IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 19, 2021, accepted June 3, 2021. Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2021.3087827

Improving Hate Speech Detection of Urdu Tweets Using Sentiment Analysis

MUHAMMAD Z. ALI, EHSAN-UL-HAQ, SAHAR RAUF, KASHIF JAVED, AND SARMAD HUSSAIN
Center for Language Engineering, Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology, Lahore 54890, Pakistan

Corresponding author: Muhammad Z. Ali (muhammad.zain@kics.edu.pk)

ABSTRACT Sentiment Analysis is a technique that is being used abundantly nowadays for customer reviews analysis, popularity analysis of electoral candidates, hate speech detection and similar applications. Sentiment analysis on tweets encounters challenges such as highly skewed classes, high dimensional feature vectors and highly sparse data. In this study, we have analyzed the improvement achieved by successively addressing these problems in order to determine their severity for sentiment analysis of tweets. Firstly, we prepared a comprehensive data set consisting of Urdu Tweets for sentiment analysis-based hate speech detection. To improve the performance of the sentiment classifier, we employed dynamic stop words filtering, Variable Global Feature Selection Scheme (VGFSS) and Synthetic Minority Optimization Technique (SMOTE) to handle the sparsity, dimensionality and class imbalance problems respectively. We used two machine learning algorithms i.e., Support Vector Machines (SVM) and Multinomial Naïve Bayes' (MNB) for investigating performance in our experiments. Our results show that addressing class skew along with alleviating the high dimensionality problem brings about the maximum improvement in the overall performance of the sentiment analysis-based hate speech detection.

INDEX TERMS Sentiment analysis, hate speech, data sparsity, highly skewed classes, high-dimensional feature vector.

I. INTRODUCTION

Sentiment analysis is one of the trending topics of research regarding Natural Language Processing and text classification. Using this technique, one is able to extract the semantic sense out of a given word, sentence or a document and therefore being widely used in various areas of life from product reviews analysis to probing the popularity of candidates contesting in the elections. There are three main types of sentiment analysis i.e. document level, sentence level and entity/aspect level [1].

In document level sentiment analysis, the semantic orientation of the entire document is determined based on the content of the whole document. It is normally used for the blogs written about a single product. In sentence level sentiment analysis, the semantic orientation of each sentence is extracted. It is useful for the analysis of product or movie reviews. Finally, the aspect level sentiment analysis is used to fine grain the individual sentences to check for the semantic orientation of a particular entity in a sentence. This type of sentiment analysis may result in multiple entities and multiple sentiments in the same sentence [2]. Apart from these three types of sentiment analysis, conversational sentiment analysis has been introduced recently which is different from sentence-level sentiment analysis in a way that it captures context information in dialogues as well [3].

There are three general methods for performing sentiment analysis i.e. lexical methods, machine learning based methods and hybrid methods (combination of lexical and machine learning) [4]. Lexical methods are rule based which involves the incorporation of a predefined lexicon consisting of the positive, negative words and their intensities [4]. The words of a given sentence or document are looked up in the lexicon and the individual weights are accumulated and compared against the defined threshold to output the final 'positive', 'negative' or 'neutral' label. The most commonly used lexical method for sentiment analysis of English data is Valence Aware Dictionary for sEntiment Reasoning (VADER) [5].The problem with the lexical methods is that they are unable to capture the underlying semantics in most of the cases since they only rely on the presence or absence of extreme words, which deems unsatisfactory results [6].

On the other hand, machine learning based methods involve the preparation of data corpus and then the algorithms

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

automatically learn using the features and the labels and improve their performance from statistical experience. Training a machine learning text classifier is generally a three steps process: (i) feature extraction or representation (ii) feature selection and (iii) classification [7]. Since text data is unstructured by nature, that is why it is converted into a vector of numerical values. The most common way of doing that is Bag-of-Words (BoW) representation [8]. In this technique, unique words of the entire corpus are extracted and each sample in the document is usually represented by either of the three ways: (i) presence/absence of words in the dictionary (binary) [9] (ii) number of times given word of the dictionary appeared in the sample (term frequency) [10] (iii) term-frequency inverse document frequency (TFIDF) [11]. The selected text representation is then fed to machine learning algorithms for classifier training. The quality of labeling and the quantity of the data set holds particular importance in this process. A well labeled data set ensures separability and therefore a better boundary for the classifier, whereas, the more data we have, the better will be the chances for the algorithm to learn from the statistical experience.

Machine learning-based sentiment classification for tweets encounters three problems namely, high sparsity, high-dimensional feature vectors and highly skewed classes. A large number of people use Twitter to express their opinions about any topic of life in limited words, which results in a vocabulary of thousands of unique words causing highly sparse and high-dimensional features vector representation of input data [12]. Highly sparse input data has information scarcity making it computationally complex for the machine learning models to learn. Similarly, high-dimensional features have asymmetrical discriminating power which makes a large number of words in the vocabulary either irrelevant or redundant [13]. These irrelevant or redundant words confuse the classifier resulting in the loss of accuracy. Therefore, dimensionality reduction or feature selection is applied to improve the performance of the classifier. Supervised feature selection methods can be of three types i.e. filters, wrappers and embedded. In text classification, filter methods are normally used since they are simple and computationally efficient as compared to other types of feature selection techniques [7]. The filter methods deal with ranking the features and selecting the top N of them while discarding the rest. In the feature ranking process, the relevance scores are calculated using the measures such as information gain (IG) [14], Gini index (GI) [15] and distinguishing feature selector (DFS) [16] etc. The local scores calculated from measures like these indicate the relevance of a given feature with a particular class whereas, the global scores are simply some function applied on the local scores of a given word, calculated for each individual class.

