IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 26 December 2023, accepted 13 February 2024, date of publication 19 February 2024, date of current version 23 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3367281

RESEARCH ARTICLE

Enhancing Hate Speech Detection in the Digital Age: A Novel Model Fusion Approach Leveraging a Comprehensive Dataset

WAQAS SHARIF[1], SAIMA ABDULLAH[1], SAMAN IFTIKHAR[2], (Member, IEEE), DANIAH AL-MADANI[2], AND SHAHZAD MUMTAZ[1,3]

[1]Faculty of Computing, The Islamia University of Bahawalpur, Bahawalpur 6300, Pakistan
[2]Faculty of Computer Studies, Arab Open University, Riyadh 11681, Saudi Arabia
[3]School of Natural and Computing Sciences, University of Aberdeen, Aberdeen, AB24 3FX Scotland, U.K.

Corresponding author: Waqas Sharif (waqas.sharif@iub.edu.pk)

This work was supported and funded by Arab Open University (AOU) research fund No. (AOUKSA-524008).

ABSTRACT In the era of digital communication, social media platforms have experienced exponential growth, becoming primary channels for information exchange. However, this surge has also amplified the rapid spread of hate speech, prompting extensive research efforts for effective mitigation. These efforts have prominently featured advanced natural language processing techniques, particularly emphasizing deep learning methods that have shown promising outcomes. This article presents a novel approach to address this pressing issue, combining a comprehensive dataset of 18 sources. It includes 0.45 million comments sourced from various digital platforms spanning different time frames. There were two models utilized to address the diversity in the data and leverage distinct strengths found within deep learning frameworks: CNN and BiLSTM with an attention mechanism. These models were tailored to handle specific subsets of the data, allowing for a more targeted approach. The unique outputs from both models were then fused into a unified model. This methodology outperformed recent models, showcasing enhanced generalization capabilities even when tested on the largest and most diverse dataset. Our model achieved an impressive accuracy of 89%, while maintaining a high precision of 0.88 and recall of 0.91.

INDEX TERMS Hate speech detection, deep learning, natural language processing, CNN, BiLSTM, model fusion.

I. INTRODUCTION

Hate speech refers to language or expression that attacks an individual or community based on characteristics such as race, caste, ethnicity, religion, gender, sexual orientation, nationality, etc., [1], [2], [3], is a growing concern in our increasingly digital world. Social media platforms, such as Twitter and Facebook, have become a breeding ground for its proliferation. These platforms enable individuals to express their opinions and engage in discussions, leveraging their extensive and diverse user base. However, they have also transformed into spaces where hate speech can spread rapidly, causing harm to society. Hate speech on social media platforms can manifest in various forms, such as posts, comments, and messages intended to intimidate, harass, or humiliate others. It's worth noting that these platforms have implemented significant measures to combat hate speech, including enforcing policies and utilizing machine learning algorithms to detect and remove abusive content. Nevertheless, the problem of hate speech on social media remains a significant challenge that requires ongoing attention and action.

Detecting hate speech manually on social media presents an enormous challenge due to the sheer volume of content generated. Hate speech can take subtle and diverse forms, making human detection without advanced algorithms excep-

The associate editor coordinating the review of this manuscript and approving it for publication was Maria Chiara Caschera.

tionally difficult. Relying solely on human moderators for precise and timely identification of hate speech is impractical, necessitating the use of advanced natural language processing (NLP) algorithms and machine learning models. The NLP community has recently made significant progress in developing hate speech identification systems, with machine learning and particularly deep learning techniques demonstrating superior effectiveness [1], [4], [5], [6], [7], [8]. Deep learning is particularly valuable in swiftly identifying hate speech, as it analyzes language and behavioral patterns linked to hate speech. Moreover, deep learning continuously improves over time by integrating new data, offering a sustainable solution.

Utilizing NLP techniques to identify hate speech on social media presents a crucial yet technically complex challenge. This complexity arises due to the nuances inherent in language, where hate speech may not always be expressed through explicit aggressive, offensive, profane, or derogatory terms. Conversely, the absence of such terms does not guarantee the absence of hate speech [9]. The task is further compounded by the diverse language use and contexts across different platforms, making the development of effective detection models a formidable task. The ever-evolving landscape of language and slang on social media adds layers of complexity to hate speech detection. Moreover, social media text often demonstrates high sparsity, featuring numerous elements with limited occurrences, including noisy components lacking useful information. This sparsity can impede the creation of precise models and lead to overfitting. Additionally, those propagating hate speech constantly seek new ways to evade detection, increasing the complexity of automatic detection [10]. Further complicating matters is the limited availability of data on social media due to the enforcement of hate speech codes of conduct [11]. This scarcity poses a significant hurdle for deep learning techniques, which rely on extensive labeled data for accurate model training.

The challenges inherent in detecting hate speech across social media platforms underscore the critical need for a robust and adaptable deep learning model. Traditionally, hate speech detection relied on limited datasets from specific platforms and time periods. Our innovative approach encompasses diverse data sources, enabling our model to learn language nuances and contextual variations adeptly. Leveraging extensive labeled data from multiple sources, this deep learning model confronts the intricacies of hate speech detection, effectively handling subtleties, variations, sparsity, and the adaptability of hate speech propagators.

This study tackles substantial challenges and makes significant contributions. A key contribution is a comprehensive dataset consolidating 18 diverse datasets, representing various social media platforms and different time spans, including platforms with varying word limits. With this model, our aim is to pioneer a more comprehensive and effective solution in combating hate speech, thereby fostering safer and more inclusive online communities.

Additionally, our study introduces a pioneering deep-learning model designed for high generalizability. This model effectively handles the diverse dataset, resulting in marked improvements in hate speech detection across multiple social media platforms.

II. LITERATURE REVIEW

In recent years, detecting hate speech in online text has become a significant focus in NLP research. Initially, studies relied on conventional machine learning algorithms like SVM, KNN, Random Forest, and Decision Tree, using various feature types (for example, syntactic, semantic, sentiment, and lexicon) to identify hate speech [2]. However, the rise of deep neural networks has prompted extensive exploration into their effectiveness for NLP-related problems [12]. Notably, Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) have emerged as prominent options and are frequently assessed for hate speech detection.

Researchers often choose different deep learning models tailored to the text's characteristics. For shorter texts where capturing detailed context matters less, CNNs have become popular due to their adeptness at grasping local patterns across various text classification tasks [13], [14], [15]. On the other hand, when dealing with longer text sequences that demand a better grasp of semantic features and context, RNNs like Long Short-Term Memory (LSTM) networks and Bidirectional LSTMs (BiLSTMs) shine [16], [17], [18]. These models efficiently capture contextual information and word dependencies, proving advantageous in tasks like sentiment analysis and document classification.

