IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 7, 2020, accepted February 20, 2020, date of publication February 27, 2020, date of current version March 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2976744

Document-Level Text Classification Using Single-Layer Multisize Filters Convolutional Neural Network

MUHAMMAD PERVEZ AKHTER[1], **ZHENG JIANGBIN**[1], **IRFAN RAZA NAQVI**[1], **MOHAMMED ABDELMAJEED**[2], **ATIF MEHMOOD**[3], **AND MUHAMMAD TARIQ SADIQ**[4]

[1]School of Software and Microelectronics, Northwestern Polytechnical University, Xian 710072, China
[2]School of Computer Science and Technology, Northwestern Polytechnical University, Xian 710072, China
[3]School of Artificial Intelligence, Xidian University, Xian 710071, China
[4]School of Automation, Northwestern Polytechnical University, Xian 710072, China

Corresponding author: Muhammad Pervez Akhter (pervezbcs@gmail.com)

This work was supported in part by the Research and Development Plan of Shaanxi Province under Grant 2017ZDXM-GY-094 and Grant 2015KTZDGY04-01, and in part by the National Natural Science Foundation of China under Grant 61972321.

ABSTRACT The rapid growth of electronic documents are causing problems like unstructured data that need more time and effort to search a relevant document. Text Document Classification (TDC) has a great significance in information processing and retrieval where unstructured documents are organized into pre-defined classes. Urdu is the most favorite research language in South Asian languages because of its complex morphology, unique features, and lack of linguistic resources like standard datasets. As compared to short text, like sentiment analysis, long text classification needs more time and effort because of large vocabulary, more noise, and redundant information. Machine Learning (ML) and Deep Learning (DL) models have been widely used in text processing. Despite the major limitations of ML models, like learn directed features, these are the favorite methods for Urdu TDC. To the best of our knowledge, it is the first study of Urdu TDC using DL model. In this paper, we design a large multi-purpose and multi-format dataset that contain more than ten thousand documents organize into six classes. We use Single-layer Multisize Filters Convolutional Neural Network (SMFCNN) for classification and compare its performance with sixteen ML baseline models on three imbalanced datasets of various sizes. Further, we analyze the effects of preprocessing methods on SMFCNN performance. SMFCNN outperformed the baseline classifiers and achieved 95.4%, 91.8%, and 93.3% scores of accuracy on medium, large and small size dataset respectively. The designed dataset would be publically and freely available in different formats for future research in Urdu text processing.

INDEX TERMS Convolutional neural network, deep learning, machine learning, natural language processing, text document classification, Urdu text classification.

I. INTRODUCTION

The rapid growth of electronics text documents on internet, World Wide Web (WWW), news blogs, and digital libraries by organizations, researchers, news media, and institutions is causing problems like a large volume of unstructured data. Organizing and searching such a large and unstructured dataset manually is almost impossible. Automatic processing, organizing and handling such textual data is a fundamental problem of information processing and retrieval. Text categorization or classification (TC) is an automatic mapping of text to a set of predefine labels (classes). It has many important applications like sentiment analysis [1], [2], sentence classification [3], and document classification [4], [5].

Text Document Classification (TDC) is the most important and a fundamental task of TC. In TDC, a document from a set of documents is assigned a label automatically from a set of pre-defined labels based on its contents. Formally, if $d_i \in D$ is a document from a set of text documents D: $D = \{d_0, d_1, d_2, \ldots, d_n\}$ and $c_i \in C$ is a label from a set of labels C: $C = \{c_0, c_1, c_2, \ldots, c_n\}$ then TDC is a task of assigning d_i to c_i: $<d_i, c_i>$. TDC helps to structure an unstructured dataset efficiently and accurately. Machine Learning (ML) and Deep Learning (DL) are the two most widely used automated methods for TDC.

As compared to short text classification [6] like spam [7], news headlines [8] and reviews, designing a classification

The associate editor coordinating the review of this manuscript and approving it for publication was Mostafa Rahimi Azghadi.

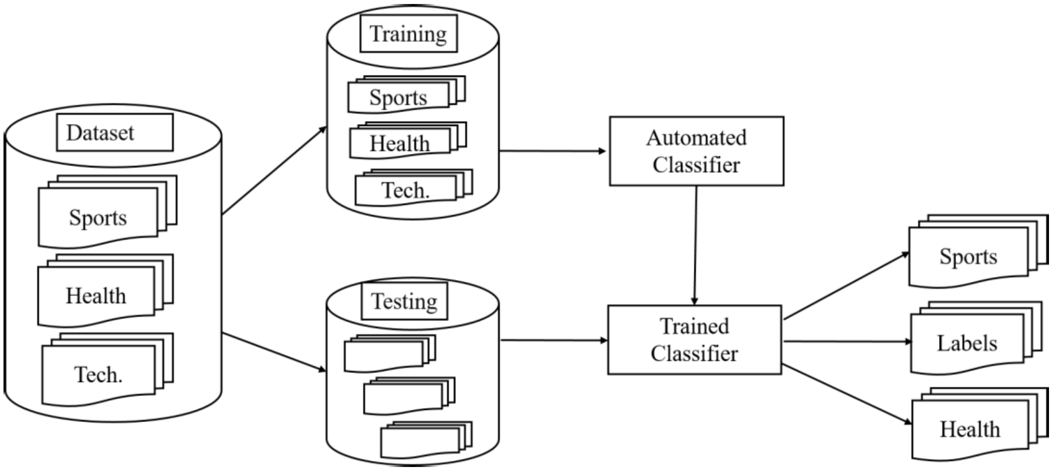


FIGURE 1. Automated text documents classification.

system for long text documents is more challenging, complex and computationally expensive. A text document may contain a sentence, a paragraph or a long text of thousands of words. A long text document usually has a large vocabulary size, more noise and redundant information as compared to short text. Similarly, multiple paragraphs within the same document semantically belong to multiple categories [4], [5].

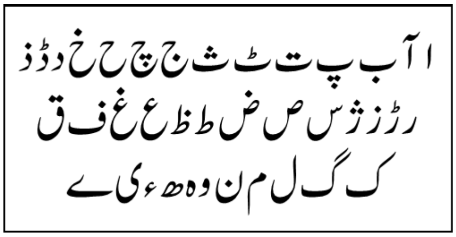
The Urdu language has a worldwide appeal. Urdu is a national language of Pakistan. It is also an official language of six states of India. It has more than 300 million speakers all over the world [9]. In the past, researchers neglected Urdu because of its complex morphology, unique characteristics and the lack of linguistic resources [10]. Because of these characteristics, text document classification of Urdu language is more complicated and challenging task than other languages. Further, automatic TDC systems designed for other languages can not be used for Urdu. From the last decade, like other languages, Urdu text documents on WWW, blogs, online libraries, and news articles are increasing rapidly. In short, all these are causing to grow the interest of researchers in TDC of Urdu language. Therefore designing and implementation of some efficient and accurate automatic TDC system for the Urdu language is very imperative.

Traditional TDC systems are based on ML methods. ML methods do not perform well on large datasets and can learn directed features only. The main challenge of ML methods is to select efficient features from high dimensional feature spaces [11] but there is no universal feature selection method that works well with all type of classifiers [12]. For Urdu language TDC, only a few studies have been performed using ML [12]–[14]. Unfortunately, the potential of DL models have not been explored for TDC of Urdu language. As compared to ML models, DL models are capable to learn complex features implicitly from high dimensional feature space and are faster than ML methods because DL models use Graphic Processing Units (GPUs) for parallel processing of data [15], [16].

A Convolutional Neural Network (CNN) is a DL model that use convolving filters in convolutional layer to extract high-level features. The major challenge in CNN is to find out the number of filters and the appropriate size of these filters for a specific task. Filters of large size make training hard while filters of small size may provide inaccurate results [17]. In this study, we use a Single-layer Multisize Filters Convolutional Neural Network (SMFCNN) model to classify text documents of Urdu language. A major advantage of SMFCNN is that it consists of multiple filters of various sizes which are applied to multiple window of various sizes to extract variable-length features (n-grams) from text document [3] (see section IV).

A common process of automated TDC system is shown in Fig. 1. A dataset is consist of multiple text documents of variable length. Usually, it is divided into two subsets: training and testing. The former is used to train the model while the latter is used to test its performance on unseen documents. Which split-ratio should be used to divide a dataset into training and testing subsets? For Urdu language TDC, no study investigated this problem. Researchers used different but randomly selected split-ratio to split their datasets. It decreased the performance of the classifier because of an insufficient number of documents in training or testing subset [18]. In this study, we investigate this problem on three datasets of different sizes. We explore the effects of various split-ratios of a dataset in the performance of SMFCNN.

A dataset may contain hundreds or thousands of text documents, also called a raw dataset. Raw dataset must be preprocessed before training a model on it because (1) it contains noise that degrades classifier performance, and (2) it increases learning time of a model. Preprocessing steps like tokenize text, eliminate none-language characters, remove stopwords, and stemming are performed on raw data. In preprocessing of Urdu text, it is a common practice to remove stopwords (frequent words) but rare words (infrequent words) are not removed. Different studies show that removing rare



ا آ ب پ ت ٹ ث ج چ ح خ د ڈ ذ
ر ڑ ز ژ س ص ض ط ظ ع غ ف ق
ک گ ل م ن و ہ ھ ء ی ے

FIGURE 2. The basic character set of Urdu.

TABLE 1. Examples of words, ligatures, and characters of Urdu language.

Urdu Words	Urdu Ligatures	Urdu Characters
پاکستان (*pakistan,* Pakistan)	پا، کستا، ن	پ ، ا ، ک، س، ت، ا، ن
سکول (*sakool*, School)	سکو، ل	س، ک، و، ل
کتاب (*kita*b, Book)	کتا، ب	ک، ت، ا، ب

words in preprocessing has a confounding effect in classification [19]–[21]. In this study, we investigate the effects in the performance of SMFCNN after removing stopwords and rare words from Urdu text.

A. URDU LANGUAGE AND ITS FEATURES

Urdu has 38 basic characters as shown in Fig. 2. Urdu is written in Nastalique font style that is a very complex and context-sensitive style. A word of Urdu is a combination of ligatures and a ligature is composed of a single or several characters. Words are joined from right-to-left order to form a sentence. Table 1 shows the composition of four words, their ligatures, and characters separated by a comma for understanding.

Features of Urdu language make it more complex for automated text processing as compared to other languages. Some of the important features are discussed below:

- *No capitalization*: unlike English, there are no uppercase or lowercase characters in Urdu. Therefore, it is difficult to identify proper nouns and start of a sentence in Urdu. For example, in ‘‘وہ پاکستان میں رہتا ہے۔’’ (*wo Pakistan me rehta ha*, He lives in Pakistan) sentence, nouns and the start of a sentence cannot be identified.
- *Right-to-left:* the direction of Urdu writing is from right-to-left. For example, ‘‘اردو پاکستان کی قومی زبان ہے۔’’ ‘اردو’ is the first word and ‘ہے’ is the last word of a sentence.
- *Diacritics:* like Arabic, Urdu text may have few diacritics. Zer (◌ِ) meaning ‘‘under’’, zabar (◌َ) meaning ‘‘over’’ and pesh (◌ُ) meaning ‘‘in front’’ are three common diacritics. Examples are: گنا (Ganna, sugarcane) and گُنا (Gunna, number of times); بِکری (Bekri, daily transaction) and بَکری (Bakri, goat).
- *Free word order:* order of words in a sentence of Urdu may be different but the meaning would be the same. For example, in ‘‘علی نے بازار سے نئی کتاب خریدی۔’’ (*ali ne bazar se nai kitab kharidi*) or ‘‘علی نے نئی کتاب بازار سے خریدی۔’’ (*ali ne nai kitab bazar se kharidi*) both sentences have different words order but the meaning is same as ‘‘Ali bought a new book from the market’’.
- *Subject-Object-Verb (SOV):*unlike English, the sentence structure of Urdu is in subject-object-verb order. For example, ‘‘جنید سکول گیا۔’’ (*junaid school gia*, Junaid went to school). جنید (*junaid*, Junaid) is a subject. سکول (*sakool*, school) is an object. گیا (*giya*, went) is a verb.
- *Context Sensitivity:* Urdu is a context-sensitive language. A character may have different shapes when joined with other characters to form a ligature or a word. For example, the character ‘ت, when joined with ‘ا’, it is ‘تا’ (ت+ا). Similarly, when joined with ‘ب’, it is ‘تب’ (ت+ب).