Similarly, class skew refers to varying, unequal occurrences of individual class samples in the data set [17]. A highly skewed data set may cause the classifier to be biased and therefore loose accuracy. It is the most common problem while preparing corpus for training sentiment classifier as getting equal amount of data per class is insurmountable. In some cases, the number of positive labeled samples are unusually high, whereas, in others negative or neutral samples are high. Among the most common approaches to deal with the imbalance dataset are oversampling [18] and under-sampling [19]. Although there are several past studies which have separately addressed the sparsity, dimensionality or class skew problems but none of them have simultaneously handled these problems to improve the performance of the sentiment classifier. The summary of our research work is shown in Figure 1. We started off by collecting the Urdu hateful data from the Twitter. The expert linguists in our team annotated the data on aspect and sentiment levels as per the guidelines discussed in Section III-A. After data evaluation and analysis, we employed machine learning algorithms to train a baseline hate speech classifier on the newly collected corpus as discussed in Section III-A. After that we improved the performance of the hate speech classifier by successively resolving the above discussed problems i.e., dimensionality, sparsity and class skew. We finally compared the results with the baseline model to analyze the performance improvement achieved by resolving each problem.

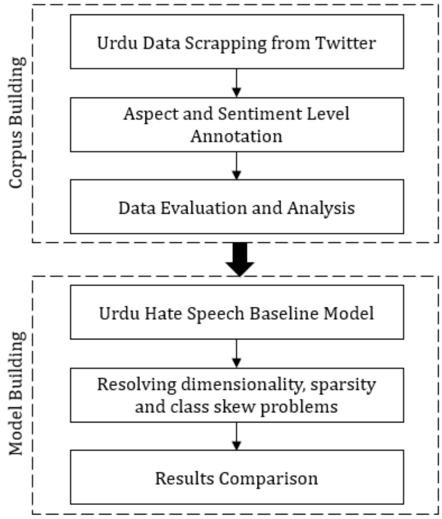
Corpus Building
Urdu Data Scrapping from Twitter
Aspect and Sentiment Level Annotation
Data Evaluation and Analysis
Model Building
Urdu Hate Speech Baseline Model
Resolving dimensionality, sparsity and class skew problems
Results Comparison

FIGURE 1. Summary of our work.

The major contributions of our research work in this paper are summarized:

- Preparation of the first comprehensive publicly available hate speech text corpus with aspect and sentiment level annotations for Urdu which is a low resource language.
- We have simultaneously analyzed the three most frequent problems (i.e., highly skewed classes, high dimensional feature vector and highly sparse data representation) encountered in sentiment analysis of Twitter data.
- For future research, this study sets new baseline results for Urdu hate speech text classification.

The rest of the paper is organized as follows. In Section 2, related works have been discussed. In Section 3, we have discussed the methodology on how we built the corpus and the techniques we used to address the class imbalance problem. In Section 4, performance evaluation parameter is discussed. In Section 5, the detailed results and discussion is presented. Conclusions and future dimensions of this research have been presented in Section 6.

II. RELATED WORK

Recent studies have explored the use of sentiment analysis in various areas of life. In [20], authors have used sentiment analysis to determine the severity of traffic accidents. Similarly, the use of sentiment analysis for health care monitoring by employing heterogeneous data has been discussed in [21]. In [22], authors have used sentiment and emotion classification to determine the avalanche point of an epidemic outbreak. The common thing in many of the recent studies regarding sentiment analysis is the use of Twitter as the primary source of data [23]-[25]. It is owing to the fact that Twitter is a universal microblogging website and it allows the users to express their thoughts in limited characters which makes the preprocessing part easy for the researchers [26]. However, as discussed in Section I, machine learning-based sentiment classification for tweets encounters three problems namely, high sparsity, high-dimensional feature vectors and highly skewed classes.

Researchers have proposed different methods to deal with these problems for text classification. These methods include feature selection, sampling, ensemble and modified word representations. Saif *et al.* [12] presented two different approaches to deal with the sparsity problem in sentiment analysis of twitter data i.e. incorporation of semantic features and semantic concepts. They showed that the interpolation and addition of sentiment topic features not only decrease the sparsity but also give superior results than the baseline model. In [24], authors proposed a dynamic stop words filtering to deal with the data sparsity problem in sentiment analysis of short texts. They concluded that removing single frequency terms from the vocabulary reduces the data sparsity to a huge amount, whereas, using mutual information (MI) to discard the irrelevant terms increases the accuracy of the classifier as well. Similar work regarding feature selection to deal with data imbalance and sparsity problem is presented in [1]. Authors proposed the use of Gini Index (GI) feature selection with SVM classifier for sentiment analysis. Their results showed that the Gini Index (GI) feature selection scheme outperforms other schemes such as maximum relevance, chi-square, information gain and correlation. In [27], authors proposed variable global feature selection scheme (VGFSS) for automatic classification of text documents. Using the unique words from the text corpus, they ranked the features by employing global distinguishing feature selector (DFS) and tagging them with most relevant class category based on the local distinguishing feature selector (DFS) score. They concluded that the variable selection of the features drastically improves the results in highly unbalanced data sets. An improved version of DFS namely, inherent distinguished feature selector (IDFS) is presented in [28]. Authors applied this feature selection scheme on five benchmark data sets and compared the results against five well know FS-metrics. They concluded that the inherent distinguished feature selector (IDFS) not only selects smaller subsets but also outperforms the existing FS metrics.

In contrast to the feature selection methods, researchers have also presented resampling the imbalance data set as a viable solution to the problem. In [29], authors proposed distributional random oversampling for imbalanced text classification. They argued that this method outperforms the existing oversampling methods since it generates new random minority-class synthetic documents by exploiting the distributional properties of the words in the vocabulary. A similar work regarding oversampling has been presented in [18]. Authors incorporated probability distribution of the features for generative oversampling. In [30], authors proposed a unique method of under-sampling to deal with the imbalanced data problem. Their technique focused on acquiring information rich samples from the majority class while discarding the rest to train the model. Their results indicated that the proposed selection technique improves the sensitivity compared to weighted space-vector machine.

Although bag-of-words (BoW) representation is the most common way to represent textual data but for short texts, it results in sparsity and causes the model to perform poorly especially for imbalanced data. Al-Anzi and AbuZeina [31], proposed Markov chains [32] or probability transition matrix for each class in the dataset thereby removing the need of bag-of-words model. They concluded that the proposed method enhances the F1-measure by 3.47%. Similarly, ensemble methods are also used to tackle the class imbalance problem. In [33], authors have presented a comprehensive overview of the works done in the past regarding ensemble techniques such as random forest classifier (RFC), weighted random forest classifier (WRFC), balanced random forest classifier (BRFC) and oversampling techniques to deal with the class imbalance problem. They postulated that dynamic integration techniques are the most useful to deal with the class skew problem.