In the realm of hate speech detection, Warner et al. [19] conducted a seminal study concentrating on identifying anti-Semitic language as a form of hate speech. Alshalan and Al-Khalifa [20] delved into classifying Arabic hate tweets using CNNs, RNNs, and bidirectional encoder representations from transformers (BERT). Employing word2vec as embedding layers via the Continuous Bag of Words (CBOW) method, their findings revealed that BERT didn't perform well for this task, resulting in an approximate 10% drop in performance, while the CNN achieved an f-score of 0.79. Another notable exploration by Waseem and Hovey [21] targeted hate speech on Twitter, particularly racism and sexism. They investigated features, including user demographics, lexical usage, geographic information, and character *n*-grams. Their study emphasized that using character *n*-grams with a maximum length of four proved to be the most effective approach. Furthermore, integrating gender as an additional feature led to a slight improvement in the obtained results.

Vashistha and Zubiaga [7] examined six publicly available datasets to identify hate speech in English and Hindi text. They constructed a logistic regression-based model, incorporating Term Frequency - Inverse Document Frequency (TF-IDF) and Part-of-Speech (POS) features. This base

model's performance was compared with a hierarchical neural network, which utilized several CNN filters and the BiLSTM model. The base model achieved an accuracy rate of 85%, while the neural network attained an accuracy rate of 83%. In Khan et al. [22], a proposed neural network architecture called BiCHAT combines BERT-based embedding, BiLSTM, and deep CNN with a hierarchical attention mechanism. The attention layers will apply on word and sentence levels, allowing focus on the most important words and phrases in the text while ignoring irrelevant information. The proposed approach was evaluated on several popular Twitter hate speech datasets and performed better than the base model.

Modha et al. [23] proposed a real-time model to identify and visualize hate comments from Facebook and Twitter. This model can be used as a plugin tool in web browsers to monitor online hate speech effectively. Initially, the authors used traditional machine learning algorithms such as SVM and logistic regression as a baseline model. Subsequently, they experimented with more advanced models such as CNN, BiLSTM, and BERT transformers. The experimental results showed that the proposed models achieved an F1-score of 0.64 on the Facebook dataset and 0.58 on the Twitter dataset. Kapil and Ekbal [24] introduced a multi-task learning framework designed to identify multiple interconnected categories of hate speech, including offensive language, racism, and sexism. Multiple neural networks were developed, encompassing architectures such as CNNs, LSTM networks, and a combination of CNN and GRU. These networks were trained for both single-task and multi-task learning scenarios. The initial training of the models was carried out for individual classes, and subsequently, a shared neural network was developed to perform the combined classification task. Rodriguez-Sanchez et al. [25] conducted an experimental study to assess the effectiveness of deep learning, machine learning, and transformer learning approaches in detecting hate speech specifically in Spanish language text. The results indicated that the transformer approach outperformed the other methods, achieving the highest F1-score of 0.75 for hate classification.

Mossie and Wang [26] introduced a method targeting the recognition of vulnerable communities through hate speech detection techniques. They utilized word2vec word embedding and *n*-grams for feature extraction, followed by classification using machine learning and deep learning algorithms. Moreover, they expanded the hate word lexicon by integrating co-occurring word vectors with the highest similarity, enabling the identification of the target ethnic community based on matched hate words. Ameur et al. [27] presented a dataset of 10,828 Arabic tweets addressing hate speech related to COVID-19. They performed fundamental analyses using pre-trained models, highlighting the efficacy of these models in detecting hate speech and false information in the complex Arabic language context. Meanwhile, Khanday et al. [28] investigated hate speech detection on Twitter during the COVID-19 pandemic, employing various feature extraction methods such as TF/IDF, bag of words, and word length. Decision tree classifiers notably emerged as the most effective, achieving a remarkable 97% accuracy in hate speech detection.

Del et al. [29] introduced SocialHaterBert, a model tailored for hate speech identification in English and Spanish tweets, showcasing improvements over the earlier HaterBert model. Employing BertForSequenceClassification and 'BERT' for hate speech classification, the model demonstrated performance gains ranging from 3% to 27% compared to HaterBert. Additionally, the authors proposed a method to construct a hate speech user graph using user profile attributes, potentially enhancing hate speech detection in multilingual social media discussions. Furthermore, Fortuna et al. [30] conducted an extensive study using a dataset for hate speech, toxicity, abusive language, and offensive content classification. They experimented with various models, including BERT, ALBERT, fasttext, and SVM, trained on nine publicly available datasets, evaluating both intra-dataset and inter-dataset model performance to gauge their generalizability across different hate speech categories and datasets.

Overall, while progress has been made in detecting hate speech, many studies have mainly used small datasets from single platforms like Twitter, Facebook. Relying on these limited sources might affect how well these methods work in the real world, especially across different languages or platforms. To make these methods more reliable, future research should consider using more diverse and larger datasets from various sources.

III. DATASET DESCRIPTION

The dataset utilized in this study incorporates 18 distinct datasets sourced from various publications spanning recent years. The curation of this dataset was conducted by a team of researchers, primarily selecting datasets based on their relevance to the study of hate speech prevalent on the web [31]. Notably, this combined dataset represents a pioneering effort, as no prior research, to the best of our knowledge, has employed such an extensive compilation for hate speech classification tasks.

This comprehensive dataset integrates diverse sources, encompassing various digital media platforms like Twitter, Facebook and Stormfront. Capturing data from multiple social media platforms and across varying time periods, the dataset offers a rich spectrum of content. Rigorous preprocessing measures were implemented to maintain coherence and compatibility across this merged collection. Nonetheless, including data from various sources and temporal spans inherently poses challenges in any text classification endeavor.

These challenges manifest in the form of linguistic variations, tonal disparities, and contextual nuances, posing obstacles in creating classification models capable of effectively capturing and generalizing patterns across different sources and time frames. While enriching the dataset, this diversity also introduces complexities that demand sophisticated

TABLE 1. Description of datasets employed in this study.