B. OUR CONTRIBUTION

Urdu is a resource-poor language but it has a rich and complex morphological script that makes it more challenging for automatic text processing. Unavailability of large, standard, public, and free of cost datasets of long text documents is a major obstacle for Urdu language TDC. In the past, TDC of Urdu has been performed using ML models but the potential of DL models have not been explored. Long TDC using DL models, and comparatively analyze the performance of DL model with ML models are the main gaps in Urdu text classification. To fill these gaps, our contribution in the study is to:

- Design a large dataset of Urdu text documents and make this dataset publically available for future research
- Classify the designed dataset using a Single-layer Multi-size Filters Convolutional Neural Network (SMFCNN)
- Analyze the impact on the accuracy of fine-tuning the hyperparameters of the model
- Analyze the performance of SMFCNN on three imbalanced datasets of small, medium and large size
- Analyze the various dataset split-ratios and their impacts on network performance
- Evaluate the effects of stopwords and rare words of Urdu on the performance of SMFCNN
- Compare the performance of SMFCNN with well-known ML classifiers

The organization of the paper is as follow: related work is discussed in section II. Section III explains the difference in ML and DL methods for text classification. A brief summary of baseline models is also given in section III. Section IV explains the SMFCNN model and a comparison of proposed dataset with two benchmark datasets. Section V gives detail about hyperparameter settings and performance measures. Section VI includes results and discussions. Section VII has conclusions and future work.

II. RELATED WORK

Classification of Urdu text documents using feature selection methods and classifiers of ML has been performed from the last decade [1], [8], [12], [14]. Text documents are preprocessed before given as input to the classifier. The accuracy of the classifier can be improved by choosing appropriate preprocessing methods on a dataset [22]. The results of a classifier are changed if the dataset is the same but the preprocessing methods are different. For TDC, studies concluded that a classifier cannot perform well on raw data and data must be preprocessed to achieve better performance [13], [23].

In the dataset, stopwords are the most frequent words while the most infrequent words are called rare words. Both are considered not useful for classification. In text processing, usually, stop words are removed while rare words are neglected [2], [24], [25]. Different studies show that removing rare words in preprocessing has a confounding effect in automatic classification [19]–[21].

After dataset preprocessing, feature selection methods can affect the performance of the classifier [11], [26]. A dataset contains thousands of features (high dimension) and only a portion of it is useful for classification. Conversion of the high dimensional feature space into low dimensional feature space is a major problem in ML methods. Tehseen performed a comparative analysis of feature selection methods and ML classifiers on Urdu TC. Experimental results shows that none of the feature selection method is dominated with all classifiers [12]. Finding the best feature selection method is one of the main problems in TC using ML [11]. In contrast to ML, the DL models consist of multiple algorithms, which automatically select the features directly from the dataset for classification. DL models do not need domain-specific knowledge to extract the features [27].

A drawback of ML classifiers is that they perform well with limited numbers of features. Classification of long text documents, the comparative study of [12] shows that maximum performance of SVM and KNN was achieved using 1000 and 200 features respectively. The performance started to decrease when the numbers of features were increased from a specific value.

A dataset of Urdu text documents was designed by collecting different news articles from online news blogs and manually arranged them into six categories [13]. Similarly, [28] designed a dataset of Urdu news articles for web content opinion mining. Usman also collected online news articles of Urdu text, saved them into text files and manually arranged them into eight categories [14]. A text document in a dataset can be of two formats: XML format and Unicode format [10], [29]. The XML format is easy to convert into another format like importing into a database while the Unicode format is easy to process by applications. Programming languages like Python and applications like WEKA can also process a dataset in a tabular or matrix form. Comma-Separated Values (CSV) files are a popular choice for such representation. In many studies of TDC of other languages, datasets were designed by collecting online news articles like Arabic [23], Chinese [30], Indonesian [25] and Turkish [31]. The designed dataset in this study is also collected from online news articles and is publically available in different formats.

A major obstacle in Urdu text document processing is the lack of big, free of cost, standard and public datasets. In different studies, researchers designed their own datasets and are not publically available [13], [14]. In literature, only the EMILLE[1] and COUNTER[2] [32] datasets are publically available and can be downloaded free of cost. Text documents in both datasets are also collected from multiple news websites. EMILLE contained only a few documents and two categories contained one document in it. COUNTER has only twelve hundred documents. Because of their small size, these datasets are not suitable for long text TDC using DL models. To fill this gap, we design a large dataset for Urdu TDC.

High dimensional feature space and class imbalanced dataset are two major challenges for both ML and DL models. A dataset is an imbalanced dataset when one of the classes dominates it. A class with more number of documents is called a major class while a class with fewer documents is called a minor class. This problem becomes more severe when an imbalanced dataset has a high dimensional feature space [33]. For classification, none of the algorithms of ML is superior on all benchmark datasets because of two reasons (1) imbalance dataset have variations in their imbalance ratio, and (2) during learning the classifier performance is different on different datasets. A good review of imbalance-class dataset processing, methods and applications can be found in [34].

Fine-tuning of hyperparameters of DL models helps to improve the performance of these models. Fine-tuning means to find out the best values of these parameters on which the model can achieve the highest performance and minimize the loss. Kim fine-tuned CNN parameters for sentence classification [3]. Zhou used hyperparameters tuning before TC using C-LSTM network [35]. Abandah also tuned hyperparameters of recurrent neural networks (RNN) for Arabic text processing [36]. Tehseen used SVM and KNN for Urdu TC after tuning the hyperparameters of these classifiers [12]. We also fine-tuned and analyzed the hyperparameters of our model for TDC of Urdu language. Comparison of various studies regarding Urdu text classification is given in Table 2.

III. MACHINE LEARNING AND DEEP LEARNING MODELS

A. MACHINE LEARNING CLASSIFICATION

ML is a subfield of artificial intelligence in which a model learns a specific task from some examples and then performs the learned task on unknown examples. For TDC using ML, major challenges are to reduce the dimensionality of a high dimensional feature space by applying feature selection methods and also to find a classifier that can learn the high dimensional feature space to classify unknown documents accurately. Basic steps of text classification using ML models are discussed below:

[1] http://ota.ox.ac.uk/scripts/download.php?approval=9d5c5288a573453a422f

[2] http://www.research.lancs.ac.uk/portal/files/127219125/COUNTER.zip

TABLE 2. Summary of the literature regarding Urdu text classification.

Approach	Limitations	Description
SVM	Proposed framework did not compare with other classifiers. Headlines of the news were considered while description was ignored. Dataset is not publically available.	Classify news headlines [8]
NB, J48, SVM, KNN	EMILLE has only few documents and is not suitable for classification with naïve dataset	Compared four classifiers and five feature selection methods using two datasets [12]
SVM, NB	Dataset is not publically available for future research. High frequency terms were removed but rare terms did not considered.	Classify a dataset of news articles. SVM outperforms NB [13]
NB, SGD, Random Forest	Eliminated stop words only and dataset is also not publically available.	Final decision was made using majority voting [14]
Lexicon approach	Urdu	Develop lexicon approach for mining contents of web pages [28]
	Designed dataset is not publically available. Dataset was not classified using automatic methods of machine or deep learning	Design a corpus of Urdu news articles [29]
Naïve Bayes, N-grams	Not designed for text classification task.	COUNTER - 1200 news articles distributed in five classes [32]

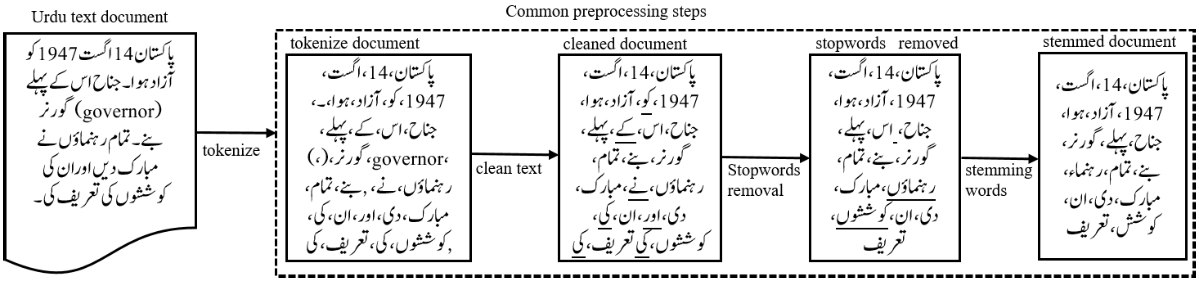


FIGURE 3. Common preprocessing steps performed for the classification of Urdu text document.

1) PREPROCESSING OF TEXT DOCUMENTS

After dataset design, the dataset is preprocessed to make it suitable for automated processing. For Urdu language TDC, common preprocessing steps are shown in Fig. 3. A document is tokenized using space or punctuation symbols [14], [37]. Non-language characters, special symbols, numeric values, and URLs are removed so that a document contains only words of the target language. Stopwords are removed and remaining words are shaped to its root word using stemming. A vector representation of a document is obtained usually term frequency and inverse document frequency methods. These vectors are further used for feature selection and classification [38].

2) DIMENSIONALITY REDUCTION AND FEATURE SELECTION

High dimensional feature space is a major hurdle for effective text classification because it makes a classifier computationally intractable and inefficient. Not all the words in a dataset are useful for TDC. Some words are more valuable, some are less valuable and others are non-valuable. Set of all the words constitute a feature space where the number of words in the feature space is called its dimensions. Feature selection methods reduced the dimensionality of a feature space by selecting a more valuable feature. The benefits of dimensionality reduction are reduced feature space, reduced training time, improved classification accuracy, and helped to avoid overfitting a classifier [11]. Information Gain (IG), Gain Ratio (GR) are some popular feature selection methods. Unfortunately, there is no universal feature selection method which can work well with all type of ML classifiers [12].

3) MACHINE LEARNING CLASSIFIERS

In the past, different studies used a variety of ML classifiers for Urdu language TDC. In this section, a short description of the baseline classifiers of ML is given to compare their performance with the SMFCNN model on long TDC of Urdu language. Detail description of these classifiers is available in the documentation of WEKA [39]. A summary of the baseline classifier is given below.

- *k-Nearest Neighbors (k-NN):* assigns a label to an instance based on the labels of its k-nearest neighbors. Its performance is based on the value of k and the similarity measure. For large datasets, it is computationally expensive.

- *Decision Tree (DT):* nodes represent attributes of an example with its importance to classify it. Leave nodes to represent classes. Easy to interpret but complex and time-consuming for high dimensional dataset.
- *Support Vector Machine (SVM):* learn n-dimensional hyperplane that separates examples into classes. It can classify both linear and non-linear data. High memory and poor interpretability are its drawbacks. SVM used kernels for pattern analysis. For degree "d," the polynomial kernel can be defined as:

$$K(x_i, x_j) = \{x_i^T x_j + c\}^d \tag{1}$$

where x_i and x_j are the input space vector and x^T is the transpose of x_i. c is a parameter used for the trade-off between the highest order and lowest order polynomial. Radial Basis Function (RBF) is a real-valued function, whose value depends upon the distance from the origin. RBF kernel can be defined as follows:

$$K\left(x_i, x_j\right) = \exp(-\gamma(x_i - x_j)^2) \quad for\ \gamma > 0 \tag{2}$$

where the value of γ can be used as $1/2\sigma^2$ where σ^2 is the variance of input data. The sigmoid kernel function can be defined as:

$$K\left(x_i, x_j\right) = \tanh(ax_i^T x_j + b) \tag{3}$$

$a > 0$ is the scaling parameter for the input data, and b is the shifting parameter that controls the threshold of mapping.