Few recent studies [34], [35] have hypothesized the use of word embeddings with deep learning algorithms as the only solution to the data sparsity and dimensionality problem. However, the necessity for a huge amount of data for optimal training of deep learning algorithms makes them unfit for data sets of small size, thereby, reinforcing the need of machine learning-based solutions [36].

III. METHODOLOGY

In this section we have discussed the methodology on how we have built the corpus for this study and the techniques we used to improve the performance of sentiment analysis-based hate speech classifier for Urdu.

A. CORPUS BUILDING

Since there is no existing data set available for Urdu hate speech with aspect and sentiment level annotation, therefore, we had to build our own. To build the corpus for this study, we followed two steps. In the first step, we developed a 530 words sentiment lexicon and in the second step, we defined the target domains (religious, national security, ethnicity) and their respective keywords (e.g. shia, wahabi, government, Punjabi etc.). Afterward, the combinations of lexicon words and the keywords from the selected domains were used to extract the desired tweets from Twitter. The process of Lexicon development and the domain selection is discussed in detail below:

1) URDU SENTIMENT LEXICON

A list of 530 words was tagged. Guidelines were developed for assigning scores to the words based on their level of perceived positivity or negativity. Extreme negative words are those offensive words which express hate, violence, threats, accusations, profanity, and general insults towards a target group. To assign a score to a given urdu word, the following 5-point scale given in Table 1 was used.

TABLE 1. Five point scale for assigning scores.

Number	Meaning
-2	Highly offensive
-1	Offensive
0	Neutral
1	Positive
2	Highly positive

Two annotators worked on labeling the data and their job was to read the words carefully and find whether the word could be perceived as offensive or not. After understanding the word, the annotators assigned a score to it using the five-point scale as mentioned above. Profane or obscene words were assigned a score of -2. The offensive words of English written in Urdu were assigned scores according to their polarity e.g., /Donkey/ /dɔːŋkiː/ word was assigned -1 score.

2) DOMAIN SELECTION

Three domains were selected for building the Urdu language sentiment analysis corpus i.e. Religious, National Security and Ethnicity. The purpose was to identify the hate speech found in Urdu tweets against religious groups, national security institutions, and ethnic groups. Keywords were selected to grab the data related to each domain. 14 keywords were selected in religious domain and 10 out of them were Islamic sects i.e., /Shia/ /ʃiːɑː/, /Sunni/ /sʊnniː/, and /Vahabi/ /vəhaːbiː/ etc., whereas, remaining 4 were other religions practiced in Pakistan. Similarly, different establishments of Pakistan related to government, provinces, border security, judiciary, army and intelligence agencies etc., were selected as the subjects for the national security domain. Approximately, 30 subjects were selected in this domain. Subjects in the third domain were selected based on ethnic groups found in Pakistan[1] e.g., Punjabis, Pashtuns, Sindhis, Saraikis, Muhajirs, Baloch, and Paharis, etc. Some keywords related to gender biases and derogatory terms used to insult people were also selected in the third domain but very less data was found on those keywords that is why we limited the third domain to just Ethnicity.

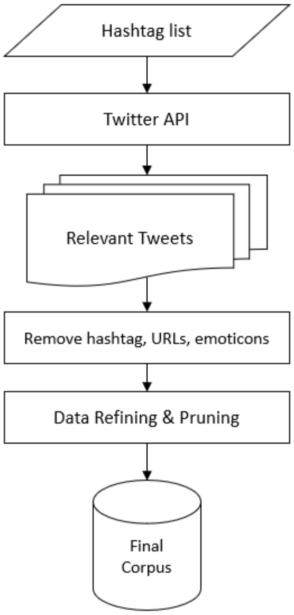

FIGURE 2. Data collection process from Twitter.

3) DATA COLLECTION

The process for data collection is shown in Figure 2. We used the concatenation of above discussed lexicon and keywords of selected domains as the hash tags list and passed that on to the Twitter API [37] for acquiring relevant tweets. We repeated the process over for 6 months to be able to collect sufficient data for building the corpus. Initial preprocessing was also performed to remove hashtags, emoticons and URLs from the tweets. During the entire process of data collection, manual refining was also done to correct the visible grammatical mistakes and removal of null entries from each subset of the data. At the end of the sixth month from the beginning of the data collection, we were able to gather a total of 16k unique tweets after final pruning.

4) DATA ANNOTATION

The annotation was carried out in three steps:

Firstly, the linguists had to read the sentence or tweet carefully and segment the words if they were joined. Word segmentation was necessary because we had to analyze the word combinations or their relationships which could not be possible in the case of joint words depicting no meaning.

[1] https://en.wikipedia.org/wiki/Ethnic_groups_in_Pakistan

Secondly, the linguists had to identify the presence of any of the selected domains in the sentence and marked the Type as per fields below:

- Type field ‘0’ was marked for National Security domain.
- Type field ‘1’ was marked for Religious domain.
- Type field ‘2’ was marked for Ethnic distinctions.

Thirdly, the linguists marked the score as 1 or −1. Here score is denoting the presence or absence of offensive content in a tweet. Score −1 means tweet is offensive and score 1 means tweet is normal. A summary of the annotation scheme is given in Table 2.

TABLE 2. A summary of annotation scheme.

Annotation scheme	Assigned values
Type field	0, 1, 2
Score	-1, 1

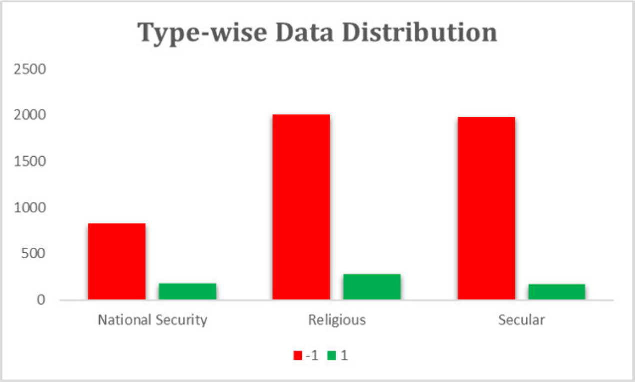

FIGURE 3. Data distribution of relevant corpus. ‘-1’ indicates hateful tweets, ‘1’ indicates normal tweets.