Data Source	Web Forum	Sample Size	Hate Type
Waseem & Hovy (2016) [21]	Twitter	136, 052	Sexist & Racist
Davidson et al. (2017) [9]	Twitter	25000	Hate & Offensive comments
Gibert et al. (2018) [32]	Stormfront	9, 916	Racist
Zampieri et al. (2019) [33]	Twitter	14, 100	Offensive comments
Frenk (2019) [34]	Facebook	21, 936	Immigrants & LGBT
Basile et al. (2019) [35]	SemEval-2019 Task 5	13, 000	Immigrants & Woman
Ousidhoum (2019) [36]	Twitter	5, 647	Hate Comments
HASOC (2019) [37]	Twitter & Facebook	5, 852	Hate & Offensive comments
HASOC (2020) [38]	Twitter & Facebook	4, 522	Hate & Offensive comments
Gautam et al. (2020) [39]	Twitter	9, 973	Hate comments
Kaggle (2018) [40]	Twitter	49, 159	Sexist & Racist
Kaggle (2021) [41]	Twitter	18, 208	Cyberbullying
Kaggle (2020) [42]	-	153, 000	Hate comments
Kaggle (2021) [43]	Twitter	32, 000	Hate comments
Kaggle (2020) [44]	-	40, 624	Hate comments
Kaggle (2020) [45]	Twitter	25, 296	Hate & Offensive comments
Edwards et al. (2020) [46]	-	170, 019	Cyberbullying
Mendlay (2020) [47]	Twitter	16848	Racism & Sexism

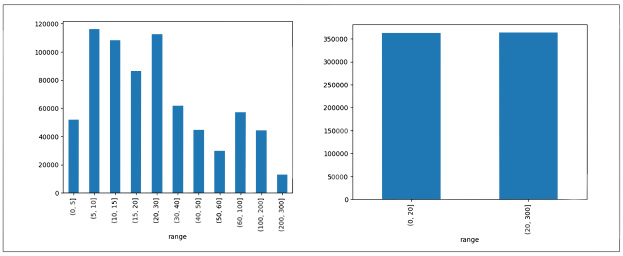

FIGURE 1. (a) Overall distribution of data (b) SSD vs LSD distribution of data.

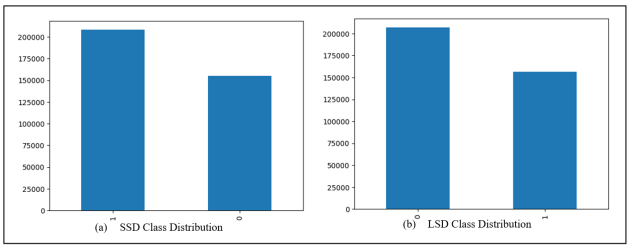

FIGURE 2. (a) SSD class distribution (b) LSD class distribution.

approaches to modeling and analysis. In Table 1, detailed descriptions of several representative subsets within the complete dataset, elucidating their significance as integral components of this expansive collection.

As observed from Table 1, the curated dataset represents a subset where various types of hate are targeted and may be categorized into multiple labels to distinguish different forms of hate. However, when these diverse datasets are combined into a unified collection, the labels are standardized so that any form of hate is classified as 'hate comment' while non-hate comments constitute the alternative category within this dataset.

The dataset, initially comprising 451, 709 English-language samples, was categorized into hate speech (371, 452) and non-hate speech (80, 257), reflecting an inherent class imbalance. To address this, the dataset underwent meticulous preprocessing, including tokenization, removal of stop words and symbols, and lemmatization. Following this, augmentation techniques were employed to rectify the class imbalance issue. Through these augmentation efforts, the final dataset expanded to 726,120 samples, achieving an equal class ratio and ensuring a more balanced representation of hate and non-hate speech categories for subsequent analysis.

During exploratory data analysis, it was noted that the dataset comprised sentences of varying lengths, reflecting the distinct writing styles associated with different web sources. This observation prompted the examination of the distribution of data based on the number of words per sample. As a result, two distinct sub-samples emerged: the Short Sequence Dataset (SSD), encompassing text up to 20 words, and the Long Sequence Dataset (LSD), containing longer text up to 300 words (depicted in Fig. 1). However, this division resulted in imbalanced subsets (Fig. 2), with the SSD skewed towards hate content and the LSD biased towards non-hate content. To rectify this imbalance, strategic application of the Synthetic Minority Over-sampling Technique (SMOTE) was performed on both sub-samples [48]. Through oversampling the minority class within each sub-sample, SMOTE effectively harmonized the class distributions, ensuring a more equitable representation of hate and non-hate content without relying on additional specific transformations.

Additionally, a word-ontology approach was utilized to manage the extensive vocabulary generated during the preprocessing stages. The dataset contained 127, 546 distinct words after preprocessing, presenting a challenge due to its substantial size and potential computational complexity in subsequent analyses. In this study, the WordNet ontology [49], [50] technique was employed to hierarchically organize words based on their semantic relationships and contextual meanings. This method categorized words into clusters or groups according to their similarities in meaning or usage, effectively consolidating redundant or closely related terms. Ultimately, the word-ontology technique significantly reduced the vocabulary size by up to 10.88%.

IV. MODEL FUSION FRAMEWORK

This section presents a detailed description of the model architecture proposed in this manuscript(see Fig. 3). The model includes word embedding layers, multiple CNN layers, a BiLSTM with attention mechanism layers, network merging layers, and a classification layer. Since our dataset contains sequences of varying lengths, short text sequences feed to the CNN model and long text sequences to the BiLSTM model followed by an attention layer. After reviewing the results of the proposed methodology, it was observe that the CNN model is particularly effective at

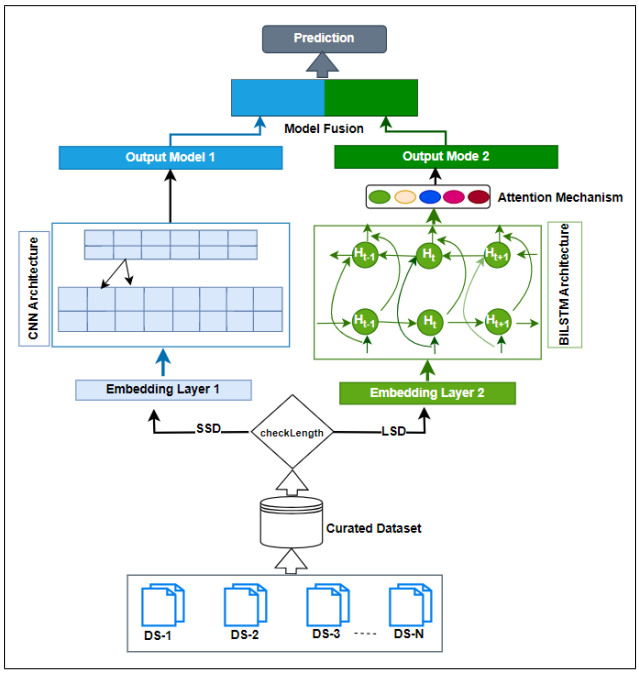

FIGURE 3. Block diagram of proposed study.

capturing targeted keywords. To a certain extent, these keywords directly determine the polarity of short text [51]. However, the CNN model may not be as effective with long text sequences because its convolutional layers operate over fixed-sized windows. As the text sequence grows longer, the fixed-sized window may not capture all relevant information [52], [53]. In such cases, the BiLSTM comes into play as it is better suited to handle longer sequences by being able to learn from the entire sequence and capture long-range dependencies between the words. Furthermore, an exploration was undertaken to utilize word ontology for reducing the feature/word count. Subsequently, detailed descriptions of the model's component structure are provided in the following sections.

