IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 August 2022, accepted 15 September 2022, date of publication 20 September 2022, date of current version 30 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3208164

RESEARCH ARTICLE

Deep Learning Based Cross Domain Sentiment Classification for Urdu Language

AMNA ALTAF[1,2], MUHAMMAD WAQAS ANWAR[1], MUHAMMAD HASAN JAMAL[1], SANA HASSAN[1], USAMA IJAZ BAJWA[1], GYU SANG CHOI[3], AND IMRAN ASHRAF[3]

[1]Department of Computer Science, COMSATS University Islamabad–Lahore Campus, Lahore 54000, Pakistan
[2]Department of Information System, Dr. Hasan Murad School of Management, University of Management and Technology, Lahore 54770, Pakistan
[3]Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, South Korea

Corresponding authors: Gyu Sang Choi (castchoi@ynu.ac.kr) and Imran Ashraf (ashrafimran@live.com)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) by the Ministry of Education under Grant NRF 2021R1A6A1A03039493.

ABSTRACT Sentiment analysis is a widely researched area due to its various applications in customer services, brand monitoring, and market research. Automatic sentiment classification is an important but challenging task. Contrary to the English language, sentiment analysis for low-resource languages like Urdu is an under-explored research area. Most of the work on sentiment analysis in the Urdu language is domain-dependent where models are mostly trained and tested on the same dataset on limited domains. However, sentiments in different domains are expressed differently, and manually annotating the datasets for all possible domains is unfeasible. Training a sentiment classifier using annotated data on one domain and testing it on another domain results in poor performance as the terms appearing in the source domain (training data) might not appear in the target (testing data) domain. In this paper, we present a baseline method for cross-domain sentiment analysis in the Urdu language using two different domains. Feature extraction is performed using n-grams and word embedding techniques. Sentiment classification is performed using machine learning and deep learning classifiers. The proposed method achieves an accuracy, precision, recall, and F1 scores of 0.77, 0.83, 0.68, and 0.75, respectively.

INDEX TERMS Cross-domain sentiment analysis, deep learning, urdu language processing, feature engineering.

I. INTRODUCTION

In this technology-driven era, online social networks (OSNs) such as Twitter and Facebook are actively involved in enabling global connectivity. Users freely consume and generate information that leads towards precedent amounts of data [1]. Due to the explosion of this data, the internet has become a huge dynamic repository of public views on a large variety of topics or genres (movie reviews, sports reviews, electronic reviews, etc.) [2]. Sentiment classification has become a key enabler of opinion summarization and extraction that automatically categorizes the sentiment in a piece of text on any topic or entity [3]. Some examples of such content include merchandise buyers, product reviews, hotel customers, etc. The emotional tendency exhibited by categorizing sentiment polarity can be a helpful indicator of consumer behavior and opinions leading to improved efficiency in information sharing among users and improved business services and solutions [4].

Sentiment analysis (SA) is performed at different levels i.e., document level, sentence level, and aspect level. In the document level SA, the whole document is considered as a basic unit, discussing a single topic. The whole document is considered positive if there are more positive sentences than negative sentences and vice versa. Whereas sentence level SA categorizes the sentiment in each sentence as positive, negative, or neutral. The aspect level SA is a more fine-grained analysis that classifies the sentiments based on the aspects that are already identified [5].

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

For reviews of a particular domain, the annotated instances of that domain can be trained to build a standard machine learning (ML) classifier. The process of annotation refers to assigning each instance with a sentiment polarity label. The trained classifier can predict the polarity of new reviews of that domain [6] reasonably well depending on the quality and availability of the labeled data. However, this process of manually annotating the content is unfeasible.

As per existing research studies, SA can be categorized as multi-domain SA [5], [7], [8], Cross Domain Sentiment Analysis (CDSA) [9], [10], bilingual SA [11], [12], and multilingual SA [13], [14]. In multi-domain SA, the dataset is collected from multiple genres, and training and testing of models are performed on the same dataset. In CDSA, the dataset from one domain, generally having a large amount of labeled data, is used to train the classifier, which can then be used to predict the sentiments of a different but related domain, generally having little available labeled data, hence minimizing the effort of manually annotating the dataset. Thus, there has been growing interest in exploring effective ways to reuse labeled reviews across different domains. Here, a domain is referred to as a collection of reviews that belong to a particular product [15]. In bilingual and multilingual SA, the sentiment classification is performed using a dataset comprising two and more languages respectively.

At present, the main methods for sentiment classification include lexicon-based methods, ML methods, and combined methods. ML approach uses traditional methods such as Naïve Bayes (NB) and deep learning (DL) methods. The lexicon-based methods depend on the construction of a sentiment lexicon that is developed by selecting appropriate sentiment words. The combined methods use both lexicon-based and ML approaches [16].

Being an active area of research, researchers have proposed many methods for SA. However, these methods have experimented with resource-rich languages like English and Chinese. In Natural Language Processing (NLP), a language is known as a resource-rich language, if there are enough resources, i.e., corpus and lexicons, for the NLP research community whereas the languages with limited resources are known as low-resource languages [17]. Considering CDSA, there are very few studies on resource-rich as well as low-resource languages like Urdu, Hindi, and Bengali. 100 million people around the world speak Urdu language [5]. Urdu is the national language of Pakistan and is frequently spoken medium in numerous states of India. In OSNs, many native Urdu speakers use different platforms like YouTube, Facebook, and Twitter to express their opinions, emotions, and feelings using Urdu script and Roman Urdu (Latin script). Consequently, it is crucial to perform SA of Urdu script to grasp the feelings, emotions, and opinions of native Urdu speakers [7]. Existing works on Urdu SA [18], [19], [20], [21] rely on the availability and quality of labeled data. Many researchers performed SA in Urdu by annotating the dataset on a single domain or multiple domains. However, manual labeling of explosively growing data is very impractical. To the best of our knowledge, no work has been done on CDSA for the Urdu language, using Urdu script. So, there is a need to develop such models that are flexible enough to train on one domain and predict the sentiments in another domain, minimizing the time and effort in manually annotating the dataset. To overcome this problem, the main aim of this study is to develop models that can adapt across domains and minimize the manual labeling effort. This paper presents a baseline method for CDSA in the Urdu language and performs classification tasks using ML and DL methods. The main contributions of this paper are summarized as follows:

- Development of annotated datasets for two different but related domains i.e., cricket and football. A total of 9221 tweets are collected from Twitter in this regard. Out of these tweets, 6221 belong to the cricket domain while the remaining 3000 tweets belong to the football domain.
- Extraction of n-gram features at both the word level and character level. Moreover, for DL methods, word embedding is generated using the one-hot encoder.
- Use of five ML classifiers, Bernoulli Naïve Bayes (BNB), multinomial Naive Bayes (MNB), logistic regression (LR), random forest (RF), and linear support vector classifier (SVC) for the task of CDSA. Moreover, three DL models i.e., recurrent neural network (RNN), long short term memory (LSTM), and gated recurrent unit (GRU) are also used for the classification of sentiments.
- To the best of our knowledge, no study exists on CDSA for the Urdu language. Therefore, this study presents a baseline method for CDSA and evaluates the proposed method using four evaluation measures i.e., accuracy, precision, recall, and F1-score. A DL model is also developed that is adaptable to handle other domains.
- Use of two existing, widely used, standard Urdu datasets to validate the proposed predictive model.

The rest of the paper is organized as follows. Section II presents the literature survey followed by Section III that explains the research methodology of this study. Section IV presents the experimental setup and explains the results and discussion. Section V concludes the research.

II. RELATED WORK

Research in the field of SA has received increasing attraction lately and many studies have been conducted in this field. The following section discusses different studies on SA using the lexicon-based approach, ML approach, and DL approach.