5) DATA EVALUATION

The performance of the system is based on the quality of the data given to the system. We also evaluated our data by keeping this in mind. The data generated on weekly basis by the linguist was also tested by another expert linguist. A 10% randomly selected data from the weekly tranches was marked by the expert linguist. The inter annotator agreement (IAA) was then measured by using Cohen’s Kappa statistics [38]. After calculating the kappa scores for each tranche, the average values were calculated. In our case, as we marked the data on two levels so we calculated kappa score for two-levels i.e., Type field and Score. The observed agreement on the Type field level was 0.896 and the score level was 0.80. Hence, a 0.799 kappa score for the Type field level and a 0.708 kappa score for the Score level was achieved on the data. The agreement is slightly better at Type field level than Score level because subjectivity sometimes hinders the marking of the annotator at the Score level. But the results showed that we have achieved substantial agreement on our data. The type and score-wise distribution of relevant data is shown in Figure 3.

B. BASELINE MODEL

Figure 3 shows that the final data set is highly skewed towards the negative class (hate class). To check the effectiveness of the solutions to the problem presented in this paper, and to compare the results, we used a baseline model. The pipeline for the baseline model is shown in Figure 4. We did initial preprocessing on the training data which involved the removal of punctuation symbols and stop words. Stop words refer to the most commonly used words in a language which don’t have any contribution in putting a given sample in a particular class. After that, we made use of term frequency-inverse document frequency (TF-IDF) [11] to represent the entire data set. Two machine learning algorithms were used separately to train the baseline classifier i.e. Space Vector Machines (SVM) and Multinomial Naïve Bayes (MNB) using the scikit-learn implementation for python [39]. MNB makes use of probability theory and Bayes’ theorem with an assumption of naïve independence among the features to learn from the data set and predict the class of unknown inputs [40]. SVM, on the other hand, works on the basis of maximal margin classifier. It is basically an optimization problem where we have to maximize the margin between the decision boundary (or hyperplane) and the nearest lying training samples in the feature space [41]. After the training process, we provided the resultant classifier with the test data to output the predicted labels. The predicted labels were validated against the actual labels to display the classification performance report with the selected parameters.

C. ADDRESSING PROBLEMS OF SENTIMENT ANALYSIS

In this paper we have made use of dynamic stop words filtering, variable global feature selection scheme (VGFSS) and SMOTE to address the high sparsity, high dimensionality and class skew problems respectively.

1) THE SPARSITY PROBLEM

Although bag-of-words (BoW) representation is the most common way to represent textual data but for short texts, it results in sparsity and causes the model to perform poorly especially for imbalanced data. To handle this problem we used dynamic stop words filtering. The word “dynamic” refers to the fact that the stop words list is taken from the data itself and is not fixed. The vocabulary analysis of our data indicated that about 79% of the terms had occurred less than five times in the entire corpus. Table 3 shows the frequency-wise distribution of terms in the vocabulary of corpus. Further inspection of the terms falling in the window of ‘less than 5’, indicated that almost all the terms having the frequency less than or equal to two are meaningless and therefore can be excluded from the final feature vector. This approach resulted in a great decrease in sparsity and therefore allowed the machine learning models to better learn from the most relevant features.

2) THE DIMENSIONALITY PROBLEM

To deal with the high dimensionality problem, we applied VGFSS on our data set. VGFSS is a type of global filter-based feature selection scheme [27]. The idea is to determine the most relevant features in order to reduce the feature

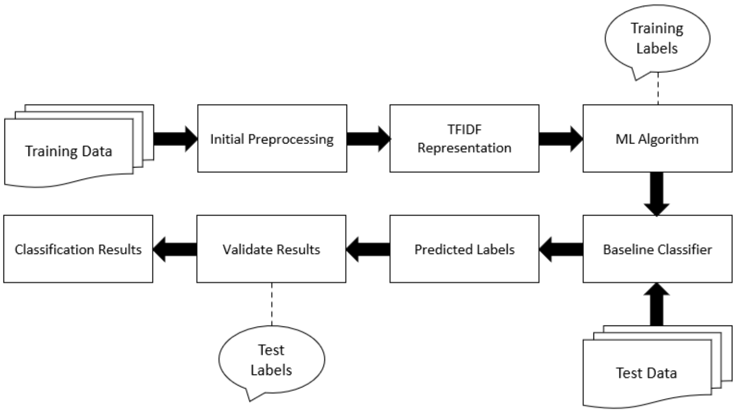

FIGURE 4. Pipeline for the baseline model.

TABLE 3. Frequency-wise Distribution of Vocabulary Terms.

Frequency of Terms	Count	Percentage
Greater than 500	16	0.08%
389-482	8	0.04%
293-388	14	0.07%
197-292	29	0.14%
101-196	143	0.67%
5-100	4163	19.52%
Less than 5	**16949**	**79.49%**
Total	21322	100%

space (dimensions) and improve the decision-making capabilities of the classifier. VGFSS approach for selecting the terms is described below:

(i) The corpus is split into training and testing sets $D = D_{train} + D_{test}$.

(ii) Preprocessing is applied on the train set.

(iii) TF-IDF vectorizer is applied on the train set as a result of which vocabulary of terms is generated $V = \{t_1, t_2, t_3, \ldots t_i\} \, \forall i = 1, 2, \ldots, m$.

(iv) Global and local Distinguishing Feature Selector (DFS) is computed for each term in V to assign the final feature score as shown in the following equation.

$$FSS\_Score(t_i) = DFS(t_i, C_j) \tag{1}$$

where $C_j \, \forall j = 1, 2, \ldots, k$, represents the available classes in the data set.

(v) The features are then arranged in the descending order of the assigned global FSS score.

(vi) Each feature t_i is assigned a class C_j based on the maximum local FSS score.

(vii) Total number of features in each class C_j are then calculated.

(viii) If N is the total number of features required in the final set, the following equation is used to hand pick the variable number of features from each class:

$$Variable_{split}(C_j) = count(C_j) \times \frac{N}{TFC} \tag{2}$$

where TFC represents the total feature count in the vocabulary V and $count(C_j)$ represents the total number of features which belong to the class C_j.