A. EMBEDDING LAYER

Firstly, a word embedding layer is used to learn a dense vector representation of words from the preprocessed data. This layer takes the tokenized text as input and maps each word to a fixed-sized dense vector. During training, the vectors are learned, capturing the semantic relationships between words in the vocabulary. The input tokenized text can be represented as a sequence of n words: $w_1, w_2, w_3, \cdots, w_n$, where each word w_i is represented as a one-hot vector of vocabulary size v. The one-hot vector for a word has a value of 1 in the position corresponding to the index of the word in the vocabulary and 0 elsewhere.

The embedding layer has a matrix E of size $v \times d$ where d is the dimension of the dense word vectors to be learned. Each row of E contains the vector representation of a word in the vocabulary. To obtain the dense vector e_i for each word w_i, the word embedding layer performs a matrix multiplication: $e_i = w_i \times E$. This results in a sequence of dense vectors $E = e_1, e_2, e_3, \cdots, e_n$ of size $n \times d$. The word embedding matrix E is updated during training by minimizing a loss function with respect to the parameters of the model. This way, the word embedding layer learns to capture the semantic relationships between words in the vocabulary.

The embedding matrix E can then be fed as input to the subsequent layers of the neural network for further processing and classification. This work has created two distinct embedding layers: one for generating vectors for short text sequences of length 20, and the other for generating vectors for long text sequences of length 300.

Additionally, a pre-trained embedding layer using Global Vectors for Word Representation (GloVe) [54] with 50 dimensions has been included for comparison. This layer aims to offer word vector representations derived from existing knowledge, presenting an alternative perspective on word relationships within the text data. Both embedding layers, pre-train and trainable, contribute diverse perspectives in capturing and representing the underlying semantics within the text data, providing nuanced approaches for subsequent analysis and classification.

B. CNN ARCHITECTURE

Our CNN model integrates two convolutional layers and pooling layers to extract local features, with the objective of obtaining more informative keywords that enhance the overall performance of the model. The CNN architecture is described below.

1) FIRST CONVOLUTIONAL LAYER

This layer applies 128 filters of size 3 to the input sequence, producing 128 feature maps as output. Each filter slides over the input sequence, computing a dot product between the filter weights and the input at each position. The output of the convolution operation is then passed through a ReLU activation function (Eq. 1).

$$H_{(i,j)} = f(\sum_{k=0}^{k-1} W_k X_{i+j-k-1} + B) \tag{1}$$

where i and j denote the position of the output feature map, k denotes the filter index, f is the activation function, and K is the kernel size. The filter weights W are learned during training to capture meaningful patterns in the input data. The bias term B is added to each output feature map to introduce a shift in the activation function. The resulting output feature map contains a set of activation values representing the presence of different patterns in the input data.

2) FIRST POOLING LAYER

This layer performs max pooling on the output of the previous convolutional layer, reducing the spatial dimension by a factor of 2. Max pooling computes the maximum value within

each pooling window, which in this case has size 2(Eq. 2).

$$y_{i,j} = Max\left\{H_{(i,2j)}, H_{(i,2j+1)}\right\} \tag{2}$$

Here, $y_{(i,j)}$ is the *jth* output of the *ith* feature map after max pooling and $H_{(i,2j)}$ and $H_{(i,2j+1)}$ are the outputs of the previous convolutional layer at positions $2j$ and $2j+1$, respectively.

3) SECOND CONVOLUTIONAL LAYER

This layer applies 64 filters of size 3 to the output of the first pooling layer and producing 64 feature maps as output similar to the Eq. 1.

4) SECOND POOLING LAYER

iv. Second Pooling Layer: This layer performs max pooling over the entire spatial dimension of the output of the previous convolutional layer, resulting in a scalar value for each feature map.

$$Z_j = max\left\{y_{1,j},\ y_{2,j}, Y_{3,j}, \ldots\ldots, y_{n,j}\right\} \tag{3}$$

5) CNN OUTPUT LAYER

A fully connected dense layer with the ReLU activation function. It returns the maximum of 0 and the input value, which means that any negative values are set to 0.

$$y = f\left(W^T X + b\right) \tag{4}$$

where y and X is the output and input layer, W is a matrix of weights, b is a vector of biases and f is the activation function(ReLUx'), defined as $f\ (x) = \max\ (0,\ x)$.

C. BiLSTM WITH ATTENTION LAYER ARCHITECTURE

The BiLSTM model allows learning representations from both the forward and backward directions. The attention mechanism then weights the learned representations based on their importance in the context of the input sequence. Finally, the weighted representations are fed to the output layer for prediction.

1) BiLSTM LAYER

The BiLSTM model concatenates the output of the forward and backward LSTM cells at each time step, producing a sequence of hidden states $h = \{h_1, h_2, \ldots, h_T\}$ where T is the length of the input sequence. The forward LSTM computes the hidden state sequence $\bar{H}_f$ for each time step t using the input sequence (X_t), the previous cell state $\bar{H}_{t-1}$ and the hidden state H_t.

$$\bar{H}_f\ (X_t) = LSTM_f\left(H_t, \bar{H}_{t-1}\right) \tag{5}$$

Similarly, the backward LSTM computes the hidden state sequence h_b for each time step t using the input sequence (X_t) and the next cell state $\bar{H}_{bt+1}$ expressed in Eq.6.

$$\bar{H}_b\ (X_t) = LSTM_b\left(X_t - \bar{H}_{bt+1}\right) \tag{6}$$

Finally, the concatenated output of the forward and backward LSTM layers is given by Eq. 7, where $\bar{H}_f$ *and* $\bar{H}_b$ are the hidden state sequences computed by the forward and backward LSTMs, respectively.

$$h = \left[\bar{H}_f \quad \bar{H}_b\right] \tag{7}$$

2) ATTENTION LAYER

The objective of using attention layers is to enable the model to focus on the most important parts of the input sequence while ignoring the irrelevant parts. The architecture of the proposed model incorporates an additive attention layer, which takes input as the output from the previous BiLSTM layer, which is a sequence of hidden states. The attention layer then computes a set of attention weights for each hidden state in the sequence. These weights indicate the importance of each hidden state with respect to the current context and are computed using a dense layer with a sigmoid activation function, followed by a dot product operation between the resulting attention probabilities and the hidden states.

More specifically, a dense layer with a sigmoid activation function generates attention probabilities for each hidden state in the BiLSTM layer. These probabilities are then multiplied with their corresponding hidden states and summed up to get the context vector, representing the input sequence's most important parts.

$$h_i = tanh(W_a[h_{i-1}\ ;\ h_{i+1}] + b_a) \tag{8}$$

Here, h_i is the hidden state of the BiLSTM at time step i, W_a is the weight matrix of the attention layer, and b_a is the bias vector of the attention layer. The concatenation of the hidden states of the BiLSTM at time step $i-1$ and $i+1$ is represented as $h_{i-1}; h_{i+1}$.