A. LEXICON-BASED APPROACH

Mukhtar *et al.* [22] perform SA in Urdu language using a lexicon-based approach by enhancing an existing SA lexicon and introducing context-dependent words in it. These words are used with or without conjunctions. Moreover, the authors develop rules to assign sentiments to these context-dependent words and these rules are further

combined with a sentiment analyzer. The results show a significant performance improvement of the Urdu sentiment analyzer from 83% to 89%. Hossein *et al.* [23] introduce a lexicon-based method for SA in the Persian language using a dataset of mobile reviews. The authors extract the aspects from the reviews using the combination of 'noun adjective' pair or 'nouns adverbs adjective' pair using a lexicon. They also consider the impact of intensifiers on the reviews and present a visual summary of aspects in the reviews. The proposed method outperforms the previous studies in terms of accuracy. Aye and Aung [24] perform SA on the Myanmar language using a lexicon-based approach using a dataset of 500 restaurant reviews collected from OSNs. The authors develop a lexicon using a dictionary-based approach. They identify sentiment targets and assign polarity to the respective sentiments of the targets. The identification of aspect terms is context-independent. The performance of the automatic polarity extraction method is compared with manually annotated reviews. Results of the proposed technique show high accuracy.

Alqaryouti *et al.* [25] perform aspect-based SA using lexicon and rule-based approaches in the Arabic language. The authors discuss the aspect categories based on the standards provided by mobile companies. They extract implicit and explicit aspects from the lexicons, match the opinion words and aspects with the lexicon to find the category of the aspects, and assign the polarity of the opinion words using lexicons based on devised rules. The proposed method achieves an accuracy of 93%. Ibrahim *et al.* [26] develop a lexicon to perform the SA of idioms in the Arabic language. The authors collect data manually through APIs consisting of proverbs, idioms, phrases, etc., and annotate it through different annotators. The developed lexicon consists of four columns, i.e., proverbs/idioms, English translation of proverbs, Buck Walter, and polarity. They detect the proverbs using bi-grams to six-grams and similarity between two texts is determined using cosine similarity and the Levenshtein distance algorithm. In the extraction phase, heuristic rules are used to classify sentiments to avoid redundancy. To detect polarity, positive proverbs are replaced by positive phrases, and negative proverbs are replaced by negative phrases. The results show an accuracy of 81.60% when the n-gram model is used with cosine similarity, an accuracy of 86.12% when n-gram is used with the edit distance algorithm, and an accuracy of 98.62% when n-gram, cosine similarity, and edit distance are used collectively.

B. MACHINE LEARNING APPROACH

Mehmood *et al.* [18] propose a novel feature spamming approach to assign weights to terms in Roman Urdu which helps to extract the most relevant and useful information from data. Firstly, they identify important features using term utility criteria (used to assign higher scores to significant topics and vice versa). Furthermore, these distinctive features are spammed using spamming factor which is adjusted by user-defined hyperparameters while the weights of all other features are not changed. Different ML classifiers are used for evaluation and results are compared to the state-of-the-art features weighting schemes. The feature spamming approach produces the best results as compared to the state-of-the-art schemes. Mehmood *et al.* [27] develop a publicly available dataset for Roman Urdu comprising 11000 reviews from six different domains. Moreover, the authors extract the features from the dataset based on word grams, character grams, and the union of both. At the character level, experiments are performed including and excluding word boundaries, respectively. Additionally, they also extract feature unions based on best-performing features in the character-gram and word grams. To improve the performance of the system, they select the three best ML classifiers to form an ensemble classifier that uses voting and weighted voting techniques. Furthermore, they apply LSTM and CNN DL classifiers over the entire dataset to further enhance the system performance. T-tests are applied to show the statistical significance of the proposed approach.

Noor *et al.* [21] perform SA in Roman Urdu for automobile reviews by extracting features from the data using the bag of words model and then assigning Term frequency-inverse document frequency (TF-IDF) weights to these features. For experimentation purposes, SVM with linear kernel and the cubic kernel is used. Moreover, one-vs-all and one-vs-one techniques are used to perform ternary classification. SVM cubic kernel outperforms linear kernel in multi-class classification. Bibi *et al.* [28] propose a technique to perform SA in the Urdu language using tweets. Features are extracted from the data using POS tags i.e., adjectives and count of positive and negative words in a sentence. Moreover, the proposed methodology is evaluated using 10-fold cross-validation. The decision table is applied to extracted features achieving an accuracy of 90%. Although the proposed method produces good results, the size of the corpus is very small and some other important POS tags like nouns, verbs, etc. are not considered during the feature extraction process.

Khan and Malik [29] perform binary sentiment classification on automobile reviews in Roman Urdu by extracting features at the word level from the dataset using the StringToWordVector filter. Moreover, different ML and DL algorithms are applied to check the performance of the system. These algorithms include Multi-Layer Perceptron (MLP), DT, RF, MNB, bagging classifier, AdaBoost, SVM, KNN, etc. MNB outperforms all the classifiers in terms of precision, recall, and F-measure. Although the achieved results are good, the proposed system is limited to handle only positive and negative reviews and the corpus standardization is also a major issue. Mukhtar *et al.* [30] perform a statistical evaluation of classifiers in Urdu SA. Naïve Bayes multinomial text, IBK, KNN, LibSVM, J48, and PART classifiers are trained on the dataset to check the performance of the best classifier and IBK performs best as compared to all other classifiers. The best three out of the five classifiers are selected for statistical comparison. Furthermore, Kappa statistics, McNemar's test, and root mean square error, are used to check the

effectiveness of classifiers. Kappa statistics are performed to check interrater annotatability. McNemar's test is applied to check the marginal frequencies of the rows and columns. Root mean square error is used to identify the error between the tested class and the predicted class. For these statistical tests, IBK still outperforms the other classifiers.

C. DEEP LEARNING APPROACH

Khan *et al.* [19] perform SA on Urdu language using a dataset comprising multiple domains including beverages, movies sports, politics, etc. Rule-based, ML, and DL approaches are used for the classification of the text. Moreover, multilingual BERT (mBERT) is also fine-tuned for the multi-class classification. The proposed study uses four text representations i.e., char-gram, n-gram, BERT word embedding, and fastText for the training of models. mBERT outperforms all methods with an F1-score of 81%. Khan *et al.* [31] perform SA for English and Roman Urdu using the DL method. The authors evaluate the performance of different word embedding and a DL model architecture is proposed that comprises two layers. LSTM is used to capture the long-term dependency of the words and the CNN layer is used for the feature extraction. For categorizing the sentiments of the text, the features learned by CNN and LSTM are passed to different ML classifiers. Experiments are performed on four different datasets using the proposed CNN-LSTM which outperforms the state-of-the-art methods with a 5% increase in accuracy.

Chandio *et al.* [20] propose a recurrent DL architecture RU-BiLSTM for Roman Urdu SA to handle the high dimensionality and sparsity of textual data. This recurrent architecture is coupled with an attention mechanism and word embedding. Moreover, the bidirectional LSTM is used to retain the context of the text in both directions and the attention mechanism assists to identify on the most important features. Next, the binary and ternary classification is obtained by a dense softmax output layer. The proposed architecture outperforms the baseline methods with improved accuracy of 8%. Naqvi *et al.* [32] propose a framework for Urdu SA using DL models combined with different word representations. The performance of LSTM, CNN, and CNN-LSTM DL models are evaluated, and additionally, stacked layers are used in sequential C-LSTM, LSTM, and BiLSTM models by applying different filters to the convolutional layer. Furthermore, unsupervised self-trained and pre-trained word embedding is used for the SA task. BiLSTM–ATT performs well as compared to other DL models with 77% accuracy and 72% F1-score. Khan *et al.* [7] perform SA in Urdu language using the DL approach to develop a dataset for Urdu language and evaluate the performance of various DL models. A comparison is established to compare the two modes of text representation i.e., count-based and fastText word embeddings for Urdu. Both ML and DL models are used to perform experiments for all features and n-gram features with LR perform best for the SA task with an F1 score of 82%.