(ix) Final Feature Set (FFS) consisting of N features is formed by removing the features other than the variable split of the class C_j from the vocabulary of terms V as described above.

3) THE CLASS SKEW PROBLEM

In machine learning, resampling refers to the methods used to reconstruct the data set in order to add balance to the original corpus. Under-sampling is one way of doing that where the samples are removed from the majority class. Similarly, over-sampling is the way to synthetically or randomly add samples to the minority class to balance out the corpus. In this study, we have also analyzed the performance of Synthetic Minority Oversampling Technique (SMOTE) [42] which was used to synthetically generate samples for the minority class in order to handle the class skew problem of our data set. In this technique, synthetic examples are generated by operating in "feature space" rather than "data space". The samples from the minority class are taken and synthetic examples are introduced along the line segments joining any/all of the k minority class nearest neighbors. The value of k nearest neighbors depends upon the amount of over-sampling required. SMOTE can't be directly applied on the text data. Hence, in our case we converted our training data into TF-IDF representation and then applied this technique to generate synthetic samples for the minority class. Therefore, we inserted the over-sampling block between TF-IDF and ML Algorithm as shown in Figure 5. The rest of the process stays the same as the baseline system in Figure 4.

IV. PERFORMANCE EVALUATION

To evaluate the effectiveness of the proposed solutions to the class imbalance, sparsity and dimensionality problems and to compare the results, we used micro F1 measure and 5-fold cross validation. The formula for micro F1 is given in the

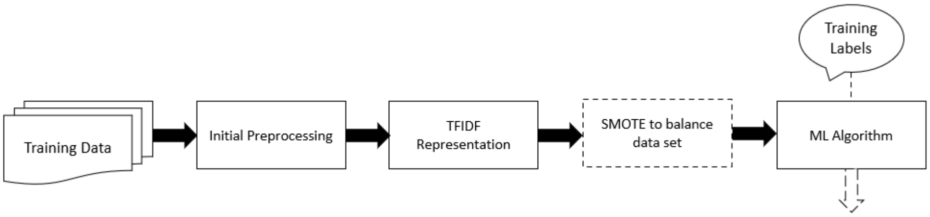

FIGURE 5. Resolving class skew by adding SMOTE block between TFIDF and ML Algorithm in the baseline pipeline.

following equation:

$$Micro\, F_1 = 2 \times \frac{precsion^{\mu} \times recall^{\mu}}{precsion^{\mu} + recall^{\mu}} \quad (3)$$

The term 'μ' represents the micro averaging of a given value. *Precision* represents the ratio of correctly predicted labels by the classifier in a given class to the total predictions made by the classifier in that class. On the other hand, *recall* represents the ratio of correctly predicted labels by the classifier in a given class to the actual number of labels in that class. *F1 measure* is just the harmonic mean of the two measures as shown in the above equation.

V. RESULTS

In this section we have presented the performance evaluation results. After getting the results for the baseline model, we improved the classifier performance by alleviating the sparsity, dimensionality and class skew problems one-by-one. The best results are bolded in the subsequent tables.

A. BASELINE MODEL

For the baseline model, we trained two machine learning models i.e. SVM and MNB by considering the entire data set as a single problem and using the sequence of steps mentioned in Section III-B. The micro F1 values of both models obtained after 5-fold cross validation are shown in Table 4. In this case, all the features were kept in the final feature set (FFS).

TABLE 4. Performance of baseline models.

5-fold Cross Validation	
Models	micro F1
Baseline Multinomial Naïve Bayes' (MNB)	0.5614
Baseline Space Vector Machines (SVM)	**0.6261**

B. IMPROVEMENT BY ADDRESSING HIGH SPARSITY

To alleviate the sparsity of the input matrix, we used dynamic stop words filtering as discussed in Section III-C1. After repeating the training process for both SVM and MNB, we got an improvement in the micro F1 values as shown in the Table 5. This improvement is due to the fact that removal of low frequency terms from the feature space allows better learning for the models.

TABLE 5. Results after alleviating sparsity by employing dynamic stop words filtering.

5-fold Cross Validation	
Models	micro F1
Multinomial Naïve Bayes' (MNB)	0.5945
Space Vector Machines (SVM)	**0.6308**

TABLE 6. Results after resolving class skew problem using SMOTE.

5-fold Cross Validation	
Models	micro F1
Multinomial Naïve Bayes (EFS)	0.8960
Multinomial Naïve Bayes (HFS)	0.9241
Space Vector Machines (EFS)	0.9787
Space Vector Machines (HFS)	**0.9852**

C. IMPROVEMENT BY ADDRESSING HIGH DIMENSIONALITY

After alleviating the sparsity problem, we made use of VGFSS as described in Section III-C2 to reduce the high dimensionality and select the most relevant features from the data set for model training. We experimented with different values of N to get variable number of features from each class and recorded the results. Additionally, we employed VGFSS in two ways i.e., by selecting features from entire features set and by selecting features after removing the low frequency terms from the feature set. Our results showed that the maximum micro F1 score of 0.91 was achieved at VGFSS ($N = 500$) for hate classifier trained using Multinomial Naïve Bayes, whereas, for SVM trained hate classifier, we got a maximum micro F1 score of 0.927 at VGFSS ($N = 1000$). The variation of micro F1 scores with different values of N for MNB and SVM are shown in figures 6 and 7 respectively. The blue line indicates the performance trend of VGFSS when features are taken from entire features set whereas, orange line indicates the performance trend when features are selected after removing the low frequency terms from the feature set. Therefore, addressing the high dimensionality problem by selecting the most relevant features using VGFSS resulted in a significant increase in micro F1 values for both MNB and SVM.

D. IMPROVEMENT BY ADDRESSING CLASS SKEW

After addressing the sparsity and dimensionality problem, we finally handled class skew problem using SMOTE as

TABLE 7. Predicted labels by SVM Pipelines on some samples from the data set along with their English translation in parenthesis. 'P1': predicted labels after resolving sparsity, 'P2': predicted labels after resolving dimensionality, 'P3': predicted labels after resolving class skew.