The energy score (e_i) of the attention layer for time step i is represented as e_i, which is computed as the dot product of the weight vector v_a and the hidden state h_i.

$$e_i = v_a^T\ .\ h_i \tag{9}$$

The attention weight (probabilities) assigned to the hidden state at time step i is given by a_i, which is computed as SoftMax (e_i).

$$a_i = softmax\ (e_i) \tag{10}$$

Lastly, the final context vector c uses the attention weights a_i to combine the hidden states h_i selectively. This context vector c encapsulates information from the input sequence elements based on their relevance or importance determined by the attention mechanism.

$$c = \sum_{i=1}^{n} a_i\ \times h_i \tag{11}$$

D. MODEL FUSION LAYER

The merging network layer in our proposed model takes advantage of both CNNs and BiLSTM networks by combining their outputs to create a new, more powerful model. The output of each neural network is a vector representation of the input text. To combine information from two models,

TABLE 2. Hyper-parameters configurations.

Hyper-parameter	Value
Embedding dimension (CNN & BiLSTM)	50
CNN layer 1 filter size	128
CNN number of filters	3
CNN layer 2 filter size	64
CNN number of filters	2
BiLSTM number of neurons	32
Dropout	0.5
Batch size	512
Learning rate	0.01
Optimizer	*Adam*

an element-wise addition operation is performed between the output tensors of the models. Let's assume that the output tensor from the CNN model is denoted as y_{CNN} and the output tensor from the BiLSTM model is denoted as y_{BiLSTM}. The element-wise addition operation between can be represented as $M = y_{CNN} \oplus y_{BiLSTM}$ where $\oplus$ denotes the element-wise addition operation. This operation involves adding the corresponding elements of both models to obtain the corresponding elements in M. The resulting tensor M represents the combined information from both models, which is then fed into a final classification layer. This layer is a dense layer with a sigmoid activation function, which maps the input tensor to a probability distribution over the hate speech or not.

V. EXPERIMENTAL CONFIGURATIONS

The experiments conducted in this study were executed on Google Colab, utilizing a standard GPU and the Python programming language. The SSD and LSD datasets were partitioned into three subsets - for training, testing, and validation by employing the *'train_test_split'* method. The training dataset comprised 80% of the total data, while the remaining 20% was equally divided between testing and validation. Both models were equipped with an embedding layer with an output dimension of 50 and multiple dropout layers implemented to prevent overfitting. Additionally, early stopping with patience of 5 was employed to mitigate overfitting risks. The models were trained for 50 epochs using the Adam optimizer [55], featuring a learning rate set at 0.01 and a batch size of 512. Throughout the training process, binary cross-entropy was utilized to compute the validation loss of our models. These parameter settings were chosen based on empirical experimentation, resulting in high accuracy for our classification task. Detailed parameter information is provided in Table 2.

To assess the performance of the models, accuracy, precision, recall, and f-score were employed as the evaluation metrics. The accuracy 12 is the ratio of the correctly classified samples to the total number of samples. Accuracy alone may not be sufficient to evaluate the model's performance, especially when the focus is on a particular target class. Therefore, precision 13, recall 14, and f-score 15 are important metrics to consider as they provide insight into how well the model can correctly identify hate content. Precision measures the frequency of correct identification of hate content by the model, while recall measures how well the model can detect hate speech. F-score combines precision and recall to give a balanced measure of the model's performance.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

$$Percision = \frac{TP}{TP + FP} \tag{13}$$

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

$$F - score = \frac{2 \times (Percision \times Recall)}{Percision + Recall} \tag{15}$$

VI. RESULTS AND DISCUSSIONS

After establishing the experimental configurations for CNN and BiLSTM models across varying data lengths and settings, the subsequent focus shifted to analyzing the results obtained from these comprehensive evaluations. Each model (CNN and BiLSTM) was tested under four distinct settings, encompassing combinations of pre-trained and trainable embedding layers, with and without the integration of word ontology. Initially, both the CNN and BiLSTM models underwent testing on the complete dataset, followed by subsequent evaluations where the CNN model processed SSD and the BiLSTM model handled LSD. This process produced four distinct outcomes for each dataset type, providing a comprehensive understanding of the models' performance variations. Finally, a fusion model emerged, integrating short sequence data into the CNN and employing long sequence data within the BiLSTM model alongside an attention mechanism. The results of the evaluation metrics, including accuracy, precision and recall, are presented in the following Table 3, providing insight into the effectiveness of our model in identifying instances of hate speech.

As one can observe from the presented results, there are notable variations in performance among the models and their respective configurations. When CNN and BiLSTM operated independently on the complete dataset (LSD + SSD), their accuracies ranged moderately between $80 - 88\%$. Specifically, in terms of identifying hate speech, CNN achieved precision rates between $75 - 87\%$, while BiLSTM exhibited precision rates from $77 - 87\%$ for the hate class.

However, a significant shift occurred when these models were separately trained on SSD and LSD. The precision for hate speech notably improved by around $5 - 6\%$ for both CNN and BiLSTM when tailored to their respective sequence lengths. This showcased the effectiveness of a data-driven strategy, highlighting CNN's suitability for shorter texts and BiLSTM effectiveness for longer ones. These findings led

TABLE 3. Performance evaluation of different model combinations of the proposed study.

Model	Ontology	Embedding	Accuracy	Precision (hate)	Precision (not-hate)	Recall (hate)	Recall (not-hate)
$CNN_{SSD+LSD}$	✓	Pre-train	80	0.75	0.87	0.90	0.71
$CNN_{SSD+LSD}$	✗	Pre-train	81	0.81	0.82	0.82	0.80
$CNN_{SSD+LSD}$	✓	trainable	87	0.84	0.92	0.93	0.82
$CNN_{SSD+LSD}$	✗	trainable	84	0.86	0.82	0.87	0.81
CNN_{SSD}	✓	Pre-train	80	0.81	0.77	0.82	0.76
CNN_{SSD}	✗	Pre-train	79	0.80	0.78	0.85	0.71
CNN_{SSD}	✓	trainable	84	0.85	0.82	0.87	0.80
CNN_{SSD}	✗	trainable	**88**	**0.85**	**0.92**	**0.92**	**0.83**
BiLSTM + $Attention_{SSD+LSD}$	✓	pre-train	80	0.77	0.83	0.85	0.74
BiLSTM + $Attention_{SSD+LSD}$	✗	pre-train	83	0.80	0.87	0.89	0.77
BiLSTM + $Attention_{SSD+LSD}$	✓	trainable	87	0.85	0.90	0.91	0.84
BiLSTM + $Attention_{SSD+LSD}$	✗	trainable	88	0.87	0.89	0.89	0.87
BiLSTM + $Attention_{LSD}$	✓	pre-train	86	0.84	0.88	0.84	0.87
BiLSTM + $Attention_{LSD}$	✗	pre-train	86	0.84	0.88	0.84	0.84
BiLSTM + $Attention_{LSD}$	✓	trainable	90	0.87	0.93	0.91	0.90
BiLSTM + $Attention_{LSD}$	✗	trainable	**92**	**0.90**	**0.94**	**0.94**	**0.90**
Model Fusion (CNN, BiLSTM + $Attention)_{SSD+LSD}$	✓	pre-train	81	0.80	0.83	0.84	0.79
Model Fusion (CNN, BiLSTM + $Attention)_{SSD+LSD}$	✗	pre-train	81	0.80	0.83	0.84	0.79
Model Fusion (CNN, BiLSTM + $Attention)_{SSD+LSD}$	✓	trainable	88	0.86	0.90	0.91	0.85
Model Fusion (CNN, BiLSTM + $Attention)_{SSD+LSD}$	✗	trainable	**89**	**0.88**	**0.90**	**0.91**	**0.88**