Dashtipour *et al.* [33] propose a SA framework in the Persian language for hotel reviews that detects the polarity of the sentence using linguistic rules and DL models. Upon pattern detection, this method allows polarity to flow from words to concepts based on the symbolic dependency relations. Furthermore, when no pattern is detected, this method uses its sub-symbolic counterpart and uses DL for sentiment classification. The proposed method achieves up to 15% higher accuracy than the baseline methods. Li *et al.* [34] propose a bidirectional LSTM with a self-attention mechanism and multi-channel features (SAMF-BiLSTM). This approach is comprised of two parts i.e., multi-channel features and a self-attention mechanism. In the first phase, existing sentiment resources and linguistic knowledge are modeled, and various features are extracted as input to the model. Then, BiLSTM is used to extract the information regarding sentiments. Additionally, the BiLSTM-D model is also developed for document-level SA. The proposed method performed better than the baseline methods.

A comprehensive literature survey highlights that most of the works on Urdu language SA are domain-dependent where the annotators annotate the datasets for multiple domains. However, annotating datasets for different domains is a tedious as well as time-consuming process. This study applies the DL model for CDSA that is flexible enough to adapt to a new domain without annotation. To the best of our knowledge, no work has been reported on the CDSA problem for the Urdu language.

III. METHODOLOGY

This section discusses the methodology adopted to solve the problem of CDSA. The architecture of the proposed methodology is illustrated in Figure 1.

A. DATA COLLECTION AND PREPROCESSING

To have a cross-domain dataset for different but related domains, we develop a dataset for two domains i.e., cricket and football, using the Twitter intelligence tool (TWINT) library. The dataset comprises a total of 9221 sentences out of which 6221 belong to the cricket domain while the remaining 3000 belong to the football domain. To evaluate the performance of our approach, two datasets are used, i.e., balanced, and unbalanced. The balanced dataset comprises an equal number of positive, negative, and neutral sentences. The statistics for both balanced and unbalanced datasets are shown in Table 1. The keywords used to search cricket and football tweets are illustrated in Table 2.

To preprocess the dataset for annotation, (i) special characters like @, #, $, !, (ii) hyperlinks, (iii) emoticons, (iv) unwanted characters, (v) extra spaces, are removed. Additional information like username, location, language, sources, etc. that was collected at the time of data crawling is also removed.

B. DATA ANNOTATION

The dataset is annotated with the help of three annotators who are native speakers of the Urdu language. Firstly,

TABLE 1. Statistics of the collected dataset.

Domains	Dataset	Unique Tokens	Average Tokens	Total Tokens	Positive Tweets	Negative Tweets	Neutral Tweets	Total Tweets
Cricket	Balanced	7111	16.6	7111	1424	1424	1424	4272
	Unbalanced	102058	16.4	102058	2309	2488	1424	6221
Football	Balanced	6215	16.6	42375	848	848	848	2544
	Unbalanced	6752	16.5	49590	1278	848	874	3000

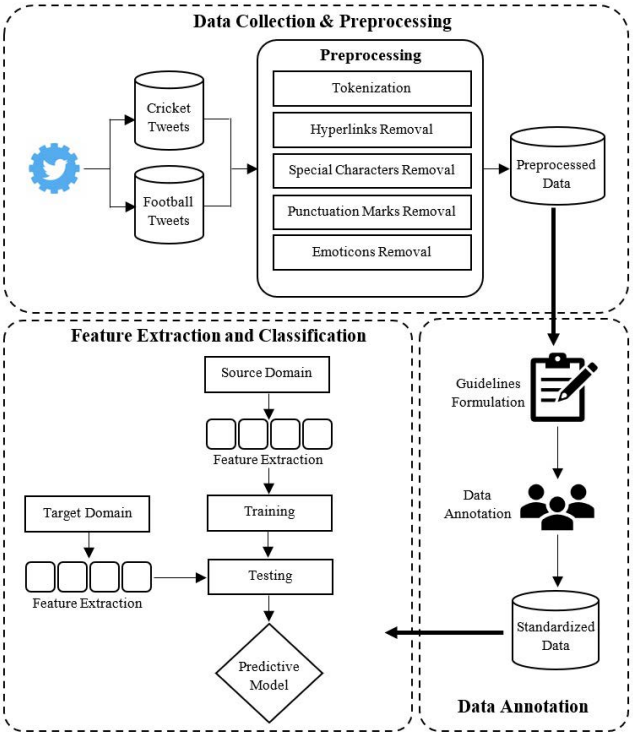


FIGURE 1. Proposed methodology for cross-domain sentiment analysis.

TABLE 2. Search words.

Sr. No	Urdu Keywords	English Translation
1	شاہد آفریدی	Shahid Afridi
2	شعیب اختر	Shoaib Akhtar
3	وسیم اکرم	Waseem Akram
4	کرکٹ	Cricket
5	ٹی ٹوئنٹی	T-Twenty
6	فٹ بال	Football
7	فٹ بال ورلڈ کپ	Football World Cup
8	گول	Goal
9	میسی	Messy
10	رونالڈو	Ronaldo

annotation guidelines are formulated with the mutual consensus of the annotators. Next, the same dataset is annotated by two annotators. In case of a conflict among these annotators, the expertise of the third annotator is acquired to solve the conflict. The annotation guidelines used to annotate the dataset are discussed in the following subsection.

1) ANNOTATION GUIDELINES

The comprehensive guidelines are formulated for data annotation. Following guidelines are used to annotate the dataset.

Guidelines for positive polarity: A sentence is assigned with positive polarity if it has a greater number of positive sentiments as compared to negative sentiments; for instance:

ظفر گوہر سپیشلسٹ بولر ہے اور بہت اچھا بیٹسمین۔

Zafar Gohar is a specialist bowler and a good batsman.

Guidelines for negative polarity: A sentence is assigned with negative polarity if it has a greater number of negative sentiments as compared to positive sentiments; for instance:

جی مجھے سب پتہ سرفراز کی بیٹنگ پر کسی کو بھروسہ نہیں تھا

Yes, I know, no one had trust in Sarfraz's batting.

Guidelines for neutral polarity: A sentence is assigned with neutral polarity if it has equal numbers of positive and negative sentiment words present in a sentence or if there is no sentiment word present in a sentence. For instance:

یونس خان دورہ انگلینڈ کے لئے پاکستان کے بیٹنگ کوچ مقرر۔

Younas Khan is appointed as a batting coach for England tour.

Interrogative and sarcastic sentences: In the collected dataset, interrogative and sarcastic sentences are assigned with a negative label. The main reason for assigning negative polarity to these sentences is that in the case of interrogative sentences there exists uncertainty about a certain event that needs to be inquired about. Similarly, sarcastic sentences are used to mock someone or show the exact opposite meaning of what is to be said by anyone.

کیا مصباح کے ہوتے ہوئے بیٹنگ کوچ کی واقعی ضرورت تھی؟

Is the batting coach really required in the presence of Misbah?

اور فیشن شو کرنے کے لئے کس کو ٹیسٹ ٹیم میں لے جایا گیا ہے؟

And who has been taken in the test team to do a fashion show?

Sentences with conjunctions: Sentences with conjunctions have two clauses i.e., subsequent clause (present before conjunction) and consequent clause (present after conjunction). The overall sentiment of the sentence is assigned based on the sentiment of the consequent clause. For example, in the following example جبکہ (while) is a conjunction, and the

sentiment this conjunction shows is positive so the overall polarity of this sentence will be positive.

انڈر19 ورلڈ کپ میں آسٹریلیا اور بھارت 3،3 جبکہ پاکستان 2 مرتبہ ٹائٹل اپنے نام کر چکا ہے۔

In the Under-19 world cup, Australia and India have won the title three times while Pakistan has won the title twice.