Samples (Translated from Urdu)	Actual Class	Baseline	P1	P2	P3
Shiites are the worst infidels on earth.	**Hate**	Normal	**Hate**	**Hate**	**Hate**
Pathans are right to say Punjabis are shameless.	**Hate**	Normal	**Hate**	**Hate**	**Hate**
According to Human Minister there is not much difference between ordinary Pashtoons and Taliban.	**Hate**	Normal	Normal	**Hate**	**Hate**
Balochi was a patriot yesterday, Balochi is still a patriot today.	**Normal**	Hate	Hate	**Normal**	**Normal**
We Sindhis know how to sacrifice without caring for any profit or loss.	**Normal**	Hate	Hate	Hate	**Normal**
Sectarianism, discrimination, violence are unacceptable.	**Normal**	Hate	Hate	Hate	**Normal**

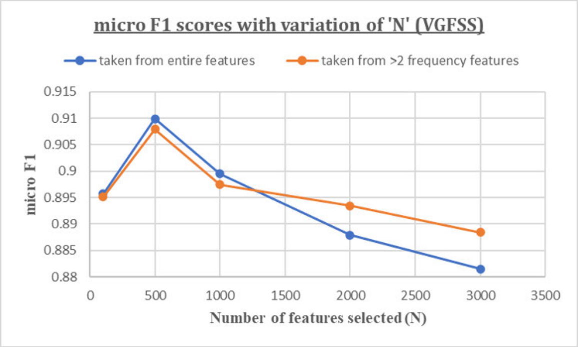

FIGURE 6. Performance of VGFSS using MNB. Peak value achieved when the top 500 features were selected in the FFS.

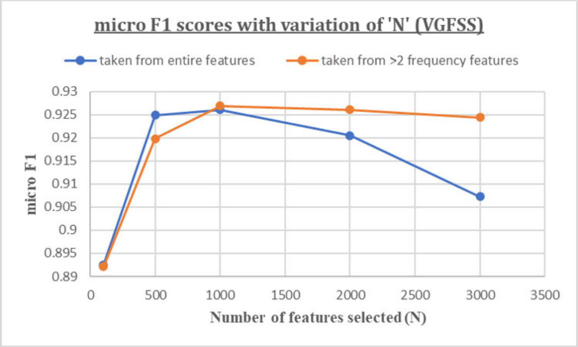

FIGURE 7. Performance of VGFSS using SVM. Peak value achieved when the top 1000 features were selected in the FFS.

discussed in Section III-C3. We employed SMOTE in two ways i.e. generating samples of entire features-based sentence representation and generating samples after alleviating sparsity and dimensionality. We termed the final feature set as **'high frequency features set (HFS)'** after applying the methodology of dynamic stop words filtering and VGFSS as described in Section III-C3 and Section III-C2 respectively, whereas, the default feature vector was named as **'entire features set (EFS)'**. The results of the models with incorporation of SMOTE samples are shown in Table 6. For both MNB and SVM, maximum value of micro F1 was achieved after training the classifiers on high frequency features set which emphasizes that addressing class skew problem along with dimensionality reduction is the key to bring about the maximum improvement in the performance of sentiment analysis classifier.

E. DISCUSSION OF RESULTS

The summary of results obtained after successively addressing the sparsity, dimensionality and class skew problem is shown in Table 8. Handling sparsity by removing the singleton terms in the final feature set resulted in a performance improvement of 3% for MNB, whereas, for SVM the improvement was only 1%. The reason is that removing low frequency terms helps in squeezing the size of input matrix representation of sentences which ensures better learning for the classifiers. However, the relatively small performance improvement for SVM can be associated with its optimization technique for maximal margin classification which allows it to learn from highly sparse representations as well. On the other hand, after using variable global feature selection scheme (VGFSS) to reduce the dimensionality and selecting the most relevant features, the micro F1 value further improved with a significant increment of nearly 30% for both MNB and SVM. This is due to the fact that VGFSS helps in variably selecting the most relevant features from each class in the data set which not only reduces the feature space but also enhances the classification ability of the machine learning classifiers. Finally, after applying SMOTE to handle the class skew and retraining the models, we got an improvement of nearly 1.5% for MNB and 6% for SVM. This difference in the performance improvement is due to the reason that SVM has steeper learning curve compared to MNB [43] which implies that the increase in the data set has more prominent effect for SVM. The overall results shown in Table 8 indicate that addressing dimensionality and class skew brings about maximum improvement in the performance of the sentiment classifier. Although, handling sparsity also boosts the results but the improvement is not as significant as in the case of addressing the other two problems. To explain the results further, some samples from the data set along with their actual labels and predicted labels by the SVM pipelines are shown in Table 7. Here 'P1', 'P2', 'P3' mean the predicted labels after resolving the sparsity, dimensionality and the class skew problems respectively. The actual class and the correct predictions have been bolded out in the table. All of these samples were wrongly classified by

TABLE 8. Performance improvement by successively addressing the problems.

Models		micro F1
Baseline	MNB	0.5614
	SVM	0.6261
After handling sparsity	MNB	0.5945
	SVM	0.6308
After handling dimensionality	MNB	0.9091
	SVM	0.9266
After handling class skew	MNB	**0.9241**
	SVM	**0.9852**

the baseline SVM model because of the problems discussed in this paper. If we look closer, the sample number 1, 2 and 6 consist of very few words. Before alleviating the sparsity, the vector representation of these samples would have been highly sparse. The information scarce in such vectors results in wrong predictions. Therefore, after alleviating the sparsity (P1), the predicted labels for samples 1 and 2 came out to be correct. However, sample 6 still got incorrectly classified because of the other two problems. Similarly, due to the existence of large number of redundant and common features in both classes, the classifier gets confused resulting in wrong predictions. After resolving the dimensionality problem (P2) by selecting the most relevant features, samples 3 and 4 came out to be correct. The last two samples got wrongly classified because of the existence of a large class skew in the original data set. This class imbalance causes the classifier to be somewhat biased resulting in incorrect predictions for the minority class. Therefore, after resolving the class skew by introducing the SMOTE samples, the last two samples of the minority class also got correctly predicted by the final classifier.