- *Naïve Bayes (NB):* based on Bayes theorem and conditional probability. It is a simple, useful and easy to build for large datasets. Bayes theorem is as follows:

$$P(C|D) = \frac{P\,(D\,|\,C) * P(C)}{P(D)} \tag{4}$$

where C and D are two events and $P\,(D) \neq 0$. $P(D)$ and $P(C)$ are the prior probabilities of observing D and C without regard to each other. $P\,(D\,|\,C)$ is the probability of observing event D given that C is true.

- *BayesNet:* based on various search algorithms and estimators. It's computationally expensive and has poor performance on high dimensional dataset.
- *Random Tree:* a collection of tree predictors based on tree and random forest idea. Input features vector classified with every tree and a final class is decided by majority voting.
- *AdaboostM:* creates a strong classifier from weak classifiers. Strong classifier handles the misclassified examples of weak classifiers. Often improves performance but sometime have overfitting problem.
- *Classification via Regression:* a regression model is created for each class where a class is binarized. A base classifier is used for classification.
- *Hoeffding Tree:* DT based incremental induction algorithm. It takes a little time to learn the data. It is based on Hoeffding bound to decide how many examples of an instance needs to achieve a certain level of confidence. This bound states that with probability $1 - \delta$, the true mean of the variable is at least $\overline{r}- \in$, and $\in$ is given by the following equation:

$$\in = \sqrt{\frac{R^2 ln1/\delta}{2n}} \tag{5}$$

where r is the real-valued random variable, with range R, n is the number of independent observations have been made and $\overline{r}$ is the mean value computed from independent observations.

- *REPTree:* Reduces Error Pruning Tree (REPTree) produces a fast decision tree using information gain and prone it using reduced-error pruning. At each iteration produce multiple trees and choose the fine one.
- *Fuzzylattice Reasoning:* a reasoning environment is created by Fuzzy Lattices. It can be used for classification using numeric predictors.
- *LogitBoost:* performs additive logistic regression. It perform classification using a regression scheme as the base learner and can also handle multi-class problems.
- *CForest*: it uses cost-sensitive voting for the classification problem. The aim is to make a prediction, which incurs the lowest classification cost. It calculates the expected total classification cost E of the whole dataset as below:

$$E = \frac{2 * C_P * C_N}{C_P + C_N} \tag{6}$$

where C_P and C_N are the costs of labeling set a of examples to positive and negative respectively. Both can be calculate as below:

$$C_P = N_{TP} * C_{TP} + N_{FP} * C_{FP} \tag{7}$$

$$C_P = N_{TN} * C_{TN} + N_{FN} * C_{FN} \tag{8}$$

N_{TP}, N_{FP}, N_{FN}, and N_{FP} are a number of true positive, true negative, false positive and false negative respectively. After E, CForest computes the ability of each attribute $A_i \in A$ to reduce the classification cost as follow:

$$E_{A_i} = 2 * \sum_{i=1}^{k} \frac{C_P^i * C_N^i}{C_P^i + C_N^i} \tag{9}$$

- *SPAARC:* implements a decision tree using split-point sampling and node attribute sampling. It uses an altered version of SimpleCart with equal accuracy to it but with 89% greater speed.

B. DEEP LEARNING CLASSIFICATION

DL models are based on an ML model called Artificial Neural Network (ANN). A DL model consists of a set of layers: an input, an output, and one or more hidden layers. The input layer receives input in a matrix form. Hidden layers perform feature selection and dimensionality reduction. Output layer

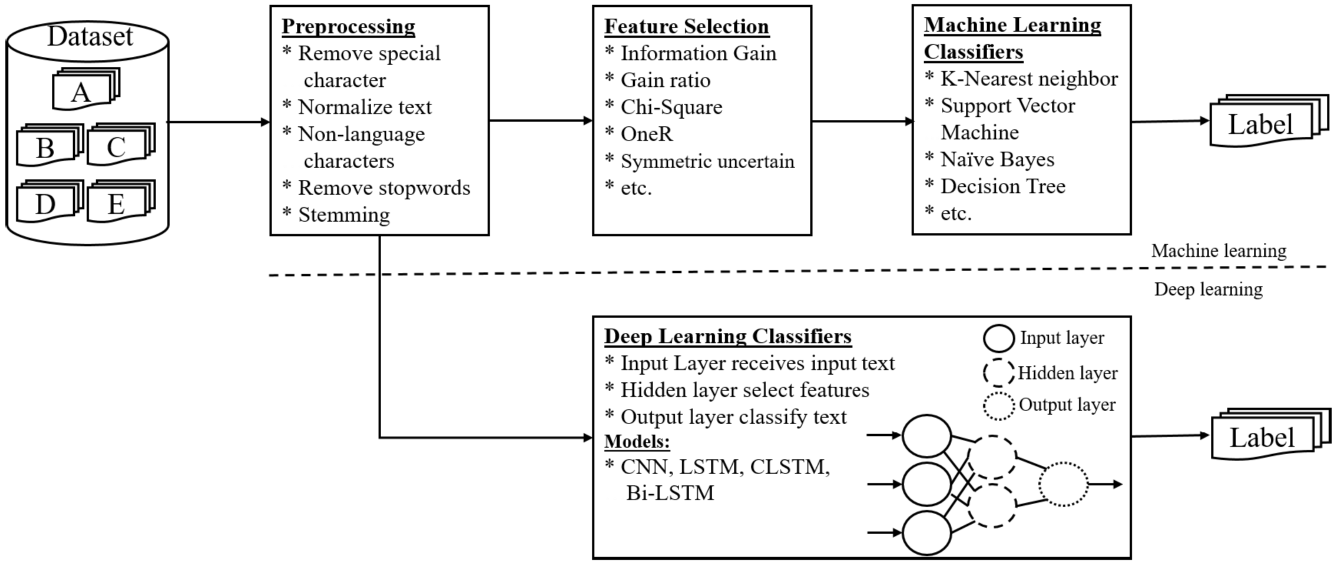


FIGURE 4. The workflow of traditional machine learning and deep learning.

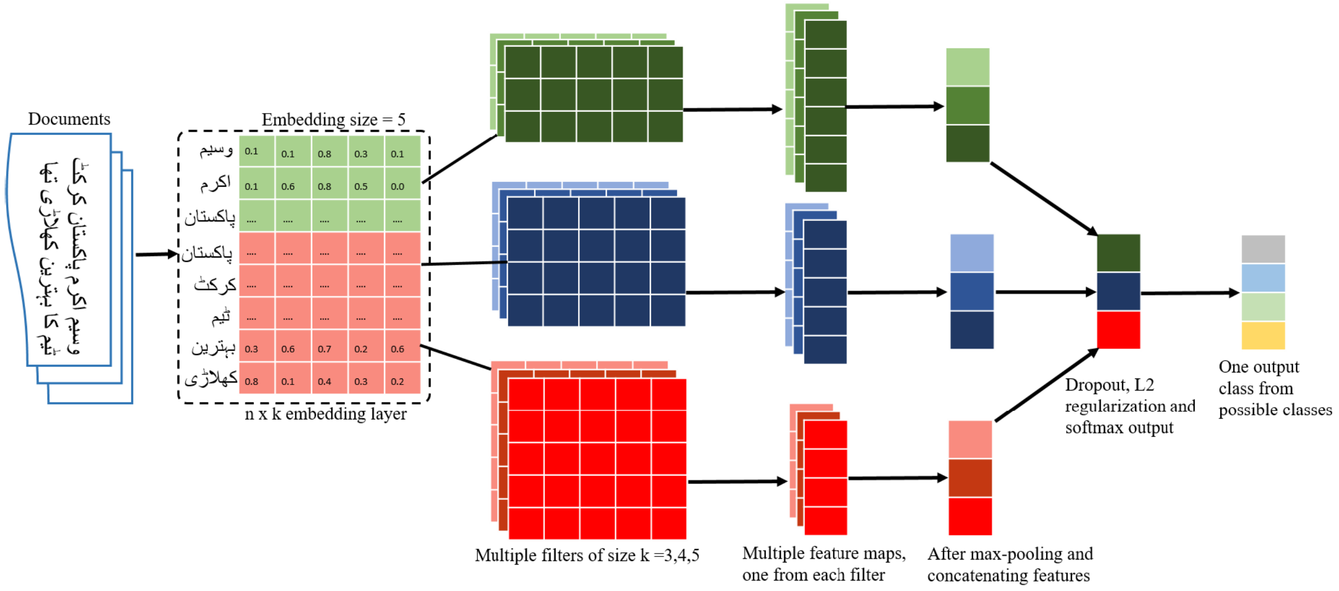

Documents
وسیم اکرم پاکستان کرکٹ ٹیم کا بہترین کھلاڑی ہے
Embedding size = 5
وسیم
اکرم
پاکستان
پاکستان
کرکٹ
ٹیم
بہترین
کھلاڑی
0.1 0.1 0.8 0.3 0.1
0.1 0.6 0.8 0.5 0.0
0.3 0.6 0.7 0.2 0.6
0.8 0.1 0.4 0.3 0.2
n x k embedding layer
Multiple filters of size k =3,4,5
Multiple feature maps, one from each filter
After max-pooling and concatenating features
Dropout, L2 regularization and softmax output
One output class from possible classes

FIGURE 5. Single-layer Multisize Filters Convolutional Neural Network.

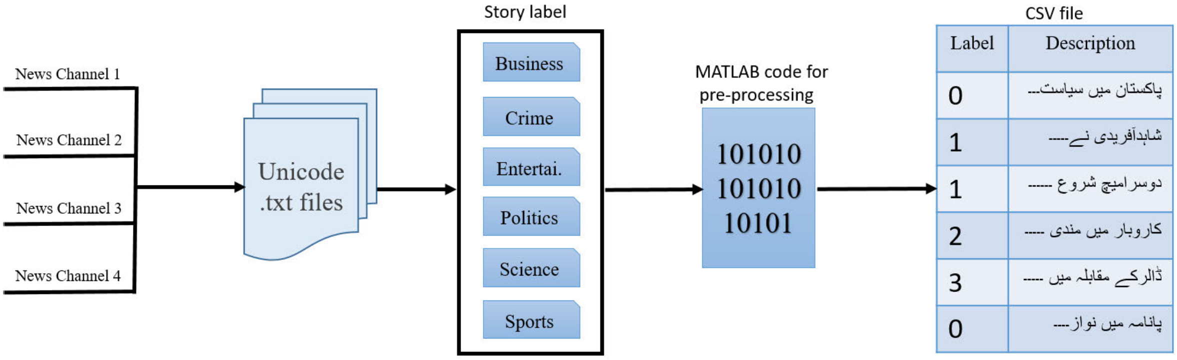
classifies the input document into one of the predefined labels. DL models automatically learn the features from the dataset without explicitly programmed methods as shown in Fig. 4. There are two main models of DL: Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). CNN showed good performance in NLP tasks [40]. CNN architecture also consists of input, hidden, output layers. Hidden layers perform the convolution operation, feature space reduction and act like multilayer perceptron classifier [41]. A CNN has a few parameters, known as hyperparameters, like a number of filters, filter size, dropout, etc. CNN performance can be increased after finding the best values of these parameters [42]. In this study, we use a CNN model with multiple filters of various sizes to extract variable-length features from the text.

C. MACHINE LEARNING VS. DEEP LEARNING TEXT CLASSIFICATION

The TC process of both traditional ML and DL methods is shown in Fig. 4. First, the dataset is preprocessed to make it suitable for input to a classifier. After preprocessing, in ML, feature selection methods have been applied to choose valuable features from a large feature space. A classifier learned the selected features to classify text documents.

TABLE 3. Differences between ML and DL methods.

Machine Learning	Deep Learning
Perform well on small data	Perform well on large data
Depends on feature selection methods	The feature extraction process is hidden and automatic
Slow because of low-end machine	Use GPU to accelerate performance
Learn directed features only	Learn a range of features from low-level to high-level
Less training time but more testing time	More training time but less testing time
Interpretable results	Non-interpretable results
Less expensive	More expensive



Label	Description
0	پاکستان میں سیاست---
1	شاہدآفریدی نے-----
1	دوسرامیچ شروع -----
2	کاروبار میں مندی -----
3	ڈالرکے مقابلہ میں -----
0	پانامہ میں نواز----

FIGURE 6. Process of converting a plain text corpus into CSV format.