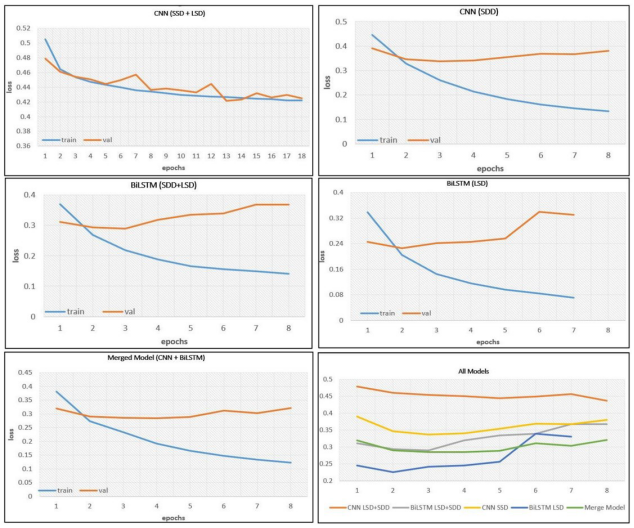

FIGURE 4. Validation and training loss over different epochs.

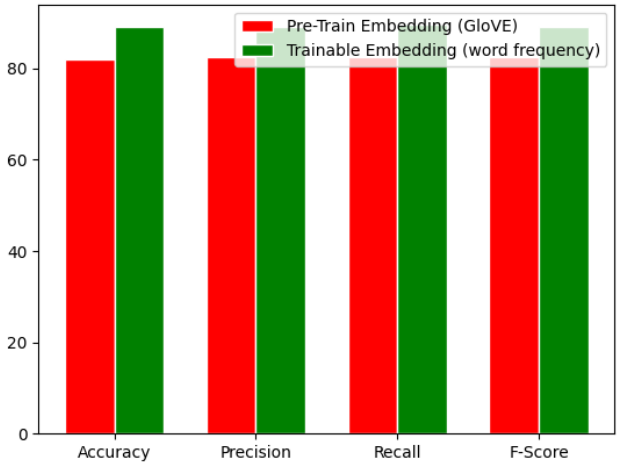

FIGURE 5. Performance comparison of embedding techniques.

to adopting a combined approach, leveraging CNN and BiLSTM into a unified architecture.

The resultant unified model not only maintained a high accuracy of $88-89\%$ but also showcased an improvement in precision for identifying hate speech by approximately $6-8\%$ compared to the individual performances on complete data. This underlines the synergy achieved by integrating their strengths, demonstrating a more comprehensive understanding and adeptness in identifying hate speech content. Figure 4 displays the training and validation loss graphs for the selected models and their comparative analysis. In Figure 4 (f), the validation loss comparison among CNN, BiLSTM, and the merged model reveals that CNN's validation loss is higher than BiLSTM. The merged model's validation loss falls between the two, aligning with expectations due to their differing input sequence lengths.

Moreover, when comparing different ontology and pre-trained embedding capacity settings, it was observed that using trainable embedding led to a decrease in model accuracy by 6 to 7%. Figure 5 illustrate the comparison of both embedding capacity settings. On the other hand, employing word ontology had a minor effect, decreasing accuracy by only 1 to 2%.

The performance of the CNN model was also evaluated with different text lengths, including 10, 20, and 30 words, to determine the optimal length for the specific task. The

TABLE 4. Performance comparison with state-of-the-art techniques using a subset of our employed datasets.

Paper	Data Source	DataSet References	Performance
[56]	Twitter	SemEval-2019 task 5 [35]	Accuracy 75.30, F-Score 0.70
[57]	Stormfront, Twitter	Gamback & Skider [58], Alatawi [57]	Accuracy 84.00, F-Score 0.84
[59]	Twitter	Founta [60], Khan [59]	Accuracy 80.00, F-Score 0.76
[61]	Twitter, Web Forum	Davidson [9], SemEval-2019 task 5 [35], [62]	Accuracy 82.00, F-Score 0.78
[63]	Twitter	Gaikwad [63]	F-Score 0.72
[64]	Twitter	Davidson [9], SemEval-2019 task 5 [35], Waseem & Hovy [21], Waseem [65], Ousidhoum [36]	Accuracy 89.00, F-Score 0.87
[7]	Twitter	HASOC2019 – EN [37], Davidson [9], SemEval-2019 task 5 [35], ElSherif [66], Ousidhoum [36], PMathur [67]	Accuracy 83.00
[51]	Twitter	Davidson [9]	Accuracy 91.00, F-Score 0.80
[68]	Twitter	Waseem & Hovy [21]	F-Score 0.88
[69]	Twitter	Waseem & Hovy [21]	F-Score 0.74
[35]	Twitter	SemEval-2019 task 5 [35]	F-Score 0.65
Proposed	Curated Dataset (Twitter, FaceBook, Stormfront)	See table 1	Accuracy 89.00, F-Score 0.89

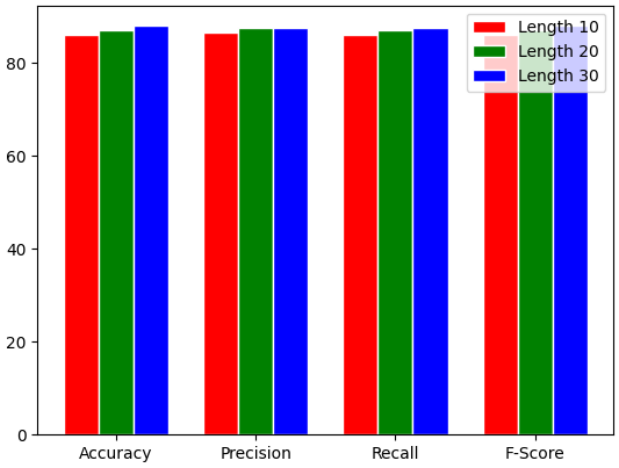

FIGURE 6. Comparison of CNN model performance with different word sequences.