VI. CONCLUSION

In this study, we prepared a comprehensive data set by acquiring Urdu language tweets and getting them labeled from expert linguists on aspect and sentiment levels. There is no existing Urdu hate speech data set annotated on aspect and sentiment levels. We addressed the three most common problems faced in machine learning-based sentiment analysis namely, sparsity, dimensionality and class skew using state-of-the-art techniques and noted the performance improvement over the baseline model. Two machine learning algorithms i.e. SVM and Multinomial Naïve Bayes' were used for training the classifier. We used dynamic stop words filtering for alleviating sparsity, variable global feature selection scheme (VGFSS) for dimensionality reduction and for class imbalance, we used synthetic minority oversampling technique (SMOTE). For performance comparison with the baseline model, we used micro F1 measure. Our results showed that addressing class skew along with alleviating the high dimensionality problem brings about the maximum improvement in the overall performance of the sentiment analysis-based hate speech detection.

This study can be further pursued by acquiring more data from other social media sources and observe the results of the approaches presented in this paper. Another thing that can be done to handle the class skew problem is; the incorporation of lexical scores of the terms in the features set along with the TF-IDF weights. Furthermore, our future aim also includes addressing the class imbalance problem for deep learning algorithms as well.

REFERENCES

[1] A. S. Manek, P. D. Shenoy, M. C. Mohan, and K. R. Venugopal, "Aspect term extraction for sentiment analysis in large movie reviews using Gini index feature selection method and SVM classifier," *World Wide Web*, vol. 20, no. 2, pp. 135–154, Mar. 2017.

[2] M. T. Khan, M. Durrani, A. Ali, I. Inayat, S. Khalid, and K. H. Khan, "Sentiment analysis and the complex natural language," *Complex Adapt. Syst. Model.*, vol. 4, no. 1, pp. 1–19, Dec. 2016.

[3] W. Li, W. Shao, S. Ji, and E. Cambria, "BiERU: Bidirectional emotional recurrent unit for conversational sentiment analysis," 2020, *arXiv:2006.00492*. [Online]. Available: http://arxiv.org/abs/2006.00492

[4] U. H. Ehsan, S. Rauf, S. Hussain, and K. Javed, "Corpus of aspect-based sentiment for Urdu political data," in *Proc. 7th Int. Conf. Lang. Technol.*, Lahore, Pakistan, 2020.

[5] C. Gilbert and E. Hutto, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," in *Proc. 8th Int. Conf. Weblogs Social Media (ICWSM)*, vol. 81, 2014, p. 82. [Online]. Available: http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf

[6] Z. Mahmood, I. Safder, R. M. A. Nawab, F. Bukhari, R. Nawaz, A. S. Alfakeeh, N. R. Aljohani, and S.-U. Hassan, "Deep sentiments in roman urdu text using recurrent convolutional neural network model," *Inf. Process. Manage.*, vol. 57, no. 4, Jul. 2020, Art. no. 102233.

[7] A. K. Uysal, "An improved global feature selection scheme for text classification," *Expert Syst. Appl.*, vol. 43, pp. 82–92, Jan. 2016.

[8] 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.

[9] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *J. Mach. Learn. Res.*, vol. 3, pp. 1289–1305, Mar. 2003.

[10] M. Lan, C.-L. Tan, H.-B. Low, and S.-Y. Sung, "A comprehensive comparative study on term weighting schemes for text categorization with support vector machines," in *Proc. Special Interest Tracks Posters 14th Int. Conf. World Wide Web*, 2005, pp. 1032–1033.

[11] W. Zhang, T. Yoshida, and X. Tang, "A comparative study of TF*IDF, LSI and multi-words for text classification," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2758–2765, Mar. 2011.

[12] H. Saif, Y. He, and H. Alani, "Alleviating data sparsity for Twitter sentiment analysis," in *Proc. CEUR Workshop*, 2012, pp. 1–9.

[13] S. S. R. Mengle and N. Goharian, "Ambiguity measure feature-selection algorithm," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 60, no. 5, pp. 1037–1050, May 2009.

[14] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Nashville, TN, USA, 1997, vol. 97, nos. 412–420, p. 35.

[15] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, and Z. Wang, "A novel feature selection algorithm for text categorization," *Expert Syst. Appl.*, vol. 33, no. 1, pp. 1–5, Jul. 2007.

[16] A. K. Uysal and S. Gunal, "A novel probabilistic feature selection method for text classification," *Knowl.-Based Syst.*, vol. 36, pp. 226–235, Dec. 2012.

[17] M. Stager, P. Lukowicz, and G. Troster, "Dealing with class skew in context recognition," in *Proc. 26th IEEE Int. Conf. Distrib. Comput. Syst. Workshops (ICDCSW)*, Jul. 2006, p. 58.

[18] A. Liu, J. Ghosh, and C. E. Martin, "Generative oversampling for mining imbalanced datasets," in *Proc. DMIN*, 2007, pp. 66–72.

[19] Q. Kang, X. Chen, S. Li, and M. Zhou, "A noise-filtered under-sampling scheme for imbalanced classification," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4263–4274, Dec. 2017.

[20] F. Ali, A. Ali, M. Imran, R. A. Naqvi, M. H. Siddiqi, and K.-S. Kwak, "Traffic accident detection and condition analysis based on social networking data," *Accident Anal. Prevention*, vol. 151, Mar. 2021, Art. no. 105973. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S000145752100004X

[21] F. Ali, S. El-Sappagh, S. M. R. Islam, A. Ali, M. Attique, M. Imran, and K.-S. Kwak, "An intelligent healthcare monitoring framework using wearable sensors and social networking data," *Future Gener. Comput. Syst.*, vol. 114, pp. 23–43, Jan. 2021.

[22] M. Z. Ali, K. Javed, E. U. Haq, and A. Tariq, "Sentiment and emotion classification of epidemic related bilingual data from social media," 2021, *arXiv:2105.01468*. [Online]. Available: http://arxiv.org/abs/2105.01468

[23] A. Al-Laith, M. Shahbaz, H. F. Alaskar, and A. Rehmat, "AraSenCorpus: A semi-supervised approach for sentiment annotation of a large arabic text corpus," *Appl. Sci.*, vol. 11, no. 5, p. 2434, Mar. 2021.

[24] H. Saif, M. Fernández, Y. He, and H. Alani, "On stopwords, filtering and data sparsity for sentiment analysis of Twitter," in *Proc. 9th Int. Conf. Lang. Resour. Eval. (LREC)*, 2014, pp. 810–817.