After preprocessing, in the DL model, features are extracted automatically by convolutional and max-pool layers.

In contrast to ML, a DL model can learn complex patterns of the dataset but these models require more data for training and also require special GPU hardware for matrix calculation quickly. Some difference between ML and DL are given in Table 3.

IV. PROPOSED MODEL AND DATASET

In this study, we use a Single-layer Multisize Filters Convolutional Neural Network (SMFCNN) that is shown in Fig. 5. SMFCNN is different from [3] model where multiple input channels were used. We use a single input channel without predefined word embedding. Multiple filters of various sizes are used to obtain multiple feature maps of each filter size. 1-max pooling is applied on each feature map. Then all the feature maps are concatenated to form a feature vector for the penultimate layer. Softmax layer is used to finally classify the document into one of the multiple classes. The hyperparameters of SMFCNN like a number of filters, filter size, dropout, batch size, etc. were fine-tuned as [7] on three datasets. Finding out the optimum values for these parameters help to minimize the loss and improve the performance of the model. Studies of [7], [35], [43], [44] conclude and suggest to fine-tune these parameters and it is described in section V.

A. PROPOSED DATASET AND ITS COMPARISON WITH OTHER DATASETS

Deep neural network (DNN) models, like CNN, are usually considered robust when dealing with high dimensional feature space. The learning of the network depends on the dataset and if the dataset are not enough then the learning would be inadequate as well as the feature selection would not be desirable because DNN is like a black box that makes it difficult to observe which features are chosen for the task [45]. To provide adequate dataset to train the model and to evaluate the performance of our model, we use three datasets of small, medium and large sizes (see Table 4 for category-wise detail of these datasets and Table 5 for their comparison) to analyze the performance comparatively. We clean these datasets by removing punctuations marks, special symbols, and other non-Urdu characters or words so that the dataset only contains the words of Urdu language. We use MATLAB to clean the dataset and saved it in the CSV file. In the CSV file, label of a document is represented as integer value and description in the Unicode text.

1) NORTHWESTERN POLYTECHNICAL UNIVERSITY URDU (NPUU) DATASET

It is a self-collected dataset of news articles of Urdu text. The process of dataset design and annotation is same as adapted in

TABLE 4. Category wise statistics for the NPUU, naïve, and COUNTER dataset.

	Category	| Docs |	Percent	Doc. Len$_{min}$	Doc. Len$_{max}$	Doc.Len$_{avg}$	|Words |
NPUU	Business	3,328	31%	08	3,059	338	1,125,700
	Crime	847	8%	33	1,990	248	210,021
	Entertainment	1,913	18%	22	3,254	287	549,117
	Politics	1,636	15%	18	3,166	350	573,786
	Science & Tech.	1,655	15%	19	3,237	470	779,331
	Sports	1,440	13%	11	2,971	256	373,801
	Total	10,819				325	3,611,756
naïve	Economy	614	12%	62	1,408	391	240,133
	Entertainment	1,482	30%	47	2,728	495	734,701
	Politics	1,271	25%	47	4,129	501	636,844
	Sports	1,636	33%	47	3,853	369	605,167
	Total	5,003				439	2,216,845
COUNTER	Business	54	05%	51	451	177	9,567
	Foreign	242	20%	43	717	152	36,904
	National	362	30%	49	2,480	340	123,171
	Showbiz	98	08%	62	529	182	17,820
	Sports	444	37%	54	1,019	228	101,373
	Total	1,200				215	288,835

TABLE 5. Summary of the three datasets used for classification.

Properties	COUNTER	naïve	NPUU
Size	Small (2.72 MB)	Medium (17.8 MB)	Large (28.9 MB)
No. of doc.	1,200	5,003	10,819
Max. length doc.	2,480	4,129	3,254
Min. length doc.	43	47	08
Avg. length doc.	215	439	325
No. of classes	5	4	6
No. of words	288,835	2,216,845	3,611,756
Imbalanced level	High	Low	High
Split-ratio	5	5	5

past studies of different languages [13], [14], [25], [31], [46]. The news articles are collected from well-known Urdu news websites like Express, ARY, and Geo. We collected 10819 news articles that belong to six categories (see Table 4). Four graduate students who are the native speakers of the Urdu language manually annotated the dataset. The news articles are saved into notepad file as Unicode encoding with the .txt file format. Each news article contains its headline and description of the news. The process of collecting and preprocessing our dataset is shown in Fig. 6. The dataset is publically available at github.[3]

2) NAÏVE DATASET

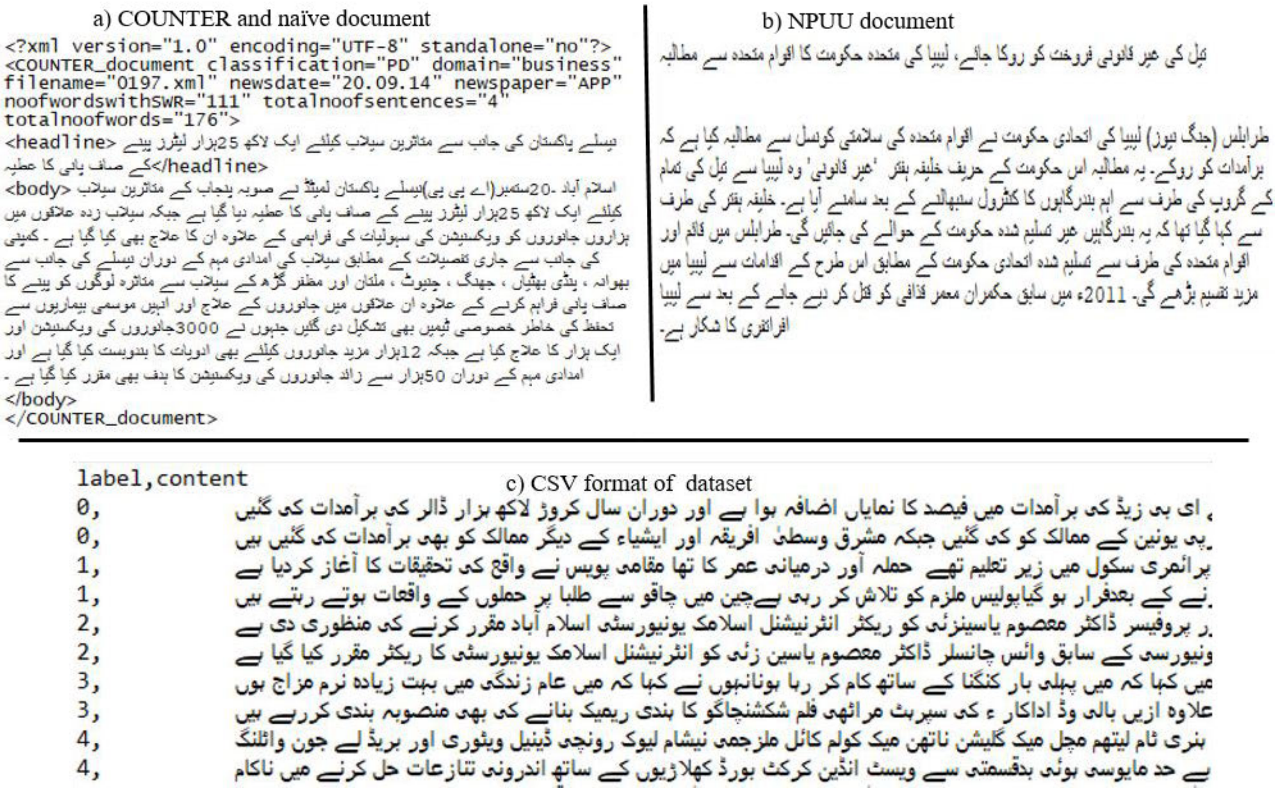
The naïve dataset contained 5003 documents of Urdu organized into four categories (see Table 4). News stories were collected from news websites like the British Broadcasting Company and Voice of America [12]. The dataset is in XML format and after cleaning it, we converted it into CSV format using MATLAB.

3) COUNTER DATASET

It is a small dataset designed for the study of Urdu text reuse [32]. It contains 1200 text documents of news articles distributed into five categories (see Table 4). This dataset is also in XML file format and is preprocessed and then converted into CSV file format using MATLAB code.

Table 4 shows the category wise detail of the NPUU dataset. The dataset is imbalanced dataset where the business category has the most number of documents while crime has the least number of documents. The format of the text documents of three datasets and their CSV file format is shown in Fig. 7. COUNTER and naïve document samples

[3] https://github.com/pervezbcs/NPUU



a) COUNTER and naïve document

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<COUNTER_document classification="PD" domain="business"
filename="0197.xml" newsdate="20.09.14" newspaper="APP"
noofwordswithSWR="111" totalnoofsentences="4"
totalnoofwords="176">
<headline> نیسلے پاکستان کی جانب سے متاثرین سیلاب کیلئے ایک لاکھ 25ہزار لیٹرز پینے
کے صاف پانی کا عطیہ</headline>
<body> اسلام آباد -20ستمبر(اے پی پی)نیسلے پاکستان لمیٹڈ نے صوبہ پنجاب کے متاثرین سیلاب
کیلئے ایک لاکھ 25ہزار لیٹرز پینے کے صاف پانی کا عطیہ دیا گیا ہے جبکہ سیلاب زدہ علاقوں میں
ہزاروں جانوروں کو ویکسینیشن کی سہولیات کی فراہمی کے علاوہ ان کا علاج بھی کیا گیا ہے ۔ کمپنی
کی جانب سے جاری تفصیلات کے مطابق سیلاب کی امدادی مہم کے دوران نیسلے کی جانب سے
بھوانہ ، پنڈی بھٹیاں ، جھنگ ، چنیوٹ ، ملتان اور مظفر گڑھ کے سیلاب سے متاثرہ لوگوں کو پینے کا
صاف پانی فراہم کرنے کے علاوہ ان علاقوں میں جانوروں کے علاج اور انہیں موسمی بیماریوں سے
تحفظ کی خاطر خصوصی ٹیمیں بھی تشکیل دی گئیں جنہوں نے 3000جانوروں کی ویکسینیشن اور
ایک ہزار کا علاج کیا ہے جبکہ 12ہزار مزید جانوروں کیلئے بھی ادویات کا بندوبست کیا گیا ہے اور
امدادی مہم کے دوران 50ہزار سے زائد جانوروں کی ویکسینیشن کا ہدف بھی مقرر کیا گیا ہے ۔
</body>
</COUNTER_document>
```

b) NPUU document

تیل کی غیر قانونی فروخت کو روکا جائے، لیبیا کی متحدہ حکومت کا اقوام متحدہ سے مطالبہ

طرابلس (جنگ نیوز) لیبیا کی اتحادی حکومت نے اقوام متحدہ کی سلامتی کونسل سے مطالبہ کیا ہے کہ
برآمدات کو روکے۔ یہ مطالبہ اس حکومت کے حریف خلیفہ ہفتر 'غیر قانونی' وہ لیبیا سے تیل کی تمام
کے گروپ کی طرف سے اہم بندرگاہوں کا کنٹرول سنبھالنے کے بعد سامنے آیا ہے۔ خلیفہ ہفتر کی طرف
سے کہا گیا تھا کہ یہ بندرگاہیں غیر تسلیم شدہ حکومت کے حوالے کی جائیں گی۔ طرابلس میں قائم اور
اقوام متحدہ کی طرف سے تسلیم شدہ اتحادی حکومت کے مطابق اس طرح کے اقدامات سے لیبیا میں
مزید تقسیم بڑھے گی۔ 2011ء میں سابق حکمران معمر قذافی کو قتل کر دیے جانے کے بعد سے لیبیا
افراتفری کا شکار ہے۔

c) CSV format of dataset