Varying perspective: Some sentences convey negative opinions in the context of the author and positive opinions in the context of the reader and vice versa. For instance:

کوالیفائنگ میچ میں مصر نے کانگو کو 2-1 سے شکست دی۔

Egypt beat Congo by 2-1 in the qualifying match.

The above-mentioned example is negative for the supporters of Congo. However, it is positive for the supporters of Egypt. In this study, these sentences are annotated according to the viewpoint of readers.

2) COHEN's KAPPA STATISTICS

The agreement between the annotators is measured using inter-rater reliability which is defined as the range up to which the two annotators assign the same score to a variable [30] and is computed using the following equations

$$Kappa = \frac{P_o - -P_e}{1 - P_e} \tag{1}$$

$$P_o = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{2}$$

$$P_e = \frac{\left[\begin{array}{l}(T_N + F_P) * (T_N + F_N) \\ +(T_P + F_P) * (T_P + F_N)\end{array}\right]}{(T_P + T_N + F_P + F_N)^2} \tag{3}$$

where P_o is the relative observed agreement among annotators and P_e is the hypothetical probability of chance agreement. T_P, T_N, F_P, and F_N are true positive (T_P), true negative (T_N), false positive (F_P) and false negative (F_N), respectively. The value of Kappa statistics for our dataset is 0.92 which indicates a high agreement between the two annotators.

C. FEATURE EXTRACTION AND CLASSIFICATION

Feature extraction is a crucial step for applying ML classifiers. In this study, features are extracted for both ML classifiers and DL models. For ML classifiers, feature extraction is performed using a TF-IDF vectorizer. In information retrieval, TF describes the number of times a term appears in the dataset and IDF explains the importance of the word in the given document. Given a dataset D, word w, and document records $d \in D$, the equation to compute TF-IDF is given below [35]

$$w_d = f(w, d) \times log \frac{|D|}{f(w, D)} \tag{4}$$

where $f(w, d)$ is the number of times a word appears in the dataset, and w_d shows the importance of the given term. For this study, uni-grams, bi-grams, and tri-gram features are extracted. Some combination of character-gram features is also combined with the word gram features to perform the classification task.

For DL models, numeric input is considered for the classification tasks, so as a part of feature engineering, the tweets are transformed into one-hot vectors. It is a $1 \times N$ vector consisting of 0's in all the cells of a vector except in a cell that is used to uniquely identify a word in a document that has a value of 1. For encoding, each token of the tweet is separately encoded and then padded to make sure that all vectors are of the same length [36].

For the classification of sentiments, several classical ML models i.e., MNB, BNB, LR, RF, and linear SVC and RNN, LSTM, and GRU DL models are used. The details of these classifiers and models are discussed in the following subsections.

1) MULTINOMIAL Naïve BAYES

A term can be pivotal in determining the polarity of a document that makes the MNB model a good choice for the classification task. To predict the sentiment of a new document, the probabilities of occurrence of all the words are multiplied against positive and negative sentiments and the higher probability value is used to assign the final polarity. A new document n with polarity p can be determined as [37]

$$P(p|n) \propto P(p) \prod_{(1<k\leq nd)} P(t_k|p) \tag{5}$$

where $P(t_k|p)$ shows the conditional probability whether a term t_k appears in a new document whose polarity can be determined as follows

$$P(t_k|p) = \frac{count(t_k|p) + 1}{count(t_p) + |V|} \tag{6}$$

2) BERNOULLI Naïve BAYES

In the BNB classifier, features are presented as independent binary variables which shows whether a term present in the document can be considered or not. This classifier is similar to MNB with the only difference that MNB inquires about TF whereas BNB determines whether a term is present or absent in the document under consideration. A new document n with polarity p can be determined as [37]

$$P(p|n) \propto P(p) \prod_{(1<k\leq nd)} P(t_k|p)(1 - P(t'_k|p)) \tag{7}$$

where $P(t_k|p)$ shows the conditional probability whether a term t_k appears in new document of polarity p and $P(t'_k|p)$ represents conditional probability of non-occurring term t_k in new document of polarity p and can be determined as

$$P(t_k|p) = \frac{count(t_k|p) + 1}{count(N_p) + 2} \tag{8}$$

$$P(t'_k|p) = \frac{count(t'_k|p) + 1}{count(N_p) + 2} \tag{9}$$

where N_p is the total number of documents having a polarity p.

3) LINEAR SUPPORT VECTOR CLASSIFIER

Linear SVC works on the principle of Structural Risk Minimization (SRM) with the focus of finding a hyperplane that segregates two classes in the input space $x^k = f^k$, and w, b which are determined by minimizing the loss function. Its final equation is given as [38]

$$L(w, b) = w^t w + c \sum max(0, 1 - y^i(w^t F^{(i)} + b))^2 \quad (10)$$

4) LOGISTIC REGRESSION

The LR classifier takes the variables' vector and determines the coefficient for the input expression and then finds the class of the text as a word vector. The LR function finds multiple linear functions exhibited as

$$Logit(P) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \cdots + \beta_k X_k \quad (11)$$

where P shows the probability of the occurrence of the feature. $X_1, X_2 \cdots X_k$ shows the value of predictor and $\beta_1, \beta_2 \cdots \beta_k$ shows the model's intercept [39].

5) RANDOM FORREST

RF links the arbitrary subspace ideas with bagging. It relies on different decision trees powered on moderately distinct information sub-sets. It is an ensemble technique based on bagging. Moreover, it is capable of categorizing massive information. This classifier has low bias and high variability which is why it can learn irregular patterns and fit in with its training [40].

6) RECURRENT NEURAL NETWORK

RNN is a DL model and is mostly used for classification. It assigns weights to the sequence of previous data points. RNN performs better for semantic analysis of data as it considers the information of previous nodes. It usually contains three layers i.e., the input layer, the hidden layer, and an output layer which can be formulated as

$$x_t = F(x_{t-1}, u_t, \theta) \quad (12)$$

where x_t represents the state at time t, and u_t states input at step t. These weights can be used to form an equation parameterized by

$$x_t = W_{rec}\sigma(x_{(t-1)}) + W_{in}u_t + b \quad (13)$$

where W_{rec} represents recurrent matrix weight, W_{in} shows input weights, b refers to bias, and σ shows element-wise function [41].

7) LONG SHORT TERM MEMORY

In contrast to RNN, LSTM is more powerful due to its capability to resolve the problem of vanishing gradient [42]. It uses an activation function to find-long term dependencies and diminishes the problem of vanishing gradient with the usage of three gates. Besides the input state, it has a call state and a hidden state. Additionally, the output probability and the activation function are computed at each time stamp. It computes the input, output, and forget gates using the sigmoid function σ. The value of this function ranges from 0 to 1. Forget gate f is used to show information that needs to be removed from the cell C. Whereas input states exhibit the new information that is to be added to the cell state C. The output gate O determines the output based on the sigmoid function [43].

$$f_t = \sigma(W_f h_{t-1} + U_f X_t + b_f) \quad (14)$$

$$i_t = \sigma(W_i h_{t-1} + U_i X_t + b_i) \quad (15)$$

$$O_t = \sigma(W_O h_{t-1} + U_O e_t + b_O) \quad (16)$$

8) GATED RECURRENT UNIT

It is a variant of LSTM that uses two gates and fewer parameters. The one gate is update gate u_t which is the combination of input and forget gate while the other gate is reset gate r_t that shows relevance with the previous cell state for calculating the next candidate. The state of cell is equivalent to hidden state i.e., *tanh* layer creates a new candidate vector C using r_t. The equations for GRU are formulated as [43]

$$u_t = \sigma(W_u h_{t-1} + U_u X_t + b_u) \quad (17)$$

$$r_t = \sigma(W_r h_{t-1} + U_r X_t + b_r) \quad (18)$$

$$C_t = tanh(W_c \cdot [r_t * h_{(t-1)}] + U_c e_t + b_c) \quad (19)$$

$$h_t = u_t * C + (1 - u_t) * h_{t-1} \quad (20)$$

IV. RESULTS AND DISCUSSION

This section contains the details of the experimental setup, evaluation parameters, and the discussion of experimental results.