[25] N. Yadav, O. Kudale, A. Rao, S. Gupta, and A. Shitole, ''Twitter sentiment analysis using supervised machine learning,'' in *Intelligent Data Communication Technologies and Internet of Things*. Singapore: Springer, 2021, pp. 631–642.

[26] N. Anand, D. Goyal, and T. Kumar, ''Analyzing and preprocessing the Twitter data for opinion mining,'' in *Proc. Int. Conf. Recent Advancement Comput. Commun.* Singapore: Springer, 2018, pp. 213–221.

[27] D. Agnihotri, K. Verma, and P. Tripathi, ''Variable global feature selection scheme for automatic classification of text documents,'' *Expert Syst. Appl.*, vol. 81, pp. 268–281, Sep. 2017.

[28] M. S. Ali and K. Javed, ''A novel inherent distinguishing feature selector for highly skewed text document classification,'' *Arabian J. Sci. Eng.*, vol. 45, no. 12, pp. 10471–10491, Dec. 2020.

[29] A. Moreo, A. Esuli, and F. Sebastiani, ''Distributional random oversampling for imbalanced text classification,'' in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2016, pp. 805–808.

[30] A. Anand, G. Pugalenthi, G. B. Fogel, and P. N. Suganthan, ''An approach for classification of highly imbalanced data using weighting and undersampling,'' *Amino Acids*, vol. 39, no. 5, pp. 1385–1391, Nov. 2010.

[31] F. S. Al-Anzi and D. AbuZeina, ''Beyond vector space model for hierarchical arabic text classification: A Markov chain approach,'' *Inf. Process. Manage.*, vol. 54, no. 1, pp. 105–115, Jan. 2018.

[32] J. G. Kemeny and J. L. Snell, *Markov Chains*. New York, NY, USA: Springer-Verlag, 1976.

[33] A. S. More and D. P. Rana, ''Review of random forest classification techniques to resolve data imbalance,'' in *Proc. 1st Int. Conf. Intell. Syst. Inf. Manage. (ICISIM)*, Oct. 2017, pp. 72–78.

[34] L. Qing, W. Linhong, and D. Xuehai, ''A novel neural network-based method for medical text classification,'' *Future Internet*, vol. 11, no. 12, p. 255, Dec. 2019.

[35] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, ''ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis,'' *Future Gener. Comput. Syst.*, vol. 115, pp. 279–294, Feb. 2021.

[36] Y. LeCun, Y. Bengio, and G. Hinton, ''Deep learning,'' *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[37] K. Makice, *Twitter API: Up and Running: Learn How to Build Applications With the Twitter API*. Sebastopol, CA, USA: O'Reilly Media, 2009.

[38] A. J. Viera and J. M. Garrett, ''Understanding interobserver agreement: The kappa statistic,'' *Family Med.*, vol. 37, no. 5, pp. 360–363, 2005.

[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, ''Scikit-learn: Machine learning in Python,'' *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[40] D. D. Lewis, ''Naive (Bayes) at forty: The independence assumption in information retrieval,'' in *Proc. Eur. Conf. Mach. Learn.* Berlin, Germany: Springer, 1998, pp. 4–15.

[41] C. Cortes and V. Vapnik, ''Support-vector networks,'' *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[42] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, ''SMOTE: Synthetic minority over-sampling technique,'' *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[43] M. Sharma and M. Bilgic, ''Learning with rationales for document classification,'' *Mach. Learn.*, vol. 107, no. 5, pp. 797–824, May 2018.

MUHAMMAD Z. ALI received the B.Sc. and M.Sc. degrees in electrical engineering from the University of Engineering and Technology, Lahore, Pakistan, in 2016 and 2020, respectively.

He has been working as a Research Officer with the Centre for Language Engineering Laboratory, KICS, UET, since July 2020. Before joining CLE, he worked as an Assistant Manager at Pakistan Telecommunication Company, from 2016 to 2020, where his main responsibilities include data analysis, reporting, and network health analysis. His research interests include machine learning, data science, computational social science, and natural language processing.

EHSAN-UL-HAQ received the B.Sc. and M.Sc. degrees in computer science from the University of Engineering and Technology, Lahore, Pakistan, in 2009 and 2013, respectively, where he is currently pursuing the Ph.D. degree in computer science. He is currently working as the Manager Research at the Center for Language Engineering, KICS, UET. While working at CLE, he has worked on the development of Urdu screen reader, Urdu text-to-speech systems, and Urdu word sense disambiguation (WSD) systems. His research interests include machine learning, data mining, and natural language processing.

SAHAR RAUF received the master's degree in English literature and linguistics from the National University of Modern Languages, Lahore, Pakistan, in 2013, and the M.Phil. degree in applied linguistics from Kinnaird College for Women University, Lahore, in 2015.

She has been working as a Senior Research Officer with the Center for Language Engineering, Lahore, since 2015. In 2019, she worked as a Visiting Lecturer with the University of the Punjab, Lahore, and taught acoustic phonetics course at the master's level. Her research interests include developing resources for local languages generally and developing Urdu corpora specifically.

KASHIF JAVED received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from UET, Lahore, Pakistan, in 2000, 2004, and 2012, respectively. He currently serves as an Associate Professor with UET. He has published a number of articles in reputed international journals hosted by the IEEE, PLoS, Springer, and Elsevier. He is currently acting as a reviewer for many journals published by the IEEE, Elsevier, and Springer. His research interests include machine learning, text classification, and natural language processing.

SARMAD HUSSAIN received the B.Sc. degree in electrical engineering from The University of Texas at Austin, Austin, USA, in 1992, the M.Phil. degree in computer speech and language processing from the University of Cambridge, Cambridge, U.K., in 1993, and the Ph.D. degree in speech science from Northwestern University, Evanston, USA.

He is currently a Professor of computer science and the Head of the Center for Language Engineering, Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology, Lahore, Pakistan. His research interests include developing computing solutions for Pakistani languages, including research in linguistics, localization, language computing standards, speech processing, and computational linguistics. He was a recipient of the Pride of Performance Award, conferred by the Government of Pakistan, in 2018, for his distinguished contributions in the area of local language computing.