```
label,content
0,      ای بی زیڈ کی برآمدات میں فیصد کا نمایاں اضافہ ہوا ہے اور دوران سال کروڑ لاکھ ہزار ڈالر کی برآمدات کی گئیں
0,      پی یونین کے ممالک کو کی گئیں جبکہ مشرق وسطیٰ افریقہ اور ایشیاء کے دیگر ممالک کو بھی برآمدات کی گئیں ہیں
1,      پرائمری سکول میں زیر تعلیم تھے حملہ آور درمیانی عمر کا تھا مقامی پولیس نے واقع کی تحقیقات کا آغاز کردیا ہے
1,      نے کے بعدفرار ہو گیاپولیس ملزم کو تلاش کر رہی ہےچین میں چاقو سے طلبا پر حملوں کے واقعات ہوتے رہتے ہیں
2,      ر پروفیسر ڈاکٹر معصوم یاسینزئی کو ریکٹر انٹرنیشنل اسلامک یونیورسٹی اسلام آباد مقرر کرنے کی منظوری دی ہے
2,      ونیورسٹی کے سابق وائس چانسلر ڈاکٹر معصوم یاسین زئی کو انٹرنیشنل اسلامک یونیورسٹی کا ریکٹر مقرر کیا گیا ہے
3,      میں کہا کہ میں پہلی بار کنگنا کے ساتھ کام کر رہا ہونانہوں نے کہا کہ میں عام زندگی میں بہت زیادہ نرم مزاج ہوں
3,      علاوہ ازیں بالی وڈ اداکار ء کی سپرہٹ مراٹھی فلم شکشنچاگو کا ہندی ریمیک بنانے کی بھی منصوبہ بندی کررہے ہیں
4,      ہنری ثام لیتھم مچل میک گلیشن ناتھن میک کولم کائل ملزجمی نیشام لیوک رونچی ڈینیل ویٹوری اور بریڈ لے جون واٹلنگ
4,      بے حد مایوسی ہوئی بدقسمتی سے ویسٹ انڈین کرکٹ بورڈ کھلاڑیوں کے ساتھ اندرونی تنازعات حل کرنے میں ناکام
```

FIGURE 7. Samples of Urdu text document and CSV file format with labels of each document.

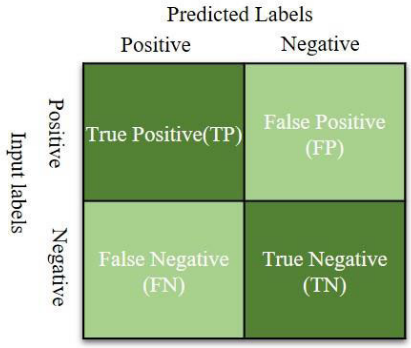


FIGURE 8. Confusion matrix for two classes.

are shown in Fig. 7 (a) that is in XML format. The document sample of NPUU is shown in Fig. 7 (b) that is in plain text format. CSV file format of the dataset is shown in Fig. 7 (c) that is a common format for all datasets. A statistical comparison of three datasets is given in Table 5. It can be seen that NPUU is a larger and a complex dataset with a maximum number of documents, classes, and words.

V. MODEL EVALUATION AND PARAMETER SETTINGS

A. PERFORMANCE MEASURES

To measure the classification performance of SMFCNN and baseline classifiers, we use most common performance measures: precision, recall, F-measure and accuracy [4], [47]. A confusion matrix is used to visualize and evaluate the performance of a classifier as shown in Fig. 8. True positive (TP) is the number of documents correctly predicted as the positive class. True negative (TN) is the number of documents correctly classified as negative class. False Positive (FP) is the number of documents predicted wrongly as the positive class when it was actually not. False Negative (FN) is the number of documents predicted wrongly as the negative class when it was actually not. Precision, recall, F-measure, and accuracy can be calculated using confusion matrix.

- *Precision:* used to measure the exactness of the classifier result and can be calculated as given below:

$$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{10}$$

- *Recall:* recall measures the completeness of the classifier results. It is calculated by the equation below:

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{11}$$

- *F-measure:* is the harmonic mean of precision and recall and can be calculated as:

$$F - measure = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

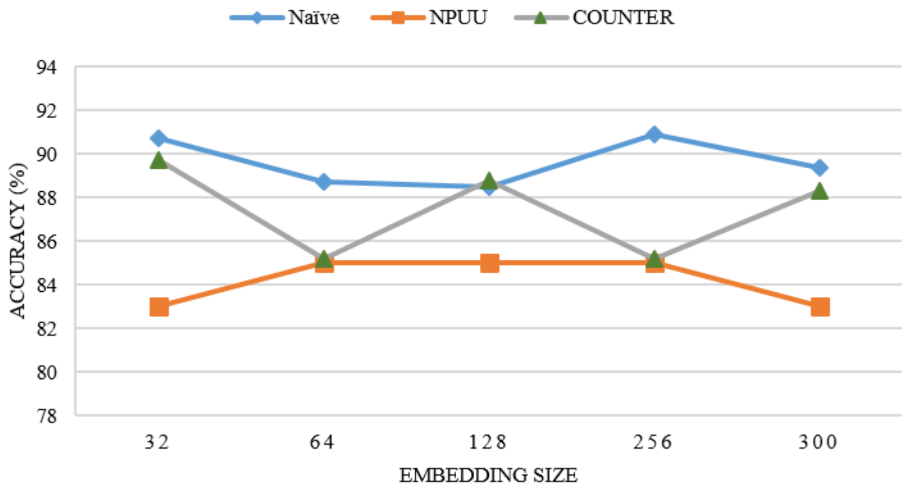


FIGURE 9. Effect of word embedding size on network performance.

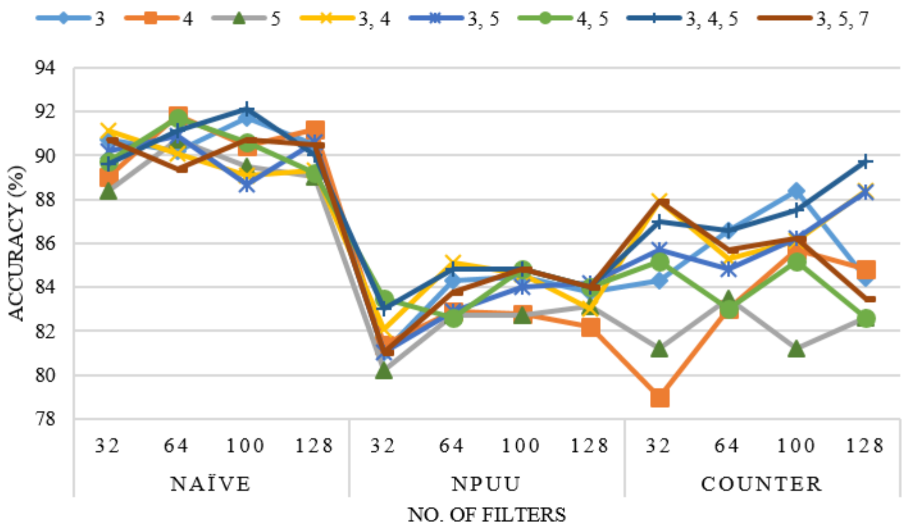


FIGURE 10. Comparison of size of the convolutional filters with a number of filters.

- *Accuracy:* most common measure for classifier performance and can be calculated as given below:

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{13}$$

B. HYPERPARAMETER SETTINGS

To optimize the network parameters, we initialized our model parameters as [3] model's parameters and the detail is given in Table 6. For all three datasets, the initial parameters are the same. All the experiments are performed on Intel Core i7-7700 3.60 GHz processor, 16 GB of RAM, NVIDIA GeForce GTX 1080 graphics card, Windows 10, and Tensorflow-GPU 1.9.0 with CUDA toolkit 9.0. To make all the documents of fixed length and create batches of fixed size, small size documents are padded with 'UNK' word to a maximum length of the document in the dataset. A vocabulary of all unique words (including 'UNK') is designed. A vocabulary index is generated by mapping each word of a document to an integer index of vocabulary. In this way, each document is represented as a vector of integers of fixed length.

TABLE 6. Initial parameters used to optimize the SMFCNN model.

Parameter	Value	Parameter	Value
Batch Size	50	Filter Size	3,4,5
Embedding Size	300	No. of Filters	100
No. of Epoch	50	Dropout	0.5
Activation Function	ReLu	Training-Testing	80-20

Word embedding is a popular way to represent the vocabulary of a text document. Words of a document are represented as a real-valued vector and the size of this vector can also affect the performance of the model. Predefined word embedding of fixed size is not used and the embedding size (E)

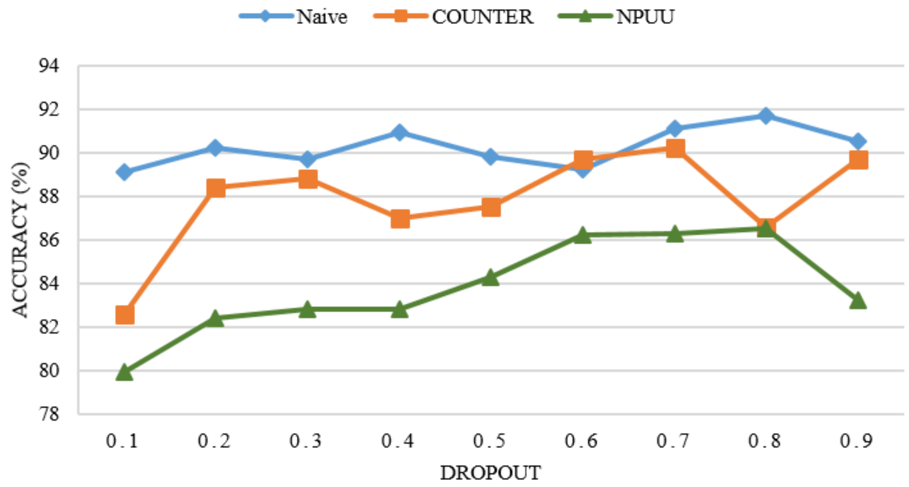


FIGURE 11. Performance of SMFCNN on different values of dropout.

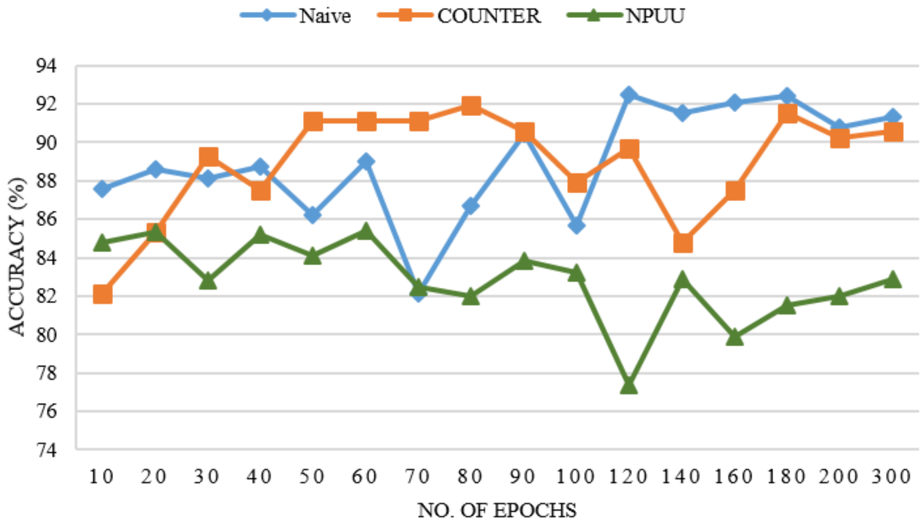


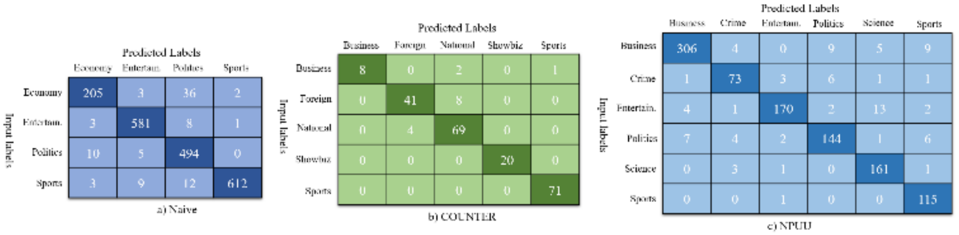
FIGURE 12. Effect of number of epochs on the performance of the model.

is chosen as a hyperparameter. Results in Fig. 9 shows that the value of E depends on the size of the dataset as well as length of the documents. Embedding size 32, 256, and 128 achieved maximum accuracies on small, medium and large size datasets.

Convolution operation uses one or more filters of different or equal size to calculate feature maps. The number of filters and the size of the filter can also affect the performance of the network and depend on the complexity of the problem and dataset. A large filter slows down the training while a small filter decrease performance by missing discriminative information [17]. Fig. 10 shows the comparison of filter size with the number of filters. Variable size filters perform well than a single size filter on three datasets. Filters of size [3, 4, 5] perform the best for small and medium size dataset while [3, 4] achieved high accuracy on a large dataset.

The performance of our model on different dropout values is shown in Fig. 11. Dropout helps to prevent the network from noise and overfitting. If the model is trained on insufficient dataset then the model may face the problem of overfitting. The solution is to increase the dataset size or reduce the number of hidden units used for computing features. Dropout deletes or inactivates the inactive units in the hidden layer of the model. These units do not participate in the calculation on next iterations. Fig. 11 shows that the model achieves the highest accuracy on 0.7 dropout for small dataset while on 0.8 dropout for medium and large datasets.

When a model passed through all the examples once in a forward pass and a backward pass in a dataset or in a batch then it is called one epoch. CNN require more epochs to learn a large, complex and noisy dataset. A number of epochs should be chosen carefully because a large number of epochs may cause overfitting the network that will show high



Input labels \ Predicted Labels	Economy	Entertain.	Politics	Sports
Economy	205	3	36	2
Entertain.	3	581	8	1
Politics	10	5	494	0
Sports	3	9	12	612

a) Naive

Input labels \ Predicted Labels	Business	Foreign	National	Showbiz	Sports
Business	8	0	2	0	1
Foreign	0	41	8	0	0
National	0	4	69	0	0
Showbiz	0	0	0	20	0
Sports	0	0	0	0	71

b) COUNTER

Input labels \ Predicted Labels	Business	Crime	Entertain.	Politics	Science	Sports
Business	306	4	0	9	5	9
Crime	1	73	3	6	1	1
Entertain.	4	1	170	2	13	2
Politics	7	4	2	144	1	6
Science	0	3	1	0	161	1
Sports	0	0	1	0	0	115

c) NPUU

FIGURE 13. Confusion matrices for the three datasets on which the maximum performance was achieved.

TABLE 7. Optimized parameters for SMFCNN.

Parameters	naïve	NPUU	COUNTER
Embedding Size	256	128	32
Filter Size	3, 4, 5	3,4	3,4,5
No. of Filters	100	64	128
No. of Epochs	120	60	70
Activation Function	Sigmoid	ReLU	Tanh
Dropout	0.8	0.8	0.7

accuracy on training dataset while low accuracy on testing dataset. Fig. 12 shows that our model achieves the maximum accuracies on 60, 80 and 120 epochs for large, small and medium size datasets.

A convolutional layer contains a set of neurons that calculate the output in two steps: calculate a weighted sum and output using a non-linear activation function on the sum. The activation function is a non-linear function that helps (1) to introduce non-linearity into the neural function, (2) to make sure the output is differentiable to support back propagation, and (3) to prevent saturation of values. We tested Rectified Linear Unit (ReLu), Sigmoid or logistic and Hyperbolic Tangent (Tanh) activation functions. ReLU on NPUU, Sigmoid on naïve and Tanh on COUNTER datasets achieve maximum accuracies.