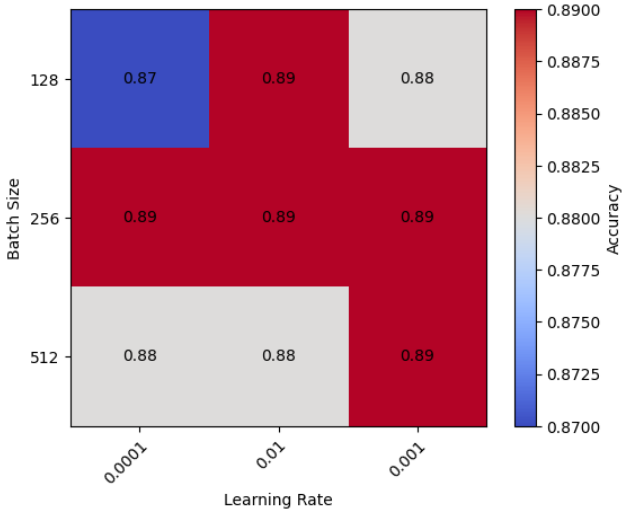

FIGURE 7. Impact of changing batch size and learning rate on the accuracy.

model's performance was similar for all three lengths, as shown in Figure 6. However, splitting the data by 20 words resulted in a balanced data distribution into two halves.

Further, for exploring the effectiveness of the proposed approach, a comparison was made with an existing state-of-the-art method. Table 4 compares the proposed fusion model with earlier research studies that utilized any subset of the dataset used in the current study. The comparison is made in terms of accuracy and F-score. Researchers sometimes presented their results separately for each dataset instead of combining them. For such cases, the average of their results was compared with our study. Moreover, in the case of multilingual data usage by an author, only the results for the English dataset were considered.

The comparative study presented highlights the superior performance of the proposed merged model over previous studies, even when using the largest dataset. These results suggest that the proposed model serves as a global model that can train on a large, diverse dataset and provide better predictions for the hate class.

In addition to the comparative study, further experiments were conducted to test the performance of the proposed model under different conditions. We also investigated the impact of different hyper-parameters, such as batch size and learning rate η, on the accuracy and training time of our model. We experimented with three values for each hyper-parameter (128, 256, and 512 for batch size and 0.01, 0.001, and 0.0001 for learning rate), resulting in nine combinations in total. These experiments aimed to assess the robustness and versatility of the proposed model in different settings.

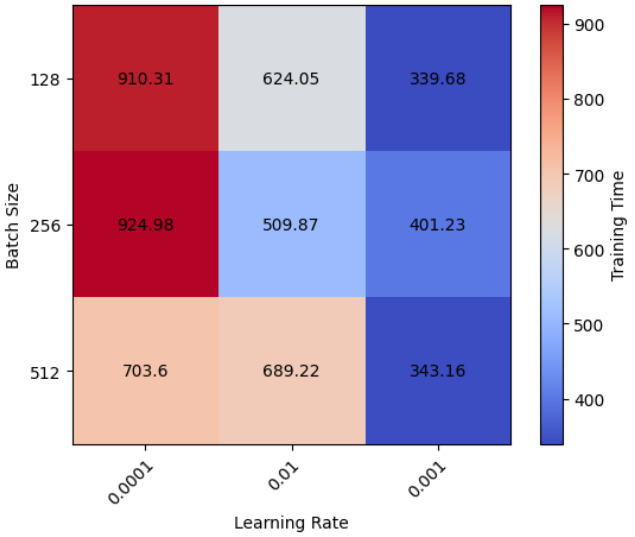

FIGURE 8. Impact of changing batch size and learning rate on the training time.

Figure 7 shows that the accuracy was almost invariant to the changes in these hyper-parameters, indicating that our model was robust and insensitive to them. However, significant variability in training time based on the batch size and learning rate was noted. Specifically, a decrease in batch size or an increase in learning rate resulted in longer training times (refer to Figure 8). This trade-off between time and stability was observed, yet it did not impact the model's performance. Furthermore, the selected embedding technique outperformed a pre-trained embedding

VII. CONCLUSION

The unprecedented growth of social media platforms in the digital age has introduced an alarming opportunity for the swift dissemination of hate speech, posing a significant threat to online discourse and community well-being. To address this pressing issue, our research presents a novel approach leveraging a comprehensive dataset comprising over 0.45 million comments from 18 diverse sources, encompassing various digital platforms across different time frames. Following thorough data preprocessing and balancing (by employment data augmentation), a comprehensive analysis revealed the presence of sentences ranging from 3 to 300 words in length. Recognizing the challenge of handling such variable-length text, the dataset divided into two distinct subsets based on sentence length—short sequence data (SSD) and long sequence data (LSD). Our approach leveraged previous research findings indicating that CNN performs exceptionally well in classifying short sequence text, and capturing local features effectively, while BiLSTM excels in understanding the context of long sentences. To harness these strengths, CNN models were trained for SSD and BiLSTM models for LSD. Acknowledging the potential for very long sequences in the LSD subset, an attention mechanism was introduced to focus on the most relevant areas within sentences, thereby enhancing the BiLSTM's performance. After training each model individually, a model fusion approach was employed to combine their outputs, resulting in a unified model.

Notably, the effectiveness of proposed ensemble approach is underscored by the results. Employing CNN for the entire dataset yielded an accuracy of 81% with an F-score of 0.82 for hate class detection. However, when CNN was exclusively applied to short text samples, the accuracy soared to 88% with an F-score of 0.88. Similarly, the exclusive use of BiLSTM for the entire dataset resulted in an accuracy of 88% with an F-score of 0.88, while for longer text, the accuracy reached an impressive 92% with an F-score of 0.92. These findings vividly illustrate the inadequacy of a single model for handling the diversity of this problem effectively. By combining both models into a unified framework, our approach achieved an outstanding accuracy of 89%, showcasing the potential of model fusion in addressing the hate speech detection challenge in the dynamic digital landscape.

Furthermore, it is important to note that the success of our approach extends beyond the specific task of hate speech detection. The principles of leveraging diverse datasets from various digital platforms and accommodating varying post lengths, combined with model fusion, can be further explored and applied to a wide range of text classification tasks. Whether it's sentiment analysis, topic categorization, or content moderation, the methodology presented in this study offers a promising avenue for enhancing the efficiency and accuracy of text classification across the digital landscape. Our research contributes to creating a safer and more inclusive online environment and paves the way for innovative solutions in addressing text classification challenges that span different digital platforms with varying post structures.