A. EXPERIMENTAL SETUP

In this study, the experiments on the dataset are carried out using the Scikit-learn toolkit, and tweets are classified as positive, negative, and neutral sentiments. For ML models, features are extracted using a TF-IDF vectorizer, and later five classifiers BNB, MNB, LR, RF, and linear SVC are applied for the classification of tweets. For the DL methods, features are extracted using the one-hot encoder, and the classification of tweets is performed using RNN, LSTM, and GRU. These classifiers are trained on cricket data and tested on a football dataset. Moreover, the aforementioned models are applied to both balanced and unbalanced datasets. The ratio of the training and testing dataset is 80% and 20% respectively. Default parameters are used for all experiments on ML classifiers.

B. PARAMETER TUNING FOR DEEP LEARNING MODELS

For the classification of tweets, parameter tuning of DL models is performed and three deep hidden layers are selected for these models. A drop out of 0.2 is applied to each of the neural layers i.e., LSTM, GRU, and RNN. Moreover, the 'Adam' optimizer is selected with a 0.001 learning rate and default settings as it worked best over other different optimizers. 'Adam' optimizer has the combined advantages of two

TABLE 3. Parameter tuning for deep learning models.

Parameters	RNN	GRU	LSTM
Hidden layers	3	3	3
Hidden units	100	100	100
Layer types	Dense	Dense	Dense
Epochs	3,7, 10	3, 7,10	3, 7,10
Weight Initialization	Uniform	Uniform	Uniform
Activation Function	Softmax	Softmax	Softmax
Optimizer	Adam	Adam	Adam
Validation Split	10	10	10
Loss Function	Categorical Cross Entropy	Categorical Cross Entropy	Categorical Cross Entropy

TABLE 4. Results of word-gram features.

Features	Classifiers	Accuracy		Precision		Recall		F1-score	
		Bal	Un	Bal	Un	Bal	Un	Bal	Un
Unigrams	MNB	0.6	0.62	0.58	0.52	0.66	0.73	0.62	0.60
	BNB	0.62	0.64	0.62	0.55	0.59	0.67	0.61	0.60
	Linear SVC	0.57	0.61	0.58	0.55	0.60	0.66	0.59	0.60
	LR	0.6	0.62	0.61	0.53	0.60	0.70	0.60	0.60
	RF	0.59	0.61	0.64	0.51	0.51	0.69	0.57	0.58
Bigrams	MNB	0.51	0.51	0.62	0.39	0.55	0.66	0.48	0.49
	BNB	0.52	0.5	0.60	0.37	0.32	0.67	0.42	0.47
	Linear SVC	0.5	0.47	0.54	0.36	0.38	0.59	0.44	0.45
	LR	0.51	0.49	0.56	0.37	0.37	0.61	0.44	0.46
	RF	0.5	0.46	0.56	0.35	0.36	0.58	0.44	0.44
Trigram	MNB	0.37	0.35	0.34	0.29	0.83	0.84	0.48	0.44
	BNB	0.37	0.35	0.43	0.30	0.07	0.87	0.12	0.44
	Linear SVC	0.37	0.35	0.42	0.29	0.08	0.83	0.13	0.43
	LR	0.36	0.35	0.44	0.29	0.08	0.83	0.13	0.43
	RF	0.37	0.35	0.40	0.29	0.08	0.82	0.14	0.43

other extensions of stochastic gradient descent i.e., adaptive gradient algorithm and root mean square propagation. The sequence length is set to 200 and the number of hidden units is set to 64. The activation function used is softmax and the kernel regularization is set to l2. The value of the embedding dimension is set to 300. Table 3 illustrates the parameter tuning for DL models.

C. EVALUATION MEASURES

For the evaluation of results, different evaluation measures i.e., accuracy, precision, recall, and F1-score are used. Accuracy is the ratio of correctly classified instances to the total number of instances; precision is the ratio of the number of instances correctly classified as positive to the total number of positively classified instances; recall is the ratio of the instances that are classified as positive to the total number of truly positive instances, and F1-score is the harmonic mean of precision and recall [44].

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{21}$$

$$Precision = \frac{T_P}{T_P + F_P} \tag{22}$$

$$Recall = \frac{T_P}{T_P + F_N} \tag{23}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{24}$$

D. RESULTS

CDSA is performed by applying five ML classifiers on both balanced and unbalanced datasets. The experiments are performed considering the cricket domain for training and the football domain for testing. Table 4 shows the results by considering word-grams features. For a balanced dataset, BNB with word unigrams gives the best results with accuracy, precision, recall, and F1 score of 0.62, 0.62, 0.59, and 0.61 respectively. Moreover, for an unbalanced dataset also, BNB outperforms other classifiers with accuracy, precision, recall, and F1 score of 0.64, 0.55, 0.67, and 0.60, respectively. LR performs poorly with accuracy, precision, recall, and F1 score of 0.36, 0.44, 0.08, and 0.13, respectively for trigram features.

Results on the union of word grams and character grams are shown in Table 5. The maximum length of character grams is selected from 3 to 12. MNB performs best on the balanced dataset in the case of combined unigrams and character grams with accuracy, precision, recall, and F1 score of 0.61, 0.66, 0.52, and 0.58, respectively. For the unbalanced dataset, BNB with combined unigrams and character grams performed best with accuracy, precision, recall, and F1 score of 0.64, 0.55, 0.67, and 0.60, respectively.

The performance of MNB for combined trigrams and character-grams is lowest with accuracy, precision, recall, and F1 score of 0.35, 0.29, 0.84, and 0.44, respectively. A possible reason that MNB and BNB show the best performance is that our dataset is comprised of tweets that are small in length in

TABLE 5. Results of combined character-grams and word-gram features.

Features	Classifiers	Accuracy		Precision		Recall		F1-score	
		Bal	Un	Bal	Un	Bal	Un	Bal	Un
Unigrams & Character-grams	MNB	0.61	0.62	0.66	0.52	0.52	0.73	0.58	0.60
	BNB	0.56	0.64	0.69	0.55	0.36	0.67	0.47	0.60
	Linear SVC	0.56	0.61	0.56	0.55	0.69	0.66	0.61	0.60
	LR	0.6	0.62	0.59	0.53	0.65	0.70	0.62	0.60
	RF	0.55	0.6	0.65	0.48	0.43	0.72	0.52	0.58
Bigrams & Character-grams	MNB	0.57	0.51	0.64	0.39	0.40	0.66	0.49	0.49
	BNB	0.54	0.5	0.68	0.37	0.32	0.51	0.43	0.47
	Linear SVC	0.55	0.47	0.54	0.36	0.57	0.5	0.56	0.45
	LR	0.58	0.49	0.59	0.37	0.52	0.47	0.56	0.46
	RF	0.54	0.46	0.64	0.35	0.40	0.49	0.49	0.43
Trigram & Character-grams	MNB	0.51	0.35	0.66	0.29	0.31	0.84	0.42	0.44
	BNB	0.52	0.35	0.66	0.30	0.31	0.86	0.42	0.44
	Linear SVC	0.53	0.35	0.52	0.29	0.60	0.83	0.56	0.43
	LR	0.55	0.35	0.57	0.29	0.56	0.83	0.56	0.43
	RF	0.54	0.34	0.63	0.29	0.42	0.82	0.51	0.43

TABLE 6. Results of the recurrent neural network model.