Batch size is the number of examples given at a time to the network for processing in a single iteration. Large batch consumed more memory of GPU and training processes would be slow. We tested our model on four batches of different size (32, 50, 64, and 128 documents). Results show\ that a batch size of 32 performs better and it endorsed the findings of [48]. We use a mini-batch size of 32 for all the experiments because the large batch size requires more memory, time and also minimize the performance while the small batch size increases the performance [48].

Pooling layer shrinks or reduces the size of the feature map by merging or selecting a high valued feature from the neighboring features so that the high-level layers can deal with more abstract or global values. We used 1-max-pooling. Gradient Descent (GD) is used for network training optimization and Adagrad algorithm used to optimize error and to adapt the learning rate based on the parameters. Softmax layer used as a probabilistic function for obtaining a confidence score for the classification decision. After performing hundreds of experiments on three datasets, Table 7 shows the optimized parameter values.

VI. RESULTS AND DISCUSSION

After hyperparameter tuning of our model, we evaluated the performance of SMFCNN using four different ways: 1) without removing both stopwords and rare words, 2) after removing stopwords 3) after removing both stopwords and rare words and 4) with different split-ratios of the dataset into training and testing subsets. To remove stopwords, a list of Urdu stopwords is used from GitHub.[4] For COUNTER, we use 80% documents for training and 20% for testing. For naïve, we use 60% documents for training and 40% for testing the model. For NPUU, 90% of documents are used for training and 10% for testing because of its large size and complexity. 10-fold cross-validation is used for validation. The confusion matric for three datasets on which the model achievs the maximum performance is shown in Fig. 13. Precision, recall, F-measure, and accuracy is given in Table 8. From the results, it can be seen that only removing stopwords from the text has minor effects for NPUU and naive datasets while greater impact for COUNTER. Maximum values of precision, recall, F-measure and accuracy are obtained after removing both stopwords and rare words on all the datasets.

It can be seen that only stopwords removal from small dataset COUNTER cause to increase SMFCNN performance that endorse the findings of [31]. For medium and large datasets, removing stopwords from text slightly decreases the SMCNN performance. Stopwords and rare words together have a great impact on the performance of a large dataset as [2] concluded on Arabic but its contradict to [19] on English. As Table 8 shows that after removing both stopwords and rare words, for a large dataset NPUU, accuracy is increased from 89.7% to 91.8%. For naïve dataset, accuracy increased from 94.9% to 95.4% only after removing both stopwords and rare words. For small size dataset COUNTER, accuracy is increased from 89.6% to 93.3%. It can be concluded from the results that removing both stopwords and rare words are very important to achieve maximum performance of SMFCNN model for Urdu text documents classification.

We used five different dataset split-ratios for each dataset and compare SMFCNN performance as shown in Fig. 14.

[4] https://github.com/urduhack/urdu-stopwords

TABLE 8. SMFCNN performance on three datasets.

Dataset	Text	P	R	F	A
COUNTER	Without stopwords and rare words removal	92.1	77.5	84.2	89.6
	Stopwords removal	93.0	81.7	86.7	90.6
	Stopwords + rare words removal	95.4	90.2	92.7	93.3
naïve	Without stopwords and rare words removal	94.5	92.6	93.5	94.9
	Stopwords removal	93.8	93.0	93.4	94.7
	Stopwords + rare words removal	94.8	93.6	94.2	95.4
NPUU	Without stopwords and rare words removal	89.9	88.0	88.9	89.7
	Stopwords removal	89.8	87.5	88.6	89.7
	Stopwords + rare words removal	90.4	91.7	91.0	91.8

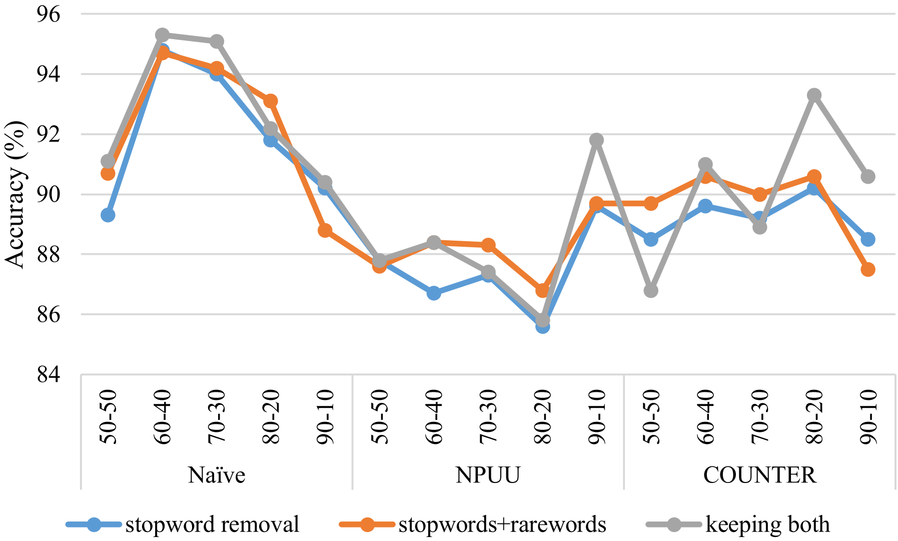


FIGURE 14. Performance of SMFCNN on three datasets with preprocessing operations and using five split-ratios.

Comparative analysis showed that choosing an appropriate split-ratio for a dataset is crucial for achieving maximum performance of a classifier. For example, the 80-20 dataset split-ratio of NPUU has accuracy of 85.8% that is the minimum accuracy across all five split-ratios. 90-10 split-ratio achieves 91.8% accuracy that is maximum accuracy among all split-ratios and is 6% higher than the accuracy of 80-20 split-ratio. Similarly, naïve dataset 60-40 split-ratio performs the best than others while the 90-10 split-ratio achieves the worst accuracy. By choosing appropriate split-ratio, in the naïve dataset is 60-40, increase the maximum performance up to 4.9%. Which dataset split-ratio should be used for a dataset? It depends on the size and imbalance level of a dataset. Our experiments show that if the dataset is more imbalanced and have a large number of features then SMFCNN requires more documents in training than testing as in the case of NPUU dataset.

We analyzed and evaluated the performance of SMFCNN after removing rare words from the datasets. Our experiments shows that rare words reduce the vocabulary size as well as increase the performance of the model. From Fig. 15 and Fig. 16, it can be noticed that SMFCNN achieves the best performance after removing the words that occurred twice by reducing 26% and 25% vocabulary size of naïve and COUNTER datasets. SMFCNN achieves high accuracy values after removing those words, which occurred three times by reducing 42% vocabulary size on large size NPUU dataset. Furthermore, SMFCNN performance on the small and more imbalanced dataset, like COUNTER, is not much stable as compared to large and balanced datasets.

We used three imbalanced datasets with different imbalanced level as shown in Table 9. COUNTER dataset is more imbalanced dataset than naïve and NPUU. To achieve high performance on imbalanced datasets is a challenging task for ML and DL models. However, the SMFCNN has the ability to handle imbalance dataset of different imbalance level and various sizes. Accuracy is not a good performance measure for a classifier in the case of the imbalanced dataset [42]. The difference between F-measure and accuracy is less for naïve dataset as compare to others because the imbalanced level (difference between the major and the minor class) of naïve is less than other two datasets.

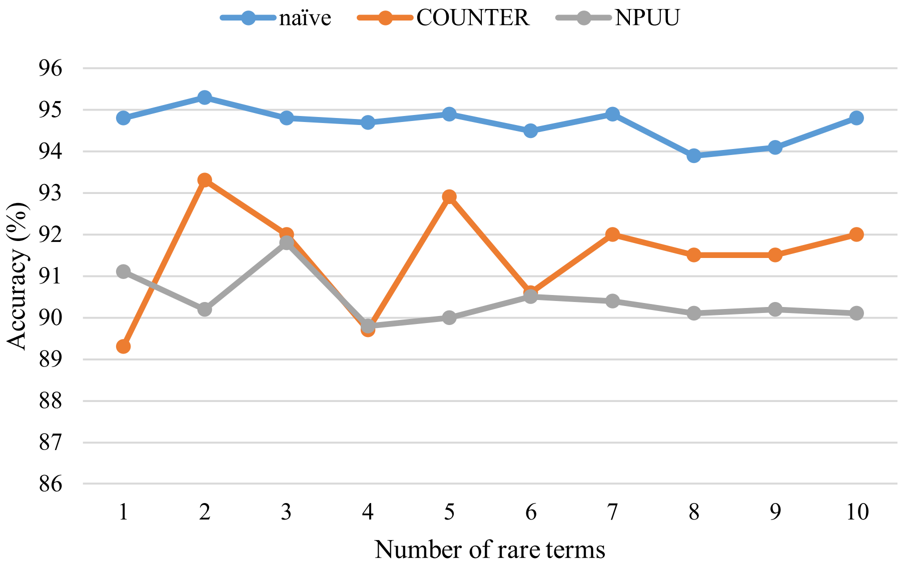


FIGURE 15. Effects of rare words on the performance of SMFCNN on three datasets.

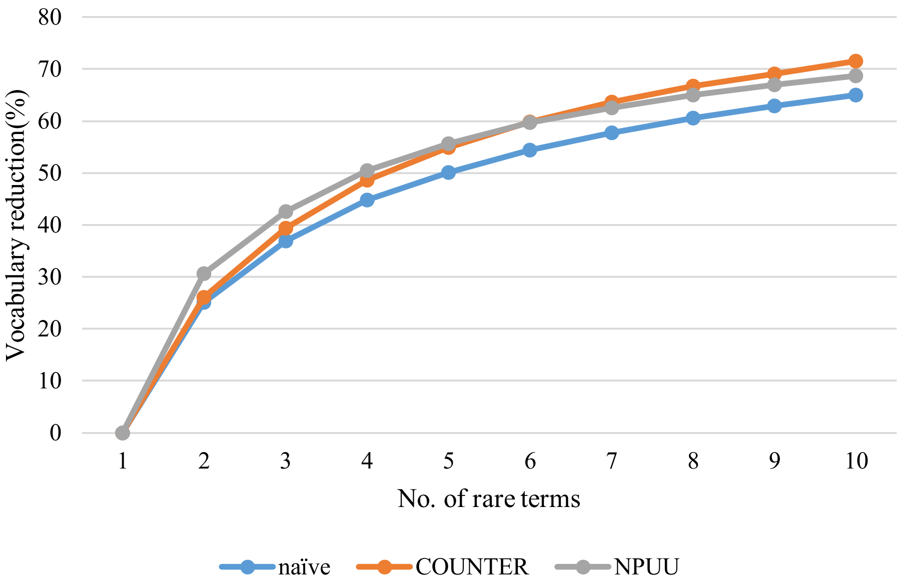


FIGURE 16. Effects of rare words on the vocabulary of three datasets.

TABLE 9. Effects of the balanced and imbalanced dataset on SMFCNN performance.

Dataset	Category max. no. of docs = %	Category min. no. of docs = %	Diff. b/w categories no. of docs = %	F-measure	Accuracy
COUNTER	Sports (444 = 37%)	Business (54 =4.5%)	390 = 32.5%	92.7%	93.3%
NPUU	Business (3328 = 30.7%)	Crime (847 = 7.82%)	2481 =22.9%	91.0%	91.8%
naïve	Sports (1636 =32.07%)	Economy (614=12.27%)	1022=20.42%	**94.2%**	**95.4%**

For all three datasets, a brief summary of SMFCNN performance is given in Table 10 where the values within the parenthesis show significance difference in the performance. SMFCNN performs well on low imbalanced datasets. Stopwords removal slightly affects the SMFCNN performance on three datasets. Removal of both stopwords and rare words

TABLE 10. Summary of the performance comparison of SMFCNN on three datasets.

Dataset Properties	COUNTER	naïve	NPUU
Size	Small	Medium	Large
Imbalanced level	High	Low	Low
Effects of imbalance level	High	Low	High
Effects of stopwords removal	High	Little	Little
Effects of Stopwords+rare words removal	Yes	Yes	Yes
SMFCNN performance	89.6	94.9	89.7
SMFCNN (stopwords removed)	90.6 (+1%)	94.7 (-0.2%)	89.7 (+0%)
SMFCNN (stopwords+rare word removed)	93.3 (+3.7%)	95.4 (+0.5%)	91.8 (+2.1%)
Best split-ratio	80-20	60-40	90-10

TABLE 11. Precision, recall, f-measure and accuracy values of SMFCNN and ML classifiers on three datasets.

Classifier	COUNTER				naïve				NPUU			
	P	R	F	A	P	R	F	A	P	R	F	A
Naïve Bayes	91.7	91.4	91.5	91.4	93.2	92.7	92.8	92.7	89.0	88.4	88.6	88.4
BayesNet	***92.5***	***91.0***	***91.2***	***91.0***	***95.3***	***95.1***	***95.1***	***95.1***	***90.5***	***90.1***	***90.2***	***90.1***
IBK	86.8	86.2	85.6	86.2	80.8	77.1	75.9	77.1	92.9	25.1	39.6	58.3
J48	89.9	89.5	89.6	89.5	90.4	90.4	90.4	90.4	84.6	84.5	84.5	84.5
Random Tree	78.7	78.3	78.5	78.3	70.4	70.3	70.3	70.3	63.7	63.8	63.7	63.8
REPTree	87.8	87.3	87.4	87.3	89.0	88.9	88.9	88.9	81.1	80.9	80.9	80.9
Hoeffding Tree	90.7	79.4	82.7	79.4	92.4	88.9	89.8	88.9	86.9	79.0	80.8	78.9
CSForest	75.9	72.8	71.9	72.7	79.3	78.8	78.5	78.9	59.6	57.1	55.6	42.9
SVM Polynomial	79.9	71.8	64.8	71.7	86.5	84.7	80.6	84.7	78.0	53.8	50.6	53.8
SVM Radial basis	91.5	90.8	89.8	90.8	91.0	90.3	89.4	90.3	82.5	74.1	73.3	74.0
SVM Sigmoid	76.8	75.6	75.6	75.6	75.4	73.5	74.2	73.5	69.4	69.1	69.1	69.1
AdaboostM1	86.8	86.2	85.6	86.2	80.7	77.1	75.9	77.0	72.7	58.3	56.2	58.3
Logit Boost	91.2	90.4	90.5	90.4	92.9	92.6	92.6	92.6	84.5	84.1	84.1	84.1
