, N. P. Rana, A. Kumar, and K. K. Kapoor, “Event classification and location prediction from tweets during disasters,” *Annals of Operations Research*, vol. 283, no. 1–2, pp. 737–757, 2019.
- [40] R. C. Paulo, D. Fillipe, and M. S. C. Sergio, “Classification of events on social media,” 2016.
- [41] Q. A. Al-Radaideh and M. A. Al-Aburat, “An Arabic text categorization approach using term weighting and multiple reducts,” *Soft Computing*, vol. 23, no. 14, pp. 5849–5863, 2019.
- [42] J. F. Allen, “Maintaining knowledge about temporal intervals,” *Communications of the ACM*, vol. 26, no. 11, pp. 832–843, 1983.
- [43] T. Joachims, “Text categorization with support vector machines: learning with many relevant features,” in *Proceedings of the European conference on machine learning*, pp. 137–142, Springer, Chemnitz, Germany, April 1998.
- [44] S. Haider, “Urdu word embeddings,” in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018.
- [45] B. Jawaid, A. Kamran, and O. Bojar, “Urdu monolingual corpus,” *LINDAT/CLARIN Digital Library at the Institute of Formal and Applied Linguistics*, Charles University, Prague, Czechia.
- [46] F. Adeeba, Q. Akram, H. Khalid, and S. Hussain, “CLE Urdu books n -grams,” in *Proceedings of the Conference on Language and Technology, CLT 14*, Karachi, Pakistan, May 2014.
- [47] A. Hassan and A. Mahmood, “Deep learning for sentence classification,” in *Proceedings of the 2017 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, pp. 1–5, IEEE, New York, NY, USA, May 2017.
- [48] D.-X. Zhou, “Universality of deep convolutional neural networks,” *Applied and Computational Harmonic Analysis*, vol. 48, no. 2, pp. 787–794, 2020.
- [49] M. V. Valueva, N. N. Nagornov, P. A. Lyakhov, G. V. Valuev, and N. I. Chervyakov, “Application of the residue number system to reduce hardware costs of the convolutional neural network implementation,” *Mathematics and Computers in Simulation*, vol. 177, pp. 232–243, 2020.
- [50] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, “KNN model-based approach in classification,” in *Proceedings of the OTM Confederated International Conferences “On the Move to Meaningful Internet Systems”*, Catania, Italy, November 2003.
- [51] Y. Zhong, “The analysis of cases based on decision tree,” in *Proceedings of the 2016 7th IEEE international conference on software engineering and service science (ICSESS)*, pp. 142–147, IEEE, Beijing, China, August 2016.
- [52] S. Xu, “Bayesian Naïve Bayes classifiers to text classification,” *Journal of Information Science*, vol. 44, no. 1, pp. 48–59, 2018.
- [53] T. Zhang and F. Oles, “Text categorization based on regularized linear classification methods,” *Information Retrieval*, vol. 4, no. 1, pp. 5–31, 2001.

- [54] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random forests and decision trees," *International Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 5, p. 272, 2012.
- [55] Y. Zhang, "Support vector machine classification algorithm and its application," in *Proceedings of the International Conference on Information Computing and Applications*, pp. 179–186, Springer, Bhubaneswar, India, September 2012.