Model	Layer	Batch	Epochs	Accuracy	Precision	Recall	F1-score
RNN Model	Layer 1	16	3	0.59	0.66	0.52	0.55
			7	0.72	0.74	0.71	0.72
			10	0.70	0.72	0.71	0.71
		32	3	0.59	0.65	0.50	0.56
			7	0.65	0.68	0.64	0.66
			10	0.71	0.73	0.71	0.72
		64	3	0.67	0.76	0.59	0.66
			7	0.72	0.74	0.70	0.72
			10	0.63	0.64	0.61	0.63
	Layer 2	16	3	0.55	0.57	0.53	0.55
			7	0.66	0.67	0.66	0.66
			10	0.61	0.62	0.59	0.60
		32	3	0.63	0.65	0.61	0.63
			7	0.56	0.57	0.55	0.56
			10	0.51	0.52	0.51	0.52
		64	3	0.61	0.62	0.60	0.61
			7	0.51	0.53	0.51	0.52
			10	0.57	0.58	0.56	0.57
	Layer 3	16	3	0.712	0.78	0.66	0.72
			7	0.56	0.57	0.55	0.56
			10	0.68	0.70	0.66	0.68
		32	3	0.49	0.50	0.47	0.49
			7	0.58	0.58	0.56	0.57
			10	0.59	0.60	0.58	0.59
		64	3	0.56	0.58	0.54	0.56
			7	0.50	0.51	0.50	0.51
			10	0.51	0.52	0.50	0.51

terms of characters, and these classifiers perform best on the text of shorter length. Moreover, our models do not perform well on trigram features as the resultant features from trigram drastically increase the sparsity of key features, whereas the single characters are more suited for our dataset.

The RNN model is applied by changing the hidden layers from 1 to 3 with different batch sizes and epochs to check the effects on the classification accuracy. Table 6 shows that the highest accuracy, precision, recall, and F1 score of 0.72, 0.74, 0.71, and 0.72, respectively, are achieved by the RNN model at layer 1 with a batch size of 16 and epoch size of 7. Moreover, at layer 1, batch size of 64 and epochs 7, the same accuracy of 0.72, the precision of 0.74, recall of 0.70, and F1 score of 0.72 is achieved.

Similarly, by changing the number of hidden layers, batch size, and the number of epochs, the difference in results of the LSTM model can be seen in Table 7. When the number of hidden layers is set to 2 with a batch size of 32 and an epoch of 3, LSTM gives better results as compared to RNN. The highest accuracy, precision, recall, and F1-score of 0.76, 0.79, 0.72, and 0.75 are achieved respectively.

By using different hidden layers, batch size, and the number of epochs the results of the GRU model are shown in Table 8. GRU performed best with one hidden layer, batch

TABLE 7. Results of the long short-term memory model.

Model	Layer	Batch	Epochs	Accuracy	Precision	Recall	F1-score
LSTM Model	Layer 1	16	3	0.68	0.75	0.60	0.67
			7	0.68	0.65	0.61	0.62
			10	0.73	0.75	0.73	0.74
		32	3	0.62	0.68	0.56	0.61
			7	0.61	0.64	0.60	0.62
			10	0.60	0.62	0.59	0.60
		64	3	0.64	0.71	0.55	0.62
			7	0.69	0.72	0.68	0.70
			10	0.58	0.60	0.58	0.59
	Layer 2	16	3	0.69	0.73	0.66	0.69
			7	0.62	0.63	0.61	0.62
			10	0.66	0.68	0.66	0.67
		32	3	0.76	0.79	0.72	0.75
			7	0.72	0.74	0.72	0.73
			10	0.59	0.59	0.59	0.59
		64	3	0.64	0.67	0.62	0.65
			7	0.60	0.69	0.59	0.59
			10	0.61	0.62	0.61	0.62
	Layer 3	16	3	0.71	0.74	0.68	0.71
			7	0.68	0.69	0.67	0.68
			10	0.69	0.70	0.67	0.69
		32	3	0.68	0.70	0.68	0.69
			7	0.68	0.70	0.68	0.69
			10	0.66	0.67	0.66	0.66
		64	3	0.64	0.68	0.60	0.64
			7	0.61	0.63	0.61	0.63
			10	0.59	0.60	0.59	0.59

TABLE 8. Results of the gated recurrent unit model.

Model	Layer	Batch	Epochs	Accuracy	Precision	Recall	F1-score
GRU Model	Layer 1	16	3	0.67	0.73	0.62	0.67
			7	0.62	0.65	0.60	0.62
			10	0.731	0.73	0.72	0.73
		32	3	0.77	0.83	0.68	0.75
			7	0.64	0.66	0.61	0.63
			10	0.65	0.67	0.65	0.65
		64	3	0.72	0.84	0.70	0.76
			7	0.60	0.62	0.58	0.60
			10	0.57	0.58	0.55	0.57
	Layer 2	16	3	0.67	0.69	0.66	0.67
			7	0.64	0.65	0.63	.64
			10	0.56	0.56	0.55	0.55
		32	3	0.72	0.73	0.69	0.71
			7	0.61	0.62	0.61	0.62
			10	0.64	0.65	0.64	0.64
		64	3	0.72	0.73	0.69	0.71
			7	0.69	0.70	0.68	0.69
			10	0.57	0.58	0.56	0.57
	Layer 3	16	3	0.66	0.62	0.64	0.64
			7	0.68	0.69	0.68	0.68
			10	0.64	0.65	0.64	0.64
		32	3	0.67	0.69	0.65	0.67
			7	0.68	0.69	0.67	0.68
			10	0.60	0.60	0.59	0.60
		64	3	0.67	0.69	0.65	0.67
			7	0.62	0.64	0.61	0.62
			10	0.63	0.65	0.64	0.64

size 32, and with 3 epochs. The highest accuracy achieved by GRU is 0.77, precision is 0.83, recall is 0.68, and F1-score is 0.75.

In this study, we attempted to accomplish agreeable accuracy for the task of CDSA as a baseline for future experimentation. In the case of ML models, the results show that word grams gave better results as compared to combined character grams and word grams on our dataset.

For DL models, the unbalanced dataset is used because it works better on larger datasets. The results of DL models are better as compared to ML classifiers. Firstly, RNN is applied to our dataset because RNN can link previously learned

information to the current data, which is our requirement because context is important in identifying the polarity of a sentence. However, RNNs are not able to keep track of long-term dependencies therefore, different variants of RNN are used, i.e., LSTM and GRU, and observed an increase in accuracy.

The highest accuracy of 77% is achieved by GRU. GRU is better than LSTM and RNN because it does not require memory units and is faster to train. Moreover, it is observed that increasing the number of hidden layers has no significant increase in accuracy. It is also observed that by increasing the number of epochs, the models start to overfit as the training accuracy increases but the testing accuracy is reduced.

E. COMPARISON WITH STANDARD DATASETS

To validate the performance of the proposed approach, our best performing model i.e., GRU is tested on two existing, widely used, standard datasets [5], [45]. The GRU model is trained on the dataset comprising the cricket domain and tested on the aforementioned standard datasets. Our model achieves encouraging results on the dataset of Khan and Nizami [45] which indicates that our model can adapt well to other datasets. However, the results on the dataset of Mukhtar and Khan [5] are not very encouraging. A possible reason for low performance is that the proposed model is trained on tweets that comprise small sentences containing fewer characters whereas the dataset used by Mukhtar and Khan [5] is composed of large length sentences and is a combination of 14 different domains.

TABLE 9. Performance comparison of the proposed model on standard datasets.