Classification via Regression	91.2	90.3	90.3	90.3	91.6	91.5	91.5	91.5	85.3	85.2	85.2	85.2
FLR	92.1	90.1	89.9	90.1	90.4	89.2	89.1	89.2	84.0	81.9	81.8	81.9
SPAARC	90.1	89.8	89.9	89.8	89.7	89.7	89.7	89.7	83.1	83.0	83.0	83.0
SMFCNN	**95.4**	**90.2**	**92.7**	**93.3**	**94.8**	**93.6**	**94.2**	**95.4**	**90.4**	**91.7**	**91.0**	**91.8**

together from both small and medium dataset also slightly affect the SMFCNN performance. However, it is not true for large dataset NPUU.

A. COMPARISON OF SMFCNN WITH ML CLASSIFIERS

We also compared the performance of SMFCNN with sixteen well-known and mostly used ML classifiers. DL models have been compared with ML models in many studies of text classification [5], [7], [17], [49] but no comparison have been made in literature for Urdu text classification because DL models have not been used for text classification of Urdu. After preprocessing the datasets, we used IG to measure the goodness of features as used by [11] and concluded by [12] as a best feature selection method for Urdu text classification. IG can be calculated as given below:

$$IG(s_i, c_j) = H(s_i) - H(s_i|c_j) \tag{14}$$

where s_i and c_j are the j-th term (word) and the class label respectively, $H(s_i)$ is the entropy of term s_i, and $H(s_i|c_j)$ is the entropy of s_i after observing class label c_j. $H\ (s_i)$ and $H(s_i|c_j)$ can be defined as:

$$H\ (s_i) = -\sum_j p(x_j)\ log_2\ p(x_j) \tag{15}$$

$$H(s_i|c_j) = -\sum_k p(c_k) \sum_j p(x_j|c_k)\ log_2\ p(x_j|c_k) \tag{16}$$

We used WEKA (Waikato Environment for Knowledge Analysis), an open-source toolkit that provides state of the art ML algorithms for text processing [39]. Results of experiments are shown in Table 11. It can be seen that SMFCNN model outperforms the other ML models on three datasets. SMFCNN achieves 93.3%, 95.4% and 91.8% accuracy on COUNTER, naïve, and NPUU datasets respectively. Similarly, SMFCNN obtains F-measure values 92.7%, 94.2%, 91.0% on the small, medium and large datasets. SMFCNN outperforms the other models because of two reasons (1) its multiple filters of various size and structure of hidden layers that captured high-level features from the text and (2) convolving filters of variable size (window size) can extract variable-length features (n-grams) that made it more suitable for text document classification.

Comparing the performance among the ML models on the three datasets shows that none of the models outperformed on the three datasets. NB outperforms the other models on COUNTER dataset only with 91.5% and 91.4% values of F-measure and accuracy. On naïve dataset, BayesNet outperforms with 95.1% value of both accuracy and F-measure. BayesNet also outperforms on NPUU dataset with 90.2% and 90.1% values of F-measure and accuracy. SVM polynomial performs the worst among all the models on the COUNTER dataset with 64.8% and 71.7% values of F-measure and accuracy. On naïve dataset, Random Tree gives the worst performance and achieves 70.3% value for both F-measure and accuracy. On NPUU dataset, IBK achieves the lowest value of 39.6% of F-measure while CSForest performs worst by showing 42.9% value of accuracy.

In contrast to ML classifiers, maximum accuracy of SMFCNN classifier achieves on many thousands of features is 95.4%, 93.3% and 91.8% on naïve, NPUU and COUNTER datasets respectively. The main problem of ML classifiers is that their performance depends on feature selection methods and comparative study of [12] concluded that there is no any unique feature selection method exist that works well with all ML classifiers.

VII. CONCLUSION

In this present study, an attempt has been made to attract the attention of the researchers to a resource-poor language Urdu by designing and providing a large, complex, and a multipurpose text dataset. Because of the comparisons of ML and DL models where DL models showed superior performance than ML models, this study has also opened a gate for text document classification using deep learning models. On Urdu TC task, the SMFCNN shows good performance to classify small, medium and large size datasets. Finding the optimized parameters of the SMFCNN is a time and resource exhausting process but it improves the performance of the classifier. Removing stopwords and rare words have a greater impact on large size dataset and minor impact on small size dataset. Dataset split-ratio also affects the performance of the classifier and dividing a dataset into proper training and testing sets resulted in a significant increase in performance. For high imbalanced datasets of Urdu like COUNTER and NPUU, more data is required for training than the low imbalanced dataset like a naïve dataset. SMFCNN outperforms the well-known ML classifiers on long text document classification of Urdu. SMFCNN achieves high accuracy on many thousands of features, which shows its ability to classify long text documents of Urdu. The results of SMFCNN on NPUU dataset shows that the dataset is well processed, well organized for automated text processing. It can also be used for multiple purposes other than classification like named entity recognition, text summarization, etc.

In future research, character-level CNN [50] can be used and compared with SMFCNN for Urdu TDC. SMFCNN can be applied on balanced datasets of Urdu if available. Semantic TC has become a trend in TC. Researchers can perform similar methods of semantic classification on Urdu text as described in this article [51]. Hybrid TC methods of DL and ML [4] can also be applied in the future. NPUU dataset can be used for multiple tasks of automated text processing and the dataset is available on GitHub to help researchers in future research after the publication of the presented work.

REFERENCES

[1] K. Mehmood, D. Essam, and K. Shafi, "Sentiment analysis system for roman urdu," in *Intelligent Computing* (Advances in Intelligent Systems and Computing). Cham, Switzerland: Springer, 2019, pp. 29–42, doi: 10.1007/978-3-030-01174-1_3.

[2] R. Duwairi and M. El-Orfali, "A study of the effects of preprocessing strategies on sentiment analysis for arabic text," *J. Inf. Sci.*, vol. 40, no. 4, pp. 501–513, May 2014.

[3] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*. [Online]. Available: https://arxiv.org/abs/1408.5882, doi: 10.3115/v1/D14-1181.

[4] A. Tripathy, A. Anand, and S. K. Rath, "Document-level sentiment classification using hybrid machine learning approach," *Knowl. Inf. Syst.*, vol. 53, no. 3, pp. 805–831, May 2017.

[5] G. Rao, W. Huang, Z. Feng, and Q. Cong, "LSTM with sentence representations for document-level sentiment classification," *Neurocomputing*, vol. 308, pp. 49–57, Sep. 2018.

[6] A. Hassan and A. Mahmood, "Deep learning approach for sentiment analysis of short texts," in *Proc. 3rd Int. Conf. Control, Automat. Robot. (ICCAR)*, Apr. 2017, pp. 705–710.

[7] G. Jain, M. Sharma, and B. Agarwal, "Optimizing semantic LSTM for spam detection," *Int. J. Inf. Technol.*, vol. 11, no. 2, pp. 239–250, Apr. 2018.

[8] 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.

[9] K. Riaz, "Comparison of hindi and urdu in computational context," *Int. J. Comput. Linguist. Nat. Lang. Process.*, vol. 1, no. 3, pp. 92–97, 2012.

[10] A. Daud, W. Khan, and D. Che, "Urdu language processing: A survey," *Artif. Intell. Rev.*, vol. 47, no. 3, pp. 279–311, 2017.

[11] X. Deng, Y. Li, J. Weng, and J. Zhang, "Feature selection for text classification: A review," *Multimedia Tools Appl.*, vol. 78, pp. 3797–3816, May 2018.

[12] Z. Tehseen, M. P. Akhter, and Q. Abbas, "Comparative study of feature selection approaches for urdu text categorization," *Malaysian J. Comput. Sci.*, vol. 28, no. 2, pp. 93–109, 2015.

[13] A. Ali and M. Ijaz, "Urdu text classification," in *Proc. FIT*, Abbottabad, Pakistan, Dec. 2009.

[14] M. Usman, Z. Shafique, S. Ayub, and K. Malik, "Urdu text classification using majority voting," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 8, pp. 265–273, 2016.

[15] N. H. Khan and A. Adnan, "Urdu optical character recognition systems: Present contributions and future directions," *IEEE Access*, vol. 6, pp. 46019–46046, 2018.

[16] J. Ahmad, H. Farman, and Z. Jan, "Deep learning methods and applications," in *Deep Learning: Convergence to Big Data Analytics*, M. Khan, B. Jan, and H. Farman, Eds. Singapore: Springer, 2019, pp. 31–42.

[17] M. Amajd, Z. Kaimuldenov, and I. Voronkov, "Text classification with deep neural networks," in *Proc. CEUR Workshop*, vol. 17, 1989, pp. 362–370. [Online]. Available: http://ceur-ws.org/Vol-1989/

[18] I. Siddiqi, C. Djeddi, A. Raza, and L. Souici-meslati, "Automatic analysis of handwriting for gender classification," *Pattern Anal. Appl.*, vol. 18, no. 4, pp. 887–899, 2015.

[19] C. C. Aggarwal, "Text sequence modeling and deep learning," in *Machine Learning for Text*, C. C. Aggarwal, Ed. Cham, Switzerland: Springer, 2018, pp. 305–360.

[20] M. Katnoria, V. Singh, and R. Kumar, "Punjabi document classification using vector evaluation method," in *Proc. Int. Conf. Comput. Methodol. Commun. (ICCMC)*, Jul. 2017, pp. 940–944.