CONFLICTS OF INTEREST

The authors declare that there are no conflicts of interest regarding the publication of this manuscript. Any affiliations, financial involvement, or relationships with organizations or entities that might pose a conflict of interest with the subject matter discussed in this work are hereby disclosed.

REFERENCES

[1] F. M. Plaza-Del-Arco, M. D. Molina-González, L. A. Ureña-López, and M. T. Martín-Valdivia, "A multi-task learning approach to hate speech detection leveraging sentiment analysis," *IEEE Access*, vol. 9, pp. 112478–112489, 2021.

[2] N. S. Mullah and W. M. N. W. Zainon, "Advances in machine learning algorithms for hate speech detection in social media: A review," *IEEE Access*, vol. 9, pp. 88364–88376, 2021.

[3] Z. Zhang and L. Luo, "Hate speech detection: A solved problem? The challenging case of long tail on Twitter," *Semantic Web*, vol. 10, no. 5, pp. 925–945, Sep. 2019.

[4] S. Mishra, S. Prasad, and S. Mishra, "Exploring multi-task multi-lingual learning of transformer models for hate speech and offensive speech identification in social media," *Social Netw. Comput. Sci.*, vol. 2, no. 2, pp. 1–19, Apr. 2021.

[5] M. Mozafari, R. Farahbakhsh, and N. Crespi, "Cross-lingual few-shot hate speech and offensive language detection using meta learning," *IEEE Access*, vol. 10, pp. 14880–14896, 2022.

[6] K. T. Mursi, M. D. Alahmadi, F. S. Alsubaei, and A. S. Alghamdi, "Detecting Islamic radicalism Arabic tweets using natural language processing," *IEEE Access*, vol. 10, pp. 72526–72534, 2022.

[7] N. Vashistha and A. Zubiaga, "Online multilingual hate speech detection: Experimenting with Hindi and English social media," *Information*, vol. 12, no. 1, p. 5, Dec. 2020.

[8] R. Singh, S. Subramani, J. Du, Y. Zhang, H. Wang, K. Ahmed, and Z. Chen, "Deep learning for multi-class antisocial behavior identification from Twitter," *IEEE Access*, vol. 8, pp. 194027–194044, 2020.

[9] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Proc. Int. AAAI Conf. Web Social Media*, 2017, vol. 11, no. 1, pp. 512–515.

[10] S. MacAvaney, H.-R. Yao, E. Yang, K. Russell, N. Goharian, and O. Frieder, "Hate speech detection: Challenges and solutions," *PLoS ONE*, vol. 14, no. 8, Aug. 2019, Art. no. e0221152.

[11] N. A. Ghani, S. Hamid, I. A. Targio Hashem, and E. Ahmed, "Social media big data analytics: A survey," *Comput. Hum. Behav.*, vol. 101, pp. 417–428, Dec. 2019.

[12] X. Sun, D. Yang, X. Li, T. Zhang, Y. Meng, H. Qiu, G. Wang, E. Hovy, and J. Li, "Interpreting deep learning models in natural language processing: A review," 2021, *arXiv:2110.10470*.

[13] H. Wang, J. He, X. Zhang, and S. Liu, "A short text classification method based on N-gram and CNN," *Chin. J. Electron.*, vol. 29, no. 2, pp. 248–254, 2020.

[14] Y. Zhou, J. Li, J. Chi, W. Tang, and Y. Zheng, "Set-CNN: A text convolutional neural network based on semantic extension for short text classification," *Knowl.-Based Syst.*, vol. 257, Dec. 2022, Art. no. 109948.

[15] J. Xu, Y. Cai, X. Wu, X. Lei, Q. Huang, H.-F. Leung, and Q. Li, "Incorporating context-relevant concepts into convolutional neural networks for short text classification," *Neurocomputing*, vol. 386, pp. 42–53, Apr. 2020.

[16] J. Du, C.-M. Vong, and C. L. P. Chen, "Novel efficient RNN and LSTM-like architectures: Recurrent and gated broad learning systems and their applications for text classification," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1586–1597, Mar. 2021.

[17] W. K. Sari, D. P. Rini, and R. F. Malik, "Text classification using long short-term memory with glove," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 5, no. 2, pp. 85–100, 2019.

[18] M. Shi, K. Wang, and C. Li, "A C-LSTM with word embedding model for news text classification," in *Proc. IEEE/ACIS 18th Int. Conf. Comput. Inf. Sci. (ICIS)*, Jun. 2019, pp. 253–257.

[19] W. Warner and J. Hirschberg, "Detecting hate speech on the world wide web," in *Proc. 2nd Workshop Lang. Social Media*, 2012, pp. 19–26.

[20] R. Alshalan and H. Al-Khalifa, "A deep learning approach for automatic hate speech detection in the Saudi Twittersphere," *Appl. Sci.*, vol. 10, no. 23, p. 8614, Dec. 2020.

[21] 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.

[22] 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.

[23] S. Modha, P. Majumder, T. Mandl, and C. Mandalia, "Detecting and visualizing hate speech in social media: A cyber watchdog for surveillance," *Expert Syst. Appl.*, vol. 161, Dec. 2020, Art. no. 113725.

[24] P. Kapil and A. Ekbal, "A deep neural network based multi-task learning approach to hate speech detection," *Knowl.-Based Syst.*, vol. 210, Dec. 2020, Art. no. 106458.

[25] F. M. Plaza-del-Arco, M. D. Molina-González, L. A. Ureña-López, and M. T. Martín-Valdivia, "Comparing pre-trained language models for Spanish hate speech detection," *Expert Syst. Appl.*, vol. 166, Mar. 2021, Art. no. 114120.

[26] Z. Mossie and J.-H. Wang, "Vulnerable community identification using hate speech detection on social media," *Inf. Process. Manage.*, vol. 57, no. 3, May 2020, Art. no. 102087.

[27] M. S. H. Ameur and H. Aliane, "AraCOVID19-MFH: Arabic COVID-19 multi-label fake news & hate speech detection dataset," *Proc. Comput. Sci.*, vol. 189, pp. 232–241, Jan. 2021.

[28] A. M. U. D. Khanday, S. T. Rabani, Q. R. Khan, and S. H. Malik, "Detecting Twitter hate speech in COVID-19 era using machine learning and ensemble learning techniques," *Int. J. Inf. Manage. Data Insights*, vol. 2, no. 2, Nov. 2022, Art. no. 100120.

[29] G. D. Valle-Cano, L. Quijano-Sánchez, F. Liberatore, and J. Gómez, "SocialHaterBERT: A dichotomous approach for automatically detecting hate speech on Twitter through textual analysis and user profiles," *Expert Syst. Appl.*