Dataset	Accuracy
Proposed	0.77
[45]	0.69
[5]	0.32

V. CONCLUSION AND FUTURE WORK

This work presents an approach to perform cross-domain sentiment analysis for the Urdu language. Tweets are collected from two domains i.e., cricket and football. The source domain is cricket, and the target domain is football. Comprehensive guidelines are developed, and tweets are annotated with the help of three linguistic experts. Training is performed on the cricket domain and testing is performed on the football domain. Classical machine learning and deep learning methods are explored for the classification of tweets. For machine learning models, five classifiers MNB, BNB, LR, RF, and linear SVC are used, and experiments are performed on both balanced and unbalanced datasets. BNB outperforms all classifiers with accuracy, precision, recall, and F1-score of 0.62, 0.62, 0.59, and 0.60, respectively, for the balanced dataset. Moreover, for an unbalanced dataset also, BNB outperforms other classifiers with accuracy, precision, recall, and F1 scores of 0.64, 0.55, 0.67, and 0.60, respectively. In addition to this, deep learning models i.e., RNN, LSTM, and GRU are also used for the classification task where GRU performs best with accuracy, precision, recall, and F1-score of 0.77, 0.83, 0.68, and 0.75, respectively. Overall, the performance of deep learning models is higher than the machine learning classifiers. It is also observed that increasing the number of hidden layers has no significant effect on accuracy. To validate our proposed model, our best performing model is tested on two standards, widely used, Urdu datasets. This study serves as the baseline for future research in cross-domain sentiment analysis in low-resource languages like Urdu. In the future, we intend to increase the accuracy of cross-domain sentiment analysis for Urdu.

REFERENCES

[1] I. Inuwa-Dutse, M. Liptrott, and I. Korkontzelos, "Detection of spam-posting accounts on Twitter," *Neurocomputing*, vol. 315, pp. 496–511, Nov. 2018.

[2] A. Geethapriya and S. Valli, "An enhanced approach to map domain-specific words in cross-domain sentiment analysis," *Inf. Syst. Frontiers*, vol. 23, no. 3, pp. 791–805, Jun. 2021.

[3] T. Li, V. Sindhwani, C. Ding, and Y. Zhang, "Knowledge transformation for cross-domain sentiment classification," in *Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, New York, NY, USA, 2009, pp. 716–717.

[4] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Jul. 2002, pp. 79–86.

[5] N. Mukhtar and M. A. Khan, "Urdu sentiment analysis using supervised machine learning approach," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 32, no. 2, 2018, Art. no. 1851001.

[6] H. Tang, S. Tan, and X. Cheng, "A survey on sentiment detection of reviews," *Expert Syst. Appl.*, vol. 36, no. 7, pp. 10760–10773, 2009.

[7] L. Khan, A. Amjad, N. Ashraf, H.-T. Chang, and A. Gelbukh, "Urdu sentiment analysis with deep learning methods," *IEEE Access*, vol. 9, pp. 97803–97812, 2021.

[8] K. Mehmood, "On multi-domain sentence level sentiment analysis for Roman Urdu," School Eng. Inf. Technol., UNSW Sydney, 2021.

[9] X. Li, W. Zhang, H. Ma, Z. Luo, and X. Li, "Deep learning-based adversarial multi-classifier optimization for cross-domain machinery fault diagnostics," *J. Manuf. Syst.*, vol. 55, pp. 334–347, Apr. 2020.

[10] Z. Gao, T. T. Han, L. Zhu, H. Zhang, and Y. Wang, "Exploring the cross-domain action recognition problem by deep feature learning and cross-domain learning," *IEEE Access*, vol. 6, pp. 68989–69008, 2018.

[11] S. K. Singh and M. K. Sachan, "Classification of code-mixed bilingual phonetic text using sentiment analysis," *Int. J. Semantic Web Inf. Syst.*, vol. 17, no. 2, pp. 59–78, Apr. 2021.

[12] I. Wedel, M. Palk, and S. Voß, "A bilingual comparison of sentiment and topics for a product event on Twitter," *Inf. Syst. Frontiers*, pp. 1–12, Jul. 2021.

[13] I. U. Khan, A. Khan, W. Khan, M. M. Su'ud, M. M. Alam, F. Subhan, and M. Z. Asghar, "A review of Urdu sentiment analysis with multilingual perspective: A case of Urdu and Roman Urdu language," *Computers*, vol. 11, no. 1, p. 3, Dec. 2021.

[14] M. M. Agüero-Torales, J. I. Abreu Salas, and A. G. López-Herrera, "Deep learning and multilingual sentiment analysis on social media data: An overview," *Appl. Soft Comput.*, vol. 107, Aug. 2021, Art. no. 107373.

[15] Y. Hao, T. Mu, R. Hong, M. Wang, X. Liu, and J. Y. Goulermas, "Cross-domain sentiment encoding through stochastic word embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 10, pp. 1909–1922, Oct. 2020.

[16] L. Yang, Y. Li, J. Wang, and R. S. Sherratt, "Sentiment analysis for e-commerce product reviews in Chinese based on sentiment lexicon and deep learning," *IEEE Access*, vol. 8, pp. 23522–23530, 2020.

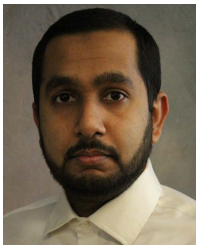
[17] Z. Nasim and S. Ghani, "Sentiment analysis on Urdu tweets using Markov chains," *SN Comput. Sci.*, vol. 1, no. 5, p. 269, Sep. 2020.

[18] K. Mehmood, D. Essam, K. Shafi, and M. K. Malik, "Discriminative feature spamming technique for Roman Urdu sentiment analysis," *IEEE Access*, vol. 7, pp. 47991–48002, 2019.