[21] S. Song, H. Huang, and T. Ruan, "Abstractive text summarization using LSTM-CNN based deep learning," *Multimedia Tools Appl.*, vol. 78, no. 1, pp. 857–875, Feb. 2018.

[22] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," *Inf. Process. Manage.*, vol. 50, no. 1, pp. 104–112, Jan. 2014.

[23] A. Ayedh, G. Tan, K. Alwesabi, and H. Rajeh, "The effect of preprocessing on arabic document categorization," *Algorithms*, vol. 9, no. 2, p. 27, Apr. 2016.

[24] M. Çağataylı and E. Çelebi, "The effect of stemming and stop-word-removal on automatic text classification in turkish language," in *Neural Information Processing*. Cham, Switzerland: Springer, 2015, pp. 168–176.

[25] R. Wongso, F. A. Luwinda, B. C. Trisnajaya, O. Rusli, and Rudy, "News article text classification in indonesian language," *Procedia Comput. Sci.*, vol. 116, pp. 137–143, Jan. 2017.

[26] H. Liang, X. Sun, Y. Sun, and Y. Gao, "Text feature extraction based on deep learning: A review," *EURASIP J. Wireless Commun. Netw.*, vol. 2017, no. 1, pp. 1–2, Dec. 2017.

[27] J. D. Prusa and T. M. Khoshgoftaar, "Improving deep neural network design with new text data representations," *J. Big Data*, vol. 4, no. 1, p. 7, Mar. 2017.

[28] A. Z. Syed, A. M. Martinez-Enriquez, A. Nazir, M. Aslam, and R. H. Basit, "Mining the urdu language-based Web content for opinion extraction," in *Pattern Recognition*. Springer, 2017, pp. 244–253, doi: 10.1007/978-3-319-59226-8_24.

[29] D. Becker and K. Riaz, "A study in urdu corpus construction," in *Proc. 3rd Workshop Asian Lang. Resour. Int. Standardization*, 2004, pp. 1–5.

[30] D. Cecchini and L. Na, "Chinese news classification," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Jan. 2018, pp. 681–684.

[31] D. Kılınç, A. Özçift, F. Bozyigit, P. Yıldırım, F. Yücalar, and E. Borandag, "TTC-3600: A new benchmark dataset for turkish text categorization," *J. Inf. Sci.*, vol. 43, no. 2, pp. 174–185, Dec. 2015.

[32] M. Sharjeel, R. M. A. Nawab, and P. Rayson, "COUNTER: Corpus of urdu news text reuse," *Lang. Resour. Eval.*, vol. 51, no. 3, pp. 777–803, Sep. 2016.

[33] L. Yin, Y. Ge, K. Xiao, X. Wang, and X. Quan, "Feature selection for high-dimensional imbalanced data," *Neurocomputing*, vol. 105, pp. 3–11, Apr. 2013.

[34] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Syst. Appl.*, vol. 73, pp. 220–239, May 2017.

[35] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, "A C-LSTM neural network for text classification," 2015, *arXiv:1511.08630*. [Online]. Available: https://arxiv.org/abs/1408.5882, doi: 10.3115/v1/D14-1181.

[36] G. A. Abandah, A. Graves, B. Al-Shagoor, A. Arabiyat, F. Jamour, and M. Al-Taee, "Automatic diacritization of arabic text using recurrent neural networks," *Int. J. Document Anal. Recognit.*, vol. 18, no. 2, pp. 183–197, Mar. 2015.

[37] A. Jabbar, S. Iqbal, M. U. G. Khan, and S. Hussain, "A survey on urdu and urdu like language stemmers and stemming techniques," *Artif. Intell. Rev.*, vol. 49, no. 3, pp. 339–373, 2018.

[38] M. M. Mirończuk and J. Protasiewicz, "A recent overview of the state-of-the-art elements of text classification," *Expert Syst. Appl.*, vol. 106, pp. 36–54, Sep. 2018.

[39] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *ACM SIGKDD Explor. Newslett.*, vol. 11, no. 1, pp. 10–18, 2009.

[40] S. Alshahrani and E. Kapetanios, "Are deep learning approaches suitable for natural language processing?" in *Natural Language Processing and Information Systems*. Cham, Switzerland: Springer, 2016, pp. 343–349.

[41] U. Pal, "Language, script, and font recognition," in *Handbook of Document Image Processing and Recognition*, D. Doermann and K. Tombre, Eds. London, U.K.: Springer, 2014, pp. 291–330.

[42] H. Chen, S. McKeever, and S. J. Delany, "A comparison of classical versus deep learning techniques for abusive content detection on social media sites," in *Social Informatics*. Springer, 2018, pp. 117–133, doi: 978-3-030-01129-1_8.

[43] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Comput.*, vol. 22, no. S1, pp. 949–961, Sep. 2017.

[44] S. Krig, "Feature learning and deep learning architecture survey," in *Computer Vision Metrics*, S. Krig, Ed. Cham, Switzerland: Springer, 2016, pp. 375–514.

[45] H.-Y. Lu, M. Zhang, Y.-Q. Liu, and S.-P. Ma, "Convolution neural network feature importance analysis and feature selection enhanced model," *Ruan Jian Xue Bao/J. Softw.*, vol. 28, no. 11, pp. 2879–2890, 2017.

[46] R. H. Basit, M. Aslam, A. M. Martinez-Enriquez, and A. Z. Syed, "Semantic similarity analysis of urdu documents," in *Pattern Recognition*. Springer, 2017, pp. 234–243, doi: 10.1007/978-3-319-59226-8_23.

[47] Q. A. Al-Radaideh and M. A. Al-Abrat, "An arabic text categorization approach using term weighting and multiple reducts," *Soft Comput.*, vol. 23, no. 14, pp. 5849–5863, Jun. 2018.

[48] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," 2016, *arXiv:1609.04836*. [Online]. Available: http://arxiv.org/abs/1609.04836

[49] G. Liu and J. Guo, "Bidirectional LSTM with attention mechanism and convolutional layer for text classification," *Neurocomputing*, vol. 337, pp. 325–338, Apr. 2019.

[50] M. Sato, R. Orihara, Y. Sei, Y. Tahara, and A. Ohsuga, "Text classification and transfer learning based on character-level deep convolutional neural networks," in *Agents and Artificial Intelligence*. Springer, 2018, pp. 62–81, doi: 10.1007/978-3-319-93581-2_4.

[51] B. Altınel and M. C. Ganiz, "Semantic text classification: A survey of past and recent advances," *Inf. Process. Manage.*, vol. 54, no. 6, pp. 1129–1153, Nov. 2018.

MUHAMMAD PERVEZ AKHTER received the B.S. and M.S. degrees in computer science from the University of Sargodha, Punjab, Pakistan, in 2009 and 2013, respectively. He is currently pursuing the Ph.D. degree in software engineering with the School of Software and Microelectronics, Northwestern Polytechnical University, Xi'an, China. He has eight years of teaching experience. He has two research articles in text processing. His research interests include natural language processing, text processing, data mining, image processing, and computer vision.

ZHENG JIANGBIN received the Ph.D. degree from the IT Academy, China.

Since 2008, he has been a Professor with Northwestern Polytechnical University, Xi'an, China, where he is currently acting as the Dean of the School of Software and Microelectronics. He has presided many scientific research projects such as the National Natural Science Foundation, the 863 Program, and provincial and ministerial funds, and has published more than 100 articles. His research interests are image processing and computer vision, the IoTs, big data processing, and embedded computing technology.

IRFAN RAZA NAQVI received the bachelor's (B.S.) degree in computer science from Bahaudin Zakariya University, Multan, Pakistan, in 2012, and the master's degree in software engineering from Air University, Pakistan. He is currently pursuing the Ph.D. degree with Northwestern Polytechnical University, Xi'an, China. His current research interests include security and privacy concerns in the context of Internet of Things (IoTs), machine learning, and big data.

MOHAMMED ABDELMAJEED received the B.Sc. degree from Omdurman Islamic University, Khartoum, Sudan, in 2009, and the M.Sc. degree from Alneelain University, Khartoum, in 2015. He is currently pursuing the Ph.D. degree in computer science with Northwestern Polytechincal University, Xi'an, China. He has four years of teaching experience. His research areas are text processing and computer vision.

ATIF MEHMOOD received the B.S. degree in computer science from COMSATS University Islamabad, Pakistan, in 2015, and the M.S. degree in computer science from Riphah International University, Islamabad, Pakistan, in 2018. He is currently pursuing the Ph.D. degree with Xidian University, Xi'an, China. His research interests are in machine learning, deep learning, and medical image processing.

MUHAMMAD TARIQ SADIQ received the B.Sc. degree (Hons.) from the Comsats Institute of Information Technology, Lahore, Pakistan, in 2009, and the M.Sc. degree in electrical engineering from the Blekinge Institute of Technology, Sweden, in 2011. He is currently the Ph.D. Scholar with Northwestern Polytechnical University, Xi'an, China. Previously, he was an Assistant Professor with the Sharif College of Engineering and Technology (SCET) which is affiliated with the University of Engineering and Technology Lahore and a Lecturer with the University of South Asia. He was also the Project Manager with SCET to manage final year student's projects, Patron of IEEE-SCET Student Branch, and a Lifetime Member of Pakistan Engineering Council, Islamabad, Pakistan. He is also an Assistant Professor with the Electrical Engineering Department, The University of Lahore. His research interests include biomedical signal analysis and classification.