Received 5 February 2023, accepted 12 February 2023, date of publication 16 February 2023, date of current version 23 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3246388

RESEARCH ARTICLE

Attentional Multi-Channel Convolution With Bidirectional LSTM Cell Toward Hate Speech Prediction

MOHD FAZIL[1], SHAKIR KHAN[2,3], (Member, IEEE), BADER M. ALBAHLAL[2], REEMIAH MUNEER ALOTAIBI[2], TAMANNA SIDDIQUI[4], AND MOHD ASIF SHAH[5,6,7]

[1]Center for Transformative Learning, University of Limerick, Limerick, V94 T9PX Ireland
[2]College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University, Riyadh 11432, Saudi Arabia
[3]University Center for Research and Development, Department of Computer Science and Engineering, Chandigarh University, Mohali 140413, India
[4]Department of Computer Science, Aligarh Muslim University, Aligarh, Uttar Pradesh 202002, India
[5]Department of Economics, Kebri Dehar University, Kebri Dehar 250, Ethiopia
[6]School of Business, Woxsen University, Kamkole, Sadasivpet, Hyderabad, Telangana 502345, India
[7]Division of Research and Development, Lovely Professional University, Phagwara, Punjab 144001, India

Corresponding authors: Mohd Asif Shah (drmohdasifshah@kdu.edu.et) and Shakir Khan (sgkhan@imamu.edu.sa)

This work was supported by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) through the Research Partnership Program under Grant RP-21-07-06.

ABSTRACT Online social networks(OSNs) facilitate their users in real-time communication but also open the door for several challenging problems like hate speech and fake news. This study discusses hate speech on OSNs and presents an automatic method to identify hate messages. We introduce an attentional multi-channel convolutional-BiLSTM network for the classification of hateful content. Our model uses existing word representation techniques in a multi-channel environment having several filters with different kernel sizes to capture semantics relations at various windows. The encoded representation from multiple channels passes through an attention-aware stacked 2-layer BiLSTM network. The output from stacked 2-layer BiLSTM is weighted by an attention layer and further concatenated and passes via a dense layer. Finally, an output layer employing a sigmoid function classifies the text. We investigate the efficacy of the presented model on three Twitter-related benchmark datasets considering four evaluation metrics. In comparative evaluation, our model beats the five state-of-the-art and the same number of baseline models. The ablation study shows that the exclusion of channels and attention mechanism has the highest impact on the performance of the presented model. The empirical analysis analyzing the impact of different word representation techniques, optimization algorithms, activation functions, and batch size on the presented model ascertains the use of their optimal values.

INDEX TERMS Multi-channel deep learning, data-driven cyber security, hate speech, online social networks.

I. INTRODUCTION

Online social media users utilize them to communicate, share life events, exhibit opinions on news and trending events, and connect with friends and celebrities [1]. The user interaction and activities over these OSNs generate a large amount of multi-model content, which are utilized by researchers for social network analysis [2], event detection [3], socialbot detection [4], [5]. The open nature, anonymity, and real-time communication also facilitate the adversaries to perform their malicious activities. Sometimes, interaction with OSNs leads to heated arguments like hate speech and abusive responses. Controversial content generally receives higher responses from users. These reactions can contain abusive, bullying, or harassing content depending on the political and ideological inclination. The offensive content on OSNs sometimes harms offline worlds like physical violence and riots.[1] The existing literature depending on

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

[1]https://www.washingtonpost.com/technology/2021/10/22/jan-6-capitol-riot-facebook/

the intent and target classifies the derogatory content as racism, sexism, hate speech, trolling, and abusive. The hate speech includes content to intimidate with the intent to physically harm or promote violence against an individual or community [6]. However, various platforms and researchers define hate speech with subtle differences. Twitter policy states that any tweet that directly/indirectly threatens or attacks a person/community based on race, religion, ethnicity, gender, religion, or disability, will be considered hate speech.

Social media platforms generally have a content moderation team to track and filter offensive content. However, manual moderation is inefficient and infeasible due to the volume of content generated every second [7]. The researchers also analyzed the various aspect of the hate speech problem and presented categories of approaches for its detection [8], [9], [10]. However, the existing methods are still not comprehensive and require improvement. The literature has machine learning models to solve a range of applications like health informatics [11]. Likewise, researchers applied machine learning toward modeling and detecting toxic content on online social networks. Broadly, hate detection models are of two types: classical machine learning models and deep learning models, wherein the first category of methods generally uses hand-crafted features. They also use the bag-of-words model for text representation, which doesn't incorporate the semantic relationship among the words.

A. OUR CONTRIBUTION

The deep learning methods employing the word embeddings capture these semantics and show promising results. The existing deep models have extensively exploited recurrent neural networks (RNN) [12], [13] and convolutional neural networks (CNNs) [14], [15] towards detecting different categories of offensive content. Both RNNs and CNN are good at capturing the semantic relationship between words. This study investigates the hate speech issue on the OSN as a binary-class problem. Most of the existing hate speech detection approaches use convolution neural networks as a single channel. Researchers have used multi-channel neural networks for sentiment analysis, comment toxicity prediction, and stance prediction [16], [17]. To the best of our knowledge, this is the first study that presents a multi-channel convolutional network with attention-aware stacked BiLSTM to detect hate speech in OSNs. The presented model replicates a CNN model with different kernel sizes. It creates three parallel networks, each having the embedding layer as the first layer populated using GloVe embedding. Next, a convolution layer with different kernel sizes after each of the three embedding layers is embedded to extract local features using multiple filters. The convolution output is passed through a 2-layers stacked BiLSTM network to learn sequential representation. The learned encoding from the BiLSTM network is assigned weight using an attention layer to give higher weight to vital features. Next, our model integrates the encoded representation from three parallel networks employing a concatenation layer. The integrated output passes through a fully connected layer. Further, the dense layer transfers the encoded representation to a sigmoid layer to classify it into the desired classes: hate speech and non-hate. Briefly, the main contributions are as follows:

- Introduce a novel multi-channel attention-aware convolutional-BiLSTM network for efficient representation learning toward detecting hate speech.
- Perform and evaluate the efficacy of our model toward hate speech detection on three Twitter-related benchmark datasets compared to 10 models, including 5 SOTA and five baseline methods.
- Perform the empirical analysis to observe the impact of various deep learning hyper-parameters on the presented model and to identify their optimal values.

We organize the remaining paper as follows. The existing approaches toward hate speech detection are briefly discussed in Section II. Next, we discuss our model and constituting layers in section III. The description of the dataset, experimental evaluation, and comparison with SOTA and baseline is performed in Section IV. Section V evaluates the impact of different deep learning hyperparameters values on the model efficacy. Finally, we conclude the manuscript in Section VI along with a brief description of the potential extension of our model.

II. RELATED WORKS

Researchers have examined hate speech in OSNs from different perspectives. The researchers employed the advancement in representation learning toward detecting various categories of hate speech. The existing approaches generally use machine learning-based modeling, including pattern mining and statistical analysis-based methods. Feature engineering and deep learning-based methods are two significant categories of machine learning methods.

The feature engineering approaches extract content, profile, template, network, and other features to represent each instance. Further, support vector machine, random forest, Naive Bayes, and other ML models are trained and validated over the test dataset. Warner and Hirschberg [7] employed word-gram, template, and part of speech-tag-based features and trained the SVM^{light} to detect hate speech in OSN. The authors investigated that the model performs poorly with bigram and tri-gram features. In [18], authors used one-gram, bigram, tri-gram, and sentiment features to train the Naive Bayes model for racist content classification. In similar approaches, authors used n-gram and typed dependencies-based attributes, and further trained logistic regression, voted ensemble, random forest, and other classifiers for hate speech detection [8], [19], [20]. In [8], authors trained training logistic regression over n-gram features for classification. They also annotated 16k tweets into hate and non-hateful categories. Davidson et al. [21] discussed various classes of offensive languages and defined their difference. They also used the n-gram feature with the sentiment and other linguistic features to train logistic regression for hate speech classification. Malmasi and Zampieri [22] employed both character and word n-gram for classification model training and found that the character-gram model outperforms the word-gram model.

Recently, researchers have introduced numerous neural network models for different classification problems because traditional machine learning models require manual devising of features, which is a tedious job [1], [4], [23]. Similarly, researchers also employed various neural network components to propose deep learning models to automatically detect hate speech [12], [15], [24], [25], [26]. In [24], authors represented and classified aggressive comments employing the paragraphIIvec language model [27]. Vignal et al. [25] used SVM and LSTM models employing syntactical, sentiment, and embedding-based features to represent hateful content for classification. Park and Fung [26] presented a CNN-based model using pre-trained word embedding at the embedding layer for abusive content detection. Zhang et al. [12] proposed a CNN-GRU incorporating hybrid network to detect hate speech. In [14], the authors employed a deep CNN network, outperforming the existing SOTA approaches toward hate speech detection. Recently, Khan et al. [10] integrated the CNN, BiLSTM with a capsule network for hate speech. In [28], authors introduced BiCHAT, a deep learning model integrating deep CNN, BiLSTM, and attention mechanism with BERT-based word representation learning. The author empirically showed that the integration of deep CNN improved the performance for hate speech classification. In [29], Rezaeinia et al. introduced a multi-block convolutional highways network and evaluated it for text classification. Quan et al. [30] proposed a multi-channel CNN for biomedical relation extraction from medical logs. Yoon and Kim [31] introduced a hybrid network integrating CNN and BiLSTM network with multi-channel lexicon embedding to classify the sentiment. In [16], the authors introduced another multi-channel network employing CNN and LSTM networks to classify sentiment in texts, outperforming the baseline methods. In another similar model, Li et al. [17] used a two-channel network to predict stance in texts. Recently, Kumar et al. [32] introduced a multi-channel model to detect toxic comments in a multi-label setting. The model employs the convolutional gated recurrent unit network. In summary, researchers have used multi-channel neural networks for sentiment analysis, comment toxicity prediction, and stance prediction [16], [17]. To the best of our knoledgem this is the first study that presents a multi-channel convolutional network with attention-aware stacked BiLSTM to detect hate speech in OSNs.

III. PROPOSED MODEL

The work flow of the proposed model is shown in Figure 1. In brief, it contains three parallel attention-aware convolutional stacked BiLSTM networks. The following subsections describe various components of the introduced model.

A. DATA GATHERING AND PREPROCESSING

The efficacy of the presented model is evaluated on three Twitter-related benchmark datasets provided by Founta et al. [33], Davidson et al. [21], and Mathew et al. [34]. This section pre-processes the raw datasets to filter useless content. In the pre-processing, we filter Twitter-centric and other noisy content such as URL, Hashtag (#), retweets (RT), and mentions (@). We also filter the username because it is not informative. We also remove alphanumeric characters, such as numbers, non-ASCII symbols, ampersands, and other special characters. We also filter the stop words. In the end, it converts the cleaned text to lowercase.

B. INPUT LAYER

The input layer accepts the pre-processed tweets and splits them into words (tokens). Further, it creates a set of unique words and assigns each one a number, called an index. This procedure creates a dictionary with key as word and index as value. After this, the input layer tokenizes each tweet of the corpus and maps the words to the underlying index values from the constructed dictionary. This procedure converts each input tweet into a numeric vector. For example, an input tweet T is tokenized into n words (tokens) and each word is replaced with an index value such that T is mapped into an n dimensional vector $T \in \Re^{1\times n}$. In the mapping process, the maximum length of the numeric vectors is fixed. Further, padding (this paper uses post-padding) is applied to fix the size of the numeric vector, i.e., for each tweet $T \in \Re^{1\times p}$. Finally, the input layer maps the corpus into an input matrix $M \in \Re^{N\times p}$ where N represents the number of tweets in the corpus.

C. MULTICHANNEL CONVOLUTIONAL RECURRENT NEURAL NETWORK

This section describes in detail the proposed multi-channel convolutional recurrent neural network. A multi-channel environment replicates a convolutional network with different kernel sizes to extract the semantic relationship at various n-gram levels. In the proposed architecture, the multi-channel convolutional network contains a convolution layer, a pooling layer, a dropout layer, a 2-layer stacked BiLSTM layer, and a final dropout layer. This network architecture is replicated thrice in parallel to capture 1-, 2-, and 3-gram semantics. In the existing literature, researchers have generally used up to 3-gram to capture word-level contextual semantics [22]. Therefore, the proposed model uses up to 3-gram. The following subsections present further discussion of each component of the network.

1) MULTICHANNEL WORD EMBEDDING

In the embedding layer, preprocessed and cleaned text is parsed and converted to a numeric matrix wherein each row corresponds to a numeric vector for the underlying word. To this end, it repaces each word with a vector using the pre-trained 100-dimensional representation of GloVe [35] word

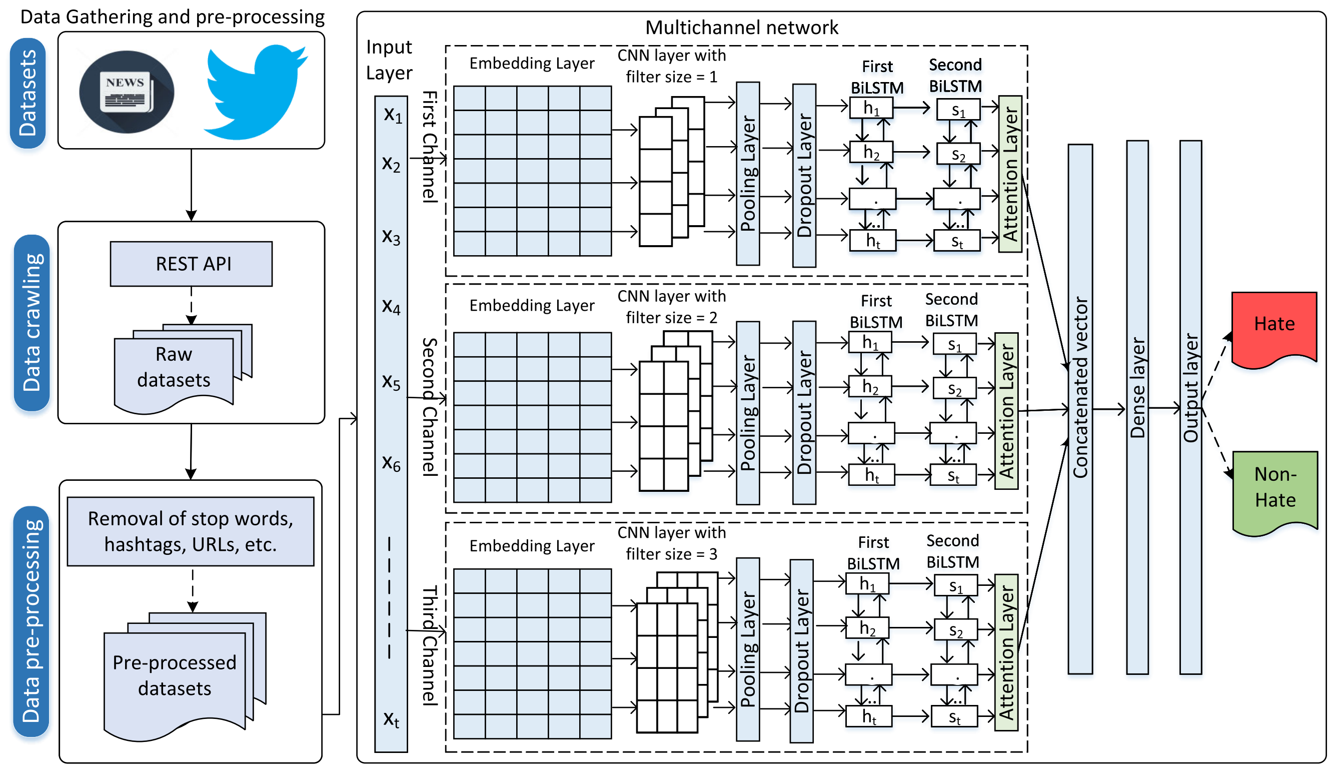


FIGURE 1. Architecture of our multi-channel model.

embedding. As shown in the figure, we generate the word embedding matrix for the three channels to extract features in parallel from the same data.

2) MULTICHANNEL CONVOLUTIONAL NEURAL NETWORK

Initially, CNN was designed to process image data and perform image classification [36], but later researchers applied it in NLP & other text processing-related applications [37]. The proposed model applies a convolution layer to each network with different kernel sizes starting from $h = 1$ up to $h = 3$. The proposed model uses a 1-D convolutional layer with several filters W in a multichannel environment. For example, for channel i, filters with kernel size h will be $W^i \in R^{hd}$, where d is the embedding dimension. The embedding matrix for channel i using maximum sequence length N is $M^i \in R^{Nd}$.

3) POOLING LAYER

The CNN-extracted feature map is passed to a pooling layer. In a CNN network, the pooling layer extracts the key features from the feature map. As a result, the pooling operation significantly reduces the feature map size. The pooling layer can apply strategies like max and average for feature selection. This study uses max pooling to select the key feature.

4) ATTENTION-AWARE STACKED BiLSTM NETWORK

The pooling layer passes the extracted key features to a dropout layer before sending it to a 2-layer stacked BiLSTM network. Bidirectional LSTM is an advanced version of LSTM and contains a memory block to process sequential information. Consequently, unlike simple RNN, it does not suffer from the vanishing gradient problem. The memory block gives LSTM the required competency to decide what to forget and what to remember. It also enables it to learn long-distance contextual dependencies. Figure 2 depicts an example LSTM cell consisting of its components: input gate, forget gate, output gate, and a memory cell state. In an LSTM cell, the input gate controls the flow of information by computing its value at time t using equation 1. Equation 2 computes the information which will be erased at time t. Similarly, equation 3, 4 and 5 compute the $\tilde{C}_t$ (candidate cell state), C_t (current cell state) and value of output gate o_t, respectively at a timestamp t. Finally, equation 6 calculates LSTM cell output at time t.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{1}$$
$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2}$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$
$$C_t = f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t \tag{4}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$
$$h_t = o_t \otimes \tanh(C_t) \tag{6}$$

We employ Bidirectional LSTM to capture both left-to-right and right-to-left contexts. To this end, it processes the sequential information using a forward LSTM to capture the left-to-right context and a backward LSTM to capture

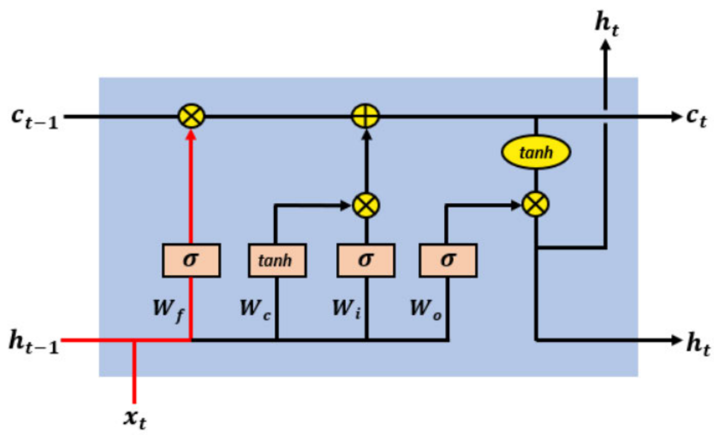


FIGURE 2. An example LSTM cell.

the right-to-left context. As a result, it effectively captures historical and ahead in sequence contexts. The forward and backward LSTM cells generate $\overrightarrow{h}_i$ and $\overleftarrow{h}_i$ hidden representations, respectively using equation 7 and 8. Finally, the the LSTM concatenates the outputs from the two cells to generate the final hidden representation h_i from the BiLSTM cell as done in equation 9. The deep RNNs are effective low-level feature extractors [38]. Therefore, to exploit the strengths of Bidirectional LSTM and deep RNN, we use stacked 2-layer BiLSTM in this paper. To assign a variable weight, stacked BiLSTM output is passed to an attention layer proposed by Bhadanau et al. [39] to assign weight to each encoded feature depending on their importance.

$$\overrightarrow{h}_i = \overleftarrow{LSTM}(f_i) \tag{7}$$

$$\overleftarrow{h}_i = \overrightarrow{LSTM}(f_i) \tag{8}$$

$$h_i = [\overleftarrow{h}_i, \overrightarrow{h}_i] \tag{9}$$

5) CONCATENATION LAYER

This layer concatenates the encoded representation from three parallel channels after applying a dropout of 0.5. The concatenation representation includes the semantic information computed at 1-gram, 2-gram, and 3-gram.

6) DENSE AND OUTPUT LAYER

The concatenated output is passed to a dense layer for further encoding. The encoded result is given to the final layer having sigmoid activation to classify the hate (T_H) from genuine content (T_{NH}).

IV. EXPERIMENTAL EVALUATION

The efficacy of the presented model toward hate speech detection is established by evaluating it on three Twitter datasets. To this end, this section describes the experimental setting, dataset, and experimental results, including comparative evaluation with 10 comparison approaches.

TABLE 1. Dataset statistics.

Datasets	Raw dataset		Pre-processed Imbalanced		Pre-processed Balanced	
	#HS	#NHS	#HS	#NHS	#HS	#NHS
DS-1	2740	40275	2615	37077	2615	2615
DS-2	1430	4163	1155	3475	1155	1155
DS-3	5935	7814	5647	7383	5647	5647

A. DATASETS

We employ three twitter-related datasets to examine the effectiveness of the presented model. The first dataset, DS-1 [33], consists of 80000 tweets, where each belongs to one of the four categories: *abusive*, *hateful*, *spam* or *normal*. The proposed approach is a binary classification problem, so we only consider hateful (T_H) and non-hateful (T_{NH}) tweets to construct the final dataset, consisting of 43, 015 tweets. The second dataset, DS-2 [21], consists of 24802 tweets, each labeled as either hateful, offensive, or neither. Like the first dataset, only the hate (T_H) and neither (T_{NH}) tweets are selected to construct the final evaluation dataset consisting of 5593 tweets to evaluate the proposed model. The third dataset, DS-3 [34], is a recent benchmark dataset consisting of 20148 tweets, where each is *hateful*, *offensive*, *normal*, or *undecided*. Like the first two datasets, we again select only the hateful (T_H) and normal (T_{NH}) tweets to create the final dataset, consisting of 13749 tweets. Table 1 presents the statistics of evaluation datasets, wherein the second column shows the statistics of *raw dataset*, constructed after the selection of T_H and T_{NH} categories of tweets. The pre-processing step filters the duplicate tweets, URLs, alphanumeric characters, and other noisy content. The first sub-column of the third column shows the final statistic of the resultant pre-processed datasets. The provided datasets are imbalanced as the number of normal tweets is many times higher than the hateful class T_H tweets. To investigate the efficacy of the proposed model over balanced datasets, we create a balanced version of each dataset, wherein normal tweets are down-sampled equal to the size of the hateful tweets. In this process, we randomly select the normal tweets.

TABLE 2. Hyper-parameters of the proposed model.

Hyper-parameter	Adjusted value
Word embedding	200
Max. sequence length	25
#CNN Filters	32
filter size	1, 2, 3
#cells in BiLSTM	128 and 256
Optimization method	Adam
Dropout value	0.4
batch size	16

B. EXPERIMENTAL SETTINGS

We perform all the experimental evaluations over Google Colab. We implement the model using Keras 2.7 in Python 3.7. In the proposed model, each of the first, second, and third channels have one convolution layer utilizing 32 filters of size 1, 2, and 3, respectively. Each channel has 2-layers stacked BiLSTM, wherein the first and second layers use 128 and 256 memory cells, respectively. The dropout value of 0.4 is used to avoid over-fitting. We analyze the efficacy of model training and evaluation using five-fold cross-validation, which ensures that every instance of the dataset is involved in the training and validation. During the experimentation, all the training has been conducted using 10 epochs employing *binary cross-entropy* and *Adam* as a loss function and optimization method, respectively. The experimental evaluation processes 16 instances at a time. Table 2 presents all the hyper-parameters and their underlying optimal values.

C. EVALUATION TECHNIQUES

In the existing literature, researchers generally evaluate the efficacy of a classification model using accuracy and fscore. Likewise, we analyzed our model using precision (Pr), recall (Rc), fscore (Fs), and accuracy (Acc). Pr is the percentage of accurately categorized positive class instances (hate tweets) from the set of all the tweets classified as the hateful tweet (true positives (TP) + false positives (FP)) as shown in equation 10. In this study, Rc is the portion of accurately classified hate tweets (true positives (TP)) out of total hateful tweets (true positives (TP) + false negatives (FN)), as defined in equation 11. Equation 12 computes the Fs as the harmonic mean of Pr and Rc. Finally, Acc is the portion of accurately classified tweets out of all input tweets (N_T), as defined in equation 13.

$$\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{10}$$

$$\mathrm{Rc} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{11}$$

$$\mathrm{Fs} = \frac{2 \times \mathrm{Pr} \times \mathrm{Rc}}{\mathrm{Pr} + \mathrm{Rc}} \tag{12}$$

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{N_T} \tag{13}$$

D. EXPERIMENTAL RESULTS

We evaluate the presented model on balanced and imbalanced categories of the datasets. We also establish our model by comparing it with ten methods, including five SOTA and five baseline approaches considering four discussed evaluation metrics. The first row of table 3 shows the underlying empirical results over the imbalanced datasets. Likewise, the first row of table 4 shows results on the balanced datasets. The comparison of experimental results from the two tables reveals that the presented model outperforms over imbalanced dataset for DS-1 and DS-2. However, it shows better results on the balanced version of DS-3. The better performance over the imbalanced dataset is significant because it is a real-life scenario. Interestingly, the result on the balanced version of DS-1 goes down significantly.

1) COMPARATIVE EVALUATION

The efficacy of our model is compared to five SOTA and five baseline models. The following paragraphs present a brief description of comparison models.

- Mossie and Wang [40]: In the paper, authors used two recurrent neural networks: GRU and LSTM layers, to detect hate speech. They also identified vulnerable communities on Facebook and Twitter.
- BiCHAT [28]: The authors presented a hierarchical attention-based deep model using a 6-layer deep convolutional network and Bidirectional LSTM for text representation learning. The model used BERT, a transformer-based SOTA language model, for input representation.
- Ding et al. [41]: The authors presented a hybrid deep learning model by integrating a stack of Bidirectional GRU with the capsule network for text representation learning for detecting hate speech in OSNs.
- Roy et al. [14]: This model presented an automated hate speech detection method employing a deep convolutional neural network using GloVe embedding.
- BERT [42]: It is the latest language model based on transformer architecture to encode the text.
- ANN: We also build an artificial neural network as a comparison baseline method. This simple ANN has 2 hidden layers having 64 and 16 neurons.
- CNN: The second baseline is a simple convolutional neural network to perform a comparison with the model. This CNN model contains 128 filters of size 3.
- LSTM: We also construct a simple LSTM network with 128 neurons as the third baseline to compare its performance with our model for detecting hate speech.
- BiLSTM: It is an advancement of the LSTM network, incorporating the context in left-to-right and right-to-left directions. We construct a simple BiLSTM network having 128 cells to compare its performance with our model.
- GRU: The fifth baseline for comparison is a GRU network having 128 neurons.

TABLE 3. Experimental results over the imbalanced datasets.

Datasets →	DS-1				DS-2				DS-3			
Methods ↓	**Acc**	**Pr**	**Rc**	**Fs**	**Acc**	**Pr**	**Rc**	**Fs**	**Acc**	**Pr**	**Rc**	**Fs**
Our model	**0.9341**	**0.9343**	**0.9382**	**0.9360**	0.9248	**0.9336**	**0.9232**	**0.9279**	**0.8611**	0.8618	**0.8263**	0.8415
Mossie and Wang [42]	0.7228	0.4149	0.2723	0.3336	0.8802	0.9112	0.8907	0.9045	0.7881	0.7753	0.7857	0.7536
BiCHAT [30]	0.8919	0.8854	0.8012	0.8439	0.7356	0.7417	0.7555	0.7534	0.7745	0.7976	0.7732	0.7484
Ding et al. [43]	0.5923	0.3543	0.3916	0.3738	0.7803	0.3315	0.2729	0.3005	0.8069	0.8169	0.7202	0.7570
Roy et al. [14]	0.7843	0.2609	0.1218	0.1613	0.7737	0.8914	0.6636	0.7625	0.5972	0.5722	0.3324	0.3185
BERT [44]	0.9257	0.9306	0.8971	0.9128	**0.9414**	0.9290	0.9118	0.9243	0.8581	**0.8847**	0.8147	**0.8474**
DNN	0.7543	0.1817	0.1131	0.1425	0.9104	0.8931	0.7529	0.8128	0.8291	0.8470	0.7355	0.7843
CNN	0.7621	0.3427	0.3818	0.3622	0.8831	0.8521	0.7123	0.7719	0.8279	0.8451	0.7350	0.7840
LSTM	0.7723	0.4012	0.1621	0.2332	0.8926	0.9332	0.8887	0.9035	0.8194	0.8030	0.7733	0.7849
BiLSTM	0.7832	0.5943	0.4028	0.4822	0.8843	0.9206	0.8923	0.9017	0.8180	0.8233	0.7904	0.8049
GRU	0.7527	0.4843	0.3333	0.3936	0.8744	0.9321	0.9012	0.9111	0.8217	0.8264	0.7915	0.8068

TABLE 4. Experimental results over the balanced datasets.

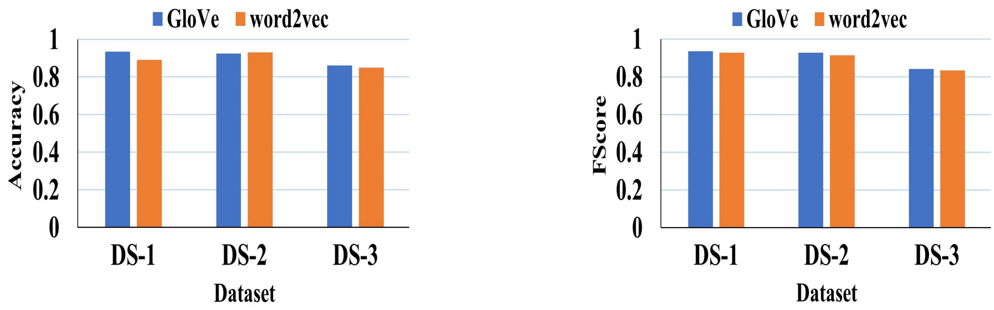
Datasets →	DS-1				DS-2				DS-3			
Methods ↓	**Acc**	**Pr**	**Rc**	**Fs**	**Acc**	**Pr**	**Rc**	**Fs**	**Acc**	**Pr**	**Rc**	**Fs**
Our model	**0.5420**	0.5376	0.5736	0.5382	**0.8940**	**0.9276**	0.9192	**0.9179**	0.8330	0.8843	**0.8488**	**0.8687**
Mossie and Wang [42]	0.4444	**0.8126**	**0.7942**	**0.8036**	0.8630	0.9223	0.9143	0.9132	0.8221	0.8130	0.8186	0.8127
BiCHAT [30]	0.4954	0.6065	0.6412	0.6233	7551	0.8054	0.7639	0.7841	0.7881	0.7914	0.7976	0.7949
Ding et al. [43]	0.4747	0.1143	0.2223	0.1535	0.6449	0.4621	0.5226	0.4923	0.7878	0.8105	0.7612	0.7799
Roy et al. [14]	0.4726	0.5243	0.5023	0.5132	0.6432	0.8821	0.8844	0.8832	0.6660	0.7627	0.5948	0.6073
BERT [44]	0.5092	0.4179	0.3104	0.3491	0.8852	0.9060	0.8148	0.8509	**0.8702**	**0.8860**	0.8418	0.8630
DNN	0.4846	0.6123	0.5827	0.5925	0.8543	0.8917	0.8521	0.8719	0.8146	0.8494	0.7691	0.8055
CNN	0.4731	0.6331	0.6135	0.6231	0.8725	0.8533	0.8444	0.8441	0.8126	0.8411	0.7745	0.8048
LSTM	0.4832	0.7132	0.7121	0.7135	0.8733	0.9221	0.9114	0.9116	0.8107	0.8266	0.7901	0.8062
BiLSTM	0.4706	0.8044	0.7936	0.7941	0.8834	0.9103	0.9021	0.9015	0.8085	0.8233	0.7904	0.8049
GRU	0.4849	0.7432	0.7333	0.7331	0.8635	0.9042	**0.9202**	0.9121	0.8107	0.8264	0.7915	0.8068

The experimental results of the comparison models over the balanced and imbalanced dataset are given in Tables 4 and 3, respectively. We present the comparative evaluation considering accuracy, precision, recall, and fscore. The best performance considering an evaluation metric among all the models is in bold typeface. Table 3 results show that our model beat all the comparison approaches except in three instances where BERT performs best. BERT also demonstrates comparative performance over DS-3. Table 3 also exhibits that BERT and BiCHAT models show relatively good performance among the SOTA models, whereas BiLSTM performs best among baseline models. By contrast, table 4 demonstrates that our model performs poorly on the balanced dataset. The table also indicates that the presented model performs best in 6 instances, whereas comparison approaches perform best in 6 cases. Over DS-1, the proposed and comparison models exhibit significantly poor performance. BERT also demonstrates the best results in terms of Acc and Pr over DS-3. Like the imbalanced dataset, BERT is again one of the best performers among SOTA models. However, Mossie and Wang show the best performance among the comparison approaches over the balanced dataset. The baseline models also demonstrate comparative performance. The presented model is comparable with the comparison methods [14], [40] considering the complexity. On the other hand, the comparison methods [28], [41] are more complex than the presented model.

The investigation of comparison results from tables 3 and 4 indicates that our model shows the best performance. Additionally, the comparative performance by BERT also establishes the strength of transformer-based large language models. The presented model shows better performance on the imbalanced dataset, which is encouraging because real-world datasets are imbalanced. Except for BERT, the proposed model significantly beats the comparison models with the highest performance difference of 10%. Interestingly, baseline models show good performance, which shows the strength of fundamental neural network components.

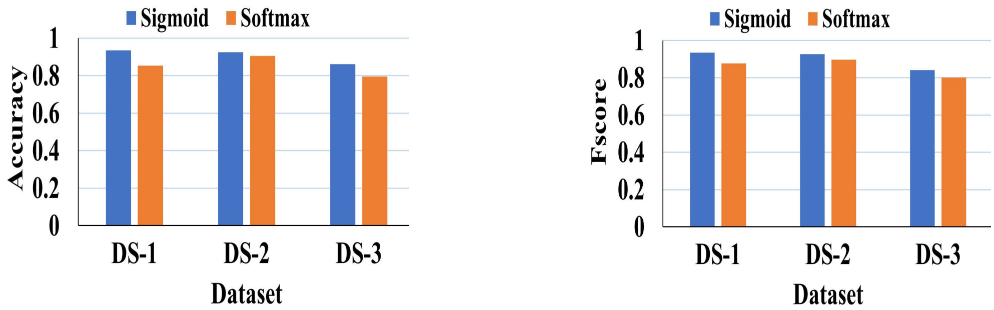
2) ABLATION ANALYSIS

This section performs an ablation study to analyze the impact of different neural network components used in the model. The ablation analysis is performed only over the imbalance dataset because it is the real dataset without any sampling. The presented model consists of three convolutional channels, stacked BiLSTM, and the attention mechanism. In the ablation study, we exclude a component from the model to examine its impact on the performance of the presented model. In this direction, we first exclude the second and third channels. The resultant updated model has only one channel with kernel size = 1. We execute the updated model to examine the impact of excluded channels on the model efficacy. Also, we remove the Bidirectional LSTM network to construct an updated model having all three channels excluding the BiLSTM network. The updated model is investigated to analyze the BiLSTM exclusion effect. The same procedure is repeated for the attention layer. Table 5 presents the evaluation results for the ablation study. The investigation of results shows that the exclusion of channels has the highest impact on DS-1 and DS-3 datasets. The exclusion of the attention layer shows a moderate impact but highest over the DS-2 dataset. On the other hand, the exclusion of the two BiLSTM layers shows the least impact and even improves the performance of the updated model in some instances. The highest impact for exclusion of the



(a) (b)

FIGURE 3. Experimental results of the proposed model employing GloVe and word2vec embeddings considering (a) accuracy (b) fscore.



(a) (b)

FIGURE 4. Experimental results of proposed model employing *sigmoid* and *softmax* considering (a) accuracy (b) fscore.

channels establishes the efficacy of the presented and justifies the inclusion of multiple channels in our model towards the hate speech classification.

V. INVESTIGATION OF HYPERPARAMETERS EFFECT

Deep learning models include various important user-controlled hyper-parameters, which impact their performance. Among these hyperparameters, embedding method, batch size, optimization algorithms, and activation function are the most important ones. This section conducts the empirical evaluation over the three datasets considering accuracy and fscore to observe the impact of the discussed hyperparameters on the model performance. We analyze the imbalanced dataset (original dataset) because it resembles more of a real-world scenario.

A. EMBEDDING METHOD

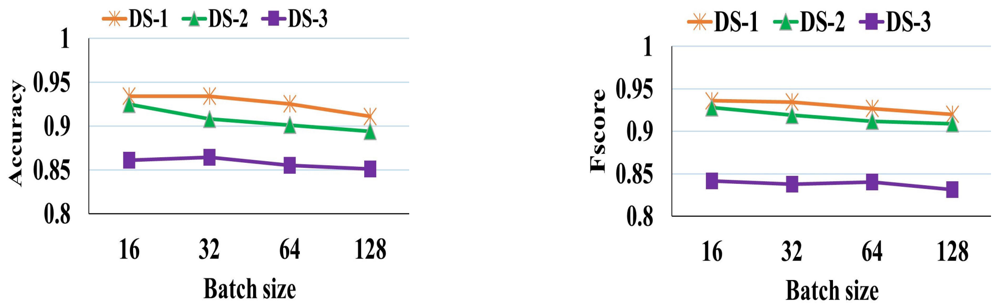
The dense numeric representation of content called embedding revolutionized the deep neural network field. The dense numeric vectors encode the input textual content in a deep learning model. GloVe [35], word2vec [43], BERT [42] are the state-of-the-art language models to encode a textual token(word/phrase/document) by a dense numeric vector. This dense numeric representation incorporates the contextual semantics of words in the training corpus. The existing language models differ considering dimension, contextual enrichment, and out-of-vocabulary word representation. Due to these characteristics, the performance of deep learning models using these language models differ. This study uses GloVe-based distributional representation at the embedding layer. Therefore, we analyze the efficacy of GloVe and word2vec language models on our model performance. Figure 3 presents the underlying results employing the Glove and word2vec embeddings considering accuracy and fscore. It depicts that model outperforms on all three datasets while using GloVe embedding, and its performance degrades when using word2vec input representation. However, in one instance, the accuracy of the presented model improves over DS-2. Figure also indicates that the impact of word2vec is most significant over DS-1 and least over DS-3. Therefore, the empirical analysis justifies the selection of GloVe embedding for the representation of input textual content at the embedding layer.

B. ACTIVATION FUNCTIONS

It determines the activation of a neuron. In other words, it computes the output of a node (neuron) based on the given input. The existing literature has various activation functions like sigmoid, softmax, and tanh to compute this output. However, only softmax and sigmoid are used at the output layer in a classification problem. Therefore, we investigate the impact of sigmoid and softmax activation functions on our model in detecting hate speech. The underlying empirical results over the three datasets are shown in figure 4. It demonstrates that the proposed model outperforms using

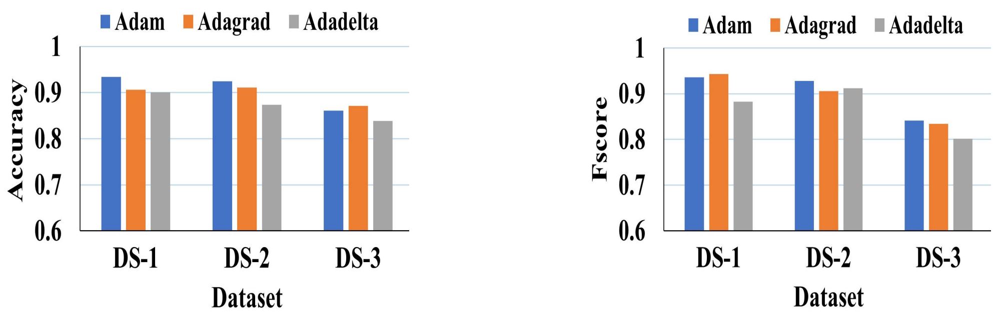
TABLE 5. Experimental results for ablation study over the imbalanced version of the three datasets.

Datasets →	DS-1				DS-2				DS-3			
Methods ↓	Acc	Pre	Rec	Fsc	Acc	Pre	Rec	Fsc	Acc	Pre	Rec	Fsc
Our model	0.9341	0.9343	0.9382	0.9360	0.9248	0.9336	0.9232	0.9279	0.8611	0.8618	0.8263	0.8415
Our model - I & II Channels	0.9050	0.9081	0.9125	0.9110	0.9100	0.9132	0.9020	0.9080	0.8081	0.8318	0.8023	0.8175
Our model - BiLSTM Layer	0.9310	0.9291	0.9321	0.9315	0.9291	0.9321	0.9275	0.9303	0.8482	0.8571	0.8181	0.8367
Our model - Attention Layer	0.9162	0.9193	0.9154	0.9170	0.9067	0.9106	0.9034	0.9063	0.8622	0.8528	0.8223	0.8379



(a) (b)

FIGURE 5. Experimental results of the proposed model using various batch sizes considering (a) accuracy (b) fscore.



(a) (b)

FIGURE 6. Experimental results of the proposed model using *Adam*, *Adagrad* and *Adadelta* considering (a) accuracy (b) fscore.

sigmoid, considering both accuracy and fscore. It is because sigmoid is for binary-class problems, while this study also models hate speech detection as a binary-class problem. Therefore, the proposed model also uses a sigmoid at the output layer due to the superior performance using a sigmoid.

C. BATCH SIZE

The *batch size* in a deep learning model is the number of training samples given to the model at a time. It also affects the quality and training speed of the model. The larger batch size trains a model fast but adversely affects its performance. Assume the training dataset has 100 samples and batch size is 10, then the model will process 10 samples at a time. As a result, the 100 data points will be processed in 10 batches. To investigate the impact of batch size, we experiment using four different batch sizes: 16, 32, 64, and 128. The underlying experimental results using four batch sizes over three datasets are shown using the line graph 5. The line graph demonstrates that the model performance degrades as the batch size increases. However, there is no impact or adverse effect in certain instances. The analysis of results shows that the impact of batch size is higher considering accuracy than fscore. Figure also exhibits that the impact is higher over DS-2 than DS-1 and DS-3 datasets. Thus, the experimental results ascertain the processing of 16 instances by the model at a time.

D. OPTIMIZATION ALGORITHMS

It is another principal parameter that affects the efficacy of a deep neural network. An optimization algorithm updates the network parameters during the training. We analyze the impact of Adam, Adagrad, and Adadelta on the model performance. Figure 6 shows the experimental results of underlying evaluation considering accuracy and fscore over the three datasets. It exhibits that the model performs best using the Adam optimization algorithm and worst with the Adadelta algorithm. However, the proposed model performs best in two instances, employing the Adagrad algorithm. Figure also shows that the performance degrades higher considering accuracy than fscore. Therefore, the empirical analysis using different algorithms ascertains Adam as the best optimization in our model.

VI. CONCLUSION AND FUTURE DIRECTIONS OF WORK

In this study, we introduced a multi-channel attention-aware CNN-BiLSTM model for representation learning. In the multi-channel, the presented model has three parallel convolutional networks using filters with different kernel sizes. Further, the convolution-encoded representation is passed to a stacked BiLSTM network to encode sequential information, which is further passed to an attention layer to assign an importance score. We evaluated the model over the three twitter-related benchmark datasets considering precision, recall, fscore, and accuracy. The comparative evaluation establishes the efficacy of the proposed model over the five state-of-the-art and five baseline models. We also performed the ablation analysis to examine the contribution of various neural network components. The examination of the hyperparameter effect on the model performance ascertains the use of their optimal value in our model.

The proposed model has not used content, network, and profile-related features, which can be vital. Further, the experimental evaluation of the proposed model over multilingual text is another dimension of further research. The comparison with various transformer-based language models is another good future direction.

REFERENCES

[1] M. Fazil and M. Abulaish, "A hybrid approach for detecting automated spammers in Twitter," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2707–2719, Nov. 2018.

[2] M. Fazil and M. Abulaish, "Why a socialbot is effective in Twitter? A statistical insight," in *Proc. 9th Int. Conf. Commun. Syst. Netw.*, Bengaluru, India, Jan. 2017, pp. 562–567.

[3] M. Abulaish, S. Sharma, and M. Fazil, "A multi-attributed graph-based approach for text data modeling and event detection in Twitter," in *Proc. 11th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Bengaluru, India, Jan. 2019, pp. 703–708.

[4] M. Fazil, A. K. Sah, and M. Abulaish, "DeepSBD: A deep neural network model with attention mechanism for SocialBot detection," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4211–4223, 2021.

[5] M. Abulaish and M. Fazil, "Socialbots: Impacts, threat-dimensions, and defense challenges," *IEEE Technol. Soc. Mag.*, vol. 39, no. 3, pp. 52–61, Sep. 2020.

[6] F. Husain and O. Uzuner, "A survey of offensive language detection for the Arabic language," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 20, no. 1, pp. 1–44, Jan. 2021.

[7] W. Warner and J. Hirschberg, "Detecting hate speech on the world wide web," in *Proc. Workshop Lang. Social Media*. Montreal, QC, Canada: ACL, 2012, pp. 19–26.

[8] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter," in *Proc. NAACL Student Res. Workshop*, 2016, pp. 88–93.

[9] E. W. Pamungkas, V. Basile, and V. Patti, "A joint learning approach with knowledge injection for zero-shot cross-lingual hate speech detection," *Inf. Process. Manage.*, vol. 58, no. 4, pp. 1–19, 2021.

[10] S. Khan, A. Kamal, M. Fazil, M. A. Alshara, V. K. Sejwal, R. M. Alotaibi, A. R. Baig, and S. Alqahtani, "HCovBi-Caps: Hate speech detection using convolutional and bi-directional gated recurrent unit with capsule network," *IEEE Access*, vol. 10, pp. 7881–7894, 2022.

[11] A. U. Haq, J. P. Li, S. Ahmad, S. Khan, M. A. Alshara, and R. M. Alotaibi, "Diagnostic approach for accurate diagnosis of COVID-19 employing deep learning and transfer learning techniques through chest X-ray images clinical data in E-healthcare," *Sensors*, vol. 21, no. 24, pp. 1–16, 2021.

[12] Z. Zhang, D. Robinson, and J. Tepper, "Detecting hate speech on Twitter using a convolution-GRU based deep neural network," in *Proc. Eur. Semantic Web Conf.* Heraklion, Greece: Springer, 2018, pp. 745–760.

[13] A. U. Haq, J. P. Li, S. Khan, M. A. Alshara, R. M. Alotaibi, and C. Mawuli, "DACBT: Deep learning approach for classification of brain tumors using MRI data in IoT healthcare environment," *Sci. Rep.*, vol. 12, no. 1, pp. 1–14, Sep. 2022.

[14] P. K. Roy, A. K. Tripathy, T. K. Das, and X.-Z. Gao, "A framework for hate speech detection using deep convolutional neural network," *IEEE Access*, vol. 8, pp. 204951–204962, 2020.

[15] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," in *Proc. 26th Int. Conf. World Wide Web Companion*, 2017, pp. 759–760.

[16] H. Zhang, J. Wang, J. Zhang, and X. Zhang, "Using a multi-channel CNN-LSTM model for sentiment classification," in *Proc. SemEval*. Vancouver, BC, Canada: ACL, 2017, pp. 796–801.

[17] W. Li, Y. Xu, and G. Wang, "Stance detection of microblog text based on two-channel CNN-GRU fusion network," *IEEE Access*, vol. 7, pp. 145944–145952, 2019.

[18] I. Kwok and Y. Wang, "Locate the hate: Detecting tweets against blacks," in *Proc. 27th AAAI Conf. Artif. Intell.* Bellevue, WA, USA: AAAI, 2013, pp. 1621–1622.

[19] P. Burnap and M. L. Williams, "Cyber hate speech on Twitter: An application of machine classification and statistical modeling for policy and decision making," *Policy Internet*, vol. 7, no. 2, pp. 223–242, 2015.

[20] I. Alfina, R. Mulia, M. I. Fanany, and Y. Ekanata, "Hate speech detection in the Indonesian language: A dataset and preliminary study," in *Proc. Int. Conf. Adv. Comput. Sci. Inf. Syst. (ICACSIS)*, Yogyakarta, Indonesia, Oct. 2017, pp. 233–238.

[21] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Proc. ICWSM*. Montreal, QC, Canada: AAAI, 2017, pp. 512–515.

[22] S. Malmasi and M. Zampieri, "Detecting hate speech in social media," in *Recent Advances in Natural Language Processing*. Varna, Bulgaria: ACL, 2017, pp. 467–472.

[23] M. Abulaish, N. Kumari, M. Fazil, and B. Singh, "A graph-theoretic embedding-based approach for rumor detection in Twitter," in *Proc. WI*. Thessaloniki, Greece: ACM, 2019, pp. 466–470.

[24] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, "Hate speech detection with comment embeddings," in *Proc. 24th Int. Conf. World Wide Web*, Florence, Italy, May 2015, pp. 29–30.

[25] F. D. Vignal, A. Cimino, F. Orletta, M. Petrocchi, and M. Tesconi, "Hate me, hate me not: Hate speech detection on Facebook," in *Proc. 1st Italian Conf. Cybersecur.* Venice, Italy, 2017, pp. 86–95.

[26] J. H. Park and P. Fung, "One-step and two-step classification for abusive language detection on Twitter," in *Proc. 1st Workshop Abusive Lang.*, Vancouver, BC, Canada, 2017, pp. 41–45.

[27] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. 31st Int. Conf. Mach. Learn.*, vol. 32, Jan. 2014, pp. 1188–1196.

[28] S. Khan, M. Fazil, V. K. Sejwal, M. A. Alshara, R. M. Alotaibi, A. Kamal, and A. R. Baig, "BiCHAT: BiLSTM with deep CNN and hierarchical attention for hate speech detection," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 7, pp. 4335–4344, Jul. 2022.

[29] S. M. Rezaeinia, A. Ghodsi, and R. Rahmani, "Text classification based on multiple block convolutional highways," 2018, *arXiv:1807.09602*.

[30] C. Quan, L. Hua, X. Sun, and W. Bai, "Multichannel convolutional neural network for biological relation extraction," *BioMed Res. Int.*, vol. 2016, no. 1, pp. 1–11, 2016.

[31] J. Yoon and H. Kim, "Multichannel lexicon integrated CNN-BiLSTM models for sentiment analysis," in *Proc. ROCLING*, Taipei, Taiwan, 2017, pp. 244–253.

[32] T. E. Trueman and E. Cambria, "Comment toxicity detection via a multichannel convolutional bidirectional gated recurrent unit," *Neurocomputing*, vol. 441, pp. 272–278, Jun. 2021.

[33] A. M. Founta, C. Djouvas, D. Chatzakou, I. Leontiadis, J. Blackburn, G. Stringhini, A. Vakali, M. Sirivianos, and N. Kourtellis, "Large scale crowdsourcing and characterization of Twitter abusive behavior," in *Proc. 12th Int. AAAI Conf. Web Social Media*, Stanford, CA, USA, Jun. 2018, pp. 491–500.

[34] B. Mathew, P. Saha, S. M. Yimam, B. Chris, P. Goyal, and A. Mukherjee, ''HateXplain: A benchmark dataset for explainable hate speech detection,'' in *Proc. AAAI*, Feb. 2021, pp. 14867–14875.
[35] J. Pennington, R. Socher, and C. Manning, ''Glove: Global vectors for word representation,'' in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, 2014, pp. 1532–1543.
[36] W. Yin, K. Kann, M. Yu, and H. Schütze, ''Comparative study of CNN and RNN for natural language processing,'' 2017, *arXiv:1702.01923*.
[37] Y. Kim, ''Convolutional neural networks for sentence classification,'' in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, 2014, pp. 1746–1751.
[38] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, ''How to construct deep recurrent neural networks,'' in *Proc. ICLR*, Banff, AB, Canada, 2014, pp. 1–13.
[39] D. Bahdanau, K. Cho, and Y. Bengio, ''Neural machine translation by jointly learning to align and translate,'' in *Proc. ICLR*, San Diego, CA, USA, 2015, pp. 1–15.
[40] Z. Mossie and J.-H. Wang, ''Vulnerable community identification using hate speech detection on social media,'' *Inf. Process. Manage.*, vol. 57, no. 3, pp. 1–16, 2020.
[41] Y. Ding, X. Zhou, and X. Zhang, ''YNU_DYX at SemEval-2019 task 5: A stacked BiGRU model based on capsule network in detection of hate,'' in *Proc. 13th Int. Workshop Semantic Eval.*, Minneapolis, MN, USA, 2019, pp. 535–539.
[42] Y. Ding, X. Zhou, and X. Zhang, ''BERT: Pre-training of deep bidirectional transformers for language understanding,'' in *Proc. NAACL-HLT*, Minneapolis, MN, USA, 2019, pp. 4171–4186.
[43] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, ''Distributed representations of words and phrases and their compositionality,'' in *Proc. NIPS*, 2013, pp. 3111–3119.

MOHD FAZIL received the master's degree in computer science from Aligarh Muslim University, Aligarh, India, and the Ph.D. degree in computer science from Jamia Millia Islamia, New Delhi. He is currently working as a Postdoctoral Research Associate with the Centre for Transformative Learning, University of Limerick, Ireland. He has published more than 17 research articles, including two articles in the IEEE Transactions on Information Forensics and Security and one in *ACM Transactions on Knowledge Discovery from Data*. His research interests include data science, social computing, and data-driven cyber security.

SHAKIR KHAN (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in computer science, in 1999, 2005, and 2011, respectively. He is currently an Associate Professor with the College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia. He teaches bachelor's and master's degree courses at the College of Computer, IMSIU. He has around 15 years of teaching, research, and IT experience in India and Saudi Arabia. His research interests include big data, data science, data mining, machine learning, the Internet of Things (IoT), e-learning, artificial intelligence, emerging technology, open-source software, library automation, and mobile/web application. He has published many research papers in international journals and conferences in his research domain. He is a member of the International Association of Online Engineering (IAOE). He is a reviewer of many international journals.

BADER M. ALBAHLAL received the Ph.D. degree in information from Florida State University. He is currently an Assistant Professor with more than five years of experience in both private and academic sectors. His work focuses on technology adoption and how it can influence the stakeholders in public organizations. His research interests include data science, cybersecurity, and emerging technology adoption.

REEMIAH MUNEER ALOTAIBI received the B.Sc. degree from Imam Mohammad Ibn Saud Islamic University (IMSIU), Saudi Arabia, and the M.Sc. and Ph.D. degrees in information management from Leeds Beckett University, U.K. Moreover, she has completed some courses, such as professional development diploma in project management, with the Midlands Academy of Business and Technology, U.K. She is currently an Assistant Professor with the College of Computer Sciences and Information, IMSIU. She has authored articles published in scientific journals, book chapters, and conference materials. She has acted as a Reviewer of *International Journal of Civic Engagement and Social Change* (IJCESC; IGI Global).

TAMANNA SIDDIQUI is currently a Professor of computer science with Aligarh Muslim University, Aligarh, Uttar Pradesh, India. She has 23 years of rich national and international teaching experience, including 12 years in Jamia Hamdard, New Delhi, three years with the University of Dammam, Saudi Arabia, and eight years with Aligarh Muslim University. She has sound research knowledge and has supervised more than a dozen Ph.D. scholars in different fields of computer science. She has more than 75 publications in reputed journals and Scopus-indexed conferences. She has also published 13 books and two patents. She has contributed to many international journals and conferences as an editor, a member of the advisory board, a reviewer, a program committee member, and a keynote speaker. Her research interests include data mining, database, software engineering, cloud applications, big data analytics, and machine learning. She has been acknowledged by Wasit University, Iraq, and the University of Dammam, for her excellence in teaching and research. She was honored by Imam Mohammad Ibn Saud Islamic University as an International Researcher of a research project under the Global Research Partnership Program.

MOHD ASIF SHAH is currently working as an Associate Professor with Kebri Dehar University, Ethiopia. He has more than 50 publications which are indexed in Scopus and Web of Science. He has more than ten years of teaching experience.