[19] L. Khan, A. Amjad, N. Ashraf, and H.-T. Chang, "Multi-class sentiment analysis of Urdu text using multilingual BERT," *Sci. Rep.*, vol. 12, no. 1, p. 5436, Dec. 2022.
[20] B. A. Chandio, A. S. Imran, M. Bakhtyar, S. M. Daudpota, and J. Baber, "Attention-based RU-BiLSTM sentiment analysis model for Roman Urdu," *Appl. Sci.*, vol. 12, no. 7, p. 3641, Apr. 2022.
[21] F. Noor, M. Bakhtyar, and J. Baber, "Sentiment analysis in E-commerce using SVM on Roman Urdu text," in *Emerging Technologies in Computing*, M. H. Miraz, P. S. Excell, A. Ware, S. Soomro, and M. Ali, Eds. Cham, Switzerland: Springer, 2019, pp. 213–222.
[22] N. Mukhtar, M. Abid Khan, N. Chiragh, S. Nazir, and A. Ullah Jan, "An intelligent unsupervised approach for handling context-dependent words in Urdu sentiment analysis," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 21, no. 5, pp. 1–15, Sep. 2022.
[23] F. HosseinzadehBendarkheili, R. MohammadiBaghmolaei, and A. Ahmadi, "Product quality assessment using opinion mining in Persian online shopping," in *Proc. 27th Iranian Conf. Electr. Eng. (ICEE)*, Apr. 2019, pp. 1917–1921.
[24] Y. M. Aye and S. S. Aung, "Sentiment analysis for reviews of restaurants in Myanmar text," in *Proc. 18th IEEE/ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distrib. Comput. (SNPD)*, Jun. 2017, pp. 321–326.
[25] O. Alqaryouti, N. Siyam, A. A. Monem, and K. Shaalan, "Aspect-based sentiment analysis using smart government review data," *Appl. Comput. Informat.*, vol. 16, no. 1, pp. 1–20, Nov. 2019.
[26] H. S. Ibrahim, S. M. Abdou, and M. Gheith, "Idioms-proverbs lexicon for modern standard Arabic and colloquial sentiment analysis," *Int. J. Comput. Appl.*, vol. 118, no. 11, pp. 26–31, May 2015.
[27] K. Mehmood, D. Essam, K. Shafi, and M. K. Malik, "Sentiment analysis for a resource poor language—Roman Urdu," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 19, no. 1, pp. 1–15, Aug. 2019.
[28] R. Bibi, U. Qamar, M. Ansar, and A. Shaheen, "Sentiment analysis for Urdu news tweets using decision tree," in *Proc. IEEE 17th Int. Conf. Softw. Eng. Res., Manage. Appl. (SERA)*, May 2019, pp. 66–70.
[29] M. Khan and K. Malik, "Sentiment classification of customer's reviews about automobiles in Roman Urdu," in *Advances in Information and Communication Networks*, K. Arai, S. Kapoor, and R. Bhatia, Eds. Cham, Switzerland: Springer, 2019, pp. 630–640.
[30] N. Mukhtar, M. A. Khan, and N. Chiragh, "Effective use of evaluation measures for the validation of best classifier in Urdu sentiment analysis," *Cognit. Comput.*, vol. 9, no. 4, pp. 446–456, Aug. 2017.
[31] L. Khan, A. Amjad, K. M. Afaq, and H.-T. Chang, "Deep sentiment analysis using CNN-LSTM architecture of English and Roman Urdu text shared in social media," *Appl. Sci.*, vol. 12, no. 5, p. 2694, Mar. 2022.
[32] U. Naqvi, A. Majid, and S. Ali Abbas, "UTSA: Urdu text sentiment analysis using deep learning methods," *IEEE Access*, vol. 9, pp. 114085–114094, 2021.
[33] K. Dashtipour, M. Gogate, J. Li, F. Jiang, B. Kong, and A. Hussain, "A hybrid Persian sentiment analysis framework: Integrating dependency grammar based rules and deep neural networks," *Neurocomputing*, vol. 380, pp. 1–10, Mar. 2020.
[34] W. Li, F. Qi, M. Tang, and Z. Yu, "Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification," *Neurocomputing*, vol. 387, pp. 63–77, Apr. 2020.
[35] M. Rathi, A. Malik, D. Varshney, R. Sharma, and S. Mendiratta, "Sentiment analysis of tweets using machine learning approach," in *Proc. 11th Int. Conf. Contemp. Comput. (IC3)*, Aug. 2018, pp. 1–3.
[36] S. Bagui, D. Nandi, S. Bagui, and R. J. White, "Classifying phishing email using machine learning and deep learning," in *Proc. Int. Conf. Cyber Secur. Protection Digit. Services (Cyber Security)*, Jun. 2019, pp. 1–2.
[37] G. Singh, B. Kumar, L. Gaur, and A. Tyagi, "Comparison between multinomial and Bernoulli Naïve Bayes for text classification," in *Proc. Int. Conf. Autom., Comput. Technol. Manage. (ICACTM)*, Apr. 2019, pp. 593–596.
[38] W. Bourequat and H. Mourad, "Sentiment analysis approach for analyzing iPhone release using support vector machine," *Int. J. Adv. Data Inf. Syst.*, vol. 2, no. 1, pp. 36–44, Apr. 2021.
[39] H. M. Ahmed, M. Javed Awan, N. S. Khan, A. Yasin, and H. M. Faisal Shehzad, "Sentiment analysis of online food reviews using big data analytics," *Elementary Educ. Online*, vol. 20, no. 2, pp. 827–836, 2021.
[40] K. Jindal and R. Aron, "A systematic study of sentiment analysis for social media data," *Mater. Today, Proc.*, pp. 1–12, 2021.
[41] K. Kowsari, K. J. Meimandi, M. Heidarysafa, and S. Mendu, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, Apr. 2019.
[42] J. Wang, B. Peng, and X. Zhang, "Using a stacked residual LSTM model for sentiment intensity prediction," *Neurocomputing*, vol. 322, pp. 93–101, Dec. 2018.
[43] S. O. Alhumoud and A. A. Al Wazrah, "Arabic sentiment analysis using recurrent neural networks: A review," *Artif. Intell. Rev.*, vol. 55, no. 1, pp. 707–748, Jan. 2022.
[44] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," *Expert Syst. Appl.*, vol. 57, pp. 117–126, Sep. 2016.
[45] M. Y. Khan and M. S. Nizami, "Urdu sentiment corpus (v1.0): Linguistic exploration and visualization of labeled dataset for Urdu sentiment analysis," in *Proc. Int. Conf. Inf. Sci. Commun. Technol. (ICISCT)*, Feb. 2020, pp. 1–15.

AMNA ALTAF received the B.S. degree in information technology from the University of the Punjab–Gujranwala Campus, and the M.S. degree in computer science from COMSATS University Islamabad (Lahore Campus), where she is currently pursuing the Ph.D. degree in computer science. She is currently working as a Lecturer at the University of Management and Technology, Lahore. Her research interests include natural language processing, computation linguistics, and data mining.

MUHAMMAD WAQAS ANWAR received the M.S. degree in computer science from Hamdard University, Pakistan, in 2001, and the Ph.D. degree in computer application technology from the Harbin Institute of Technology, China, in 2008. From 2008 to 2012, he was a Lecturer and an Assistant Professor with the Department of Computer Science, COMSATS Institute of Information Technology, Lahore, Pakistan (currently COMSATS University Islamabad, Lahore campus). Since 2012, he has been an Associate Professor with the Department of Computer Science, COMSATS University Islamabad (Lahore campus), Pakistan. His research interests include natural language processing, computational intelligence, and bioinformatics.



MUHAMMAD HASAN JAMAL received the B.S. degree in computer and information engineering from International Islamic University Malaysia, in 2005, the M.S. degree in computer engineering from the University of Engineering and Technology, Lahore, Pakistan, in 2008, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2015. From 2006 to 2010, he was a Research Associate and a Senior Research Associate with the Al-Khawarizmi Institute of Computer Science, UET, Lahore. He was a Graduate Technical Summer Intern at the Sandia National Laboratory, Albuquerque, NM, in summer 2015. Since 2016, he has been an Assistant Professor with the Department of Computer Science, COMSATS University Islamabad (Lahore Campus). His research interests include parallel and distributed systems, scientific computing, data analytics, and natural language processing. He was a recipient of the Fulbright Scholarship for his Ph.D. studies.

SANA HASSAN received the B.S. and M.S. degrees in computer science from COMSATS University Islamabad (Lahore Campus), in 2018 and 2020, respectively. She is currently working as a Course Author and a Product Associate at Arbisoft, one of the leading software houses in Pakistan. Her research interests include natural language processing and robotics.

USAMA IJAZ BAJWA received the M.S. and Ph.D. degrees from the Center for Advanced Studies in Engineering, Islamabad, Pakistan, in 2006 and 2013, respectively. He is currently a Tenured Associate Professor with the Department of Computer Science, COMSATS University Islamabad, Lahore Campus, a Research Supervisor, the Program Chair of international conference on FIT, CO-PI of two funded projects, the Head of the Machine Perception and Visual Intelligence Research Group. Previously, an Assistant Professor and a Graduate Program Coordinator at CUI Abbottabad, the Head of the Information Security and Image Processing Research Group, CUI Abbottabad, a Team Lead in and ICT Research and Development Funded Project, a Co-Founder of Technology Nucleus Pvt. Ltd., a Visiting Researcher at the Medical Imaging Laboratory (University of South Wales). He has authored/coauthored 60 international conference and journal research papers. His research interests include video analytics, medical image analysis, and biometrics.

GYU SANG CHOI received the Ph.D. degree from the Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, in 2005. He was a Research Staff Member at the Samsung Advanced Institute of Technology (SAIT) for Samsung Electronics, from 2006 to 2009. Since 2009, he has been a Faculty Member at the Department of Information and Communication, Yeungnam University, South Korea. His research interests include non-volatile memory and storage systems.

IMRAN ASHRAF received the M.S. degree in computer science from the Blekinge Institute of Technology, Karlskrona, Sweden, in 2010, and the Ph.D. degree in information and communication engineering from Yeungnam University, Gyeongsan, South Korea, in 2018. He has worked as a Postdoctoral Fellow at Yeungnam University. He is currently working as an Assistant Professor at the Information and Communication Engineering Department, Yeungnam University. His research interests include indoor positioning and localization, indoor location-based services in wireless communication, smart sensors (LiDAR) for smart cars, and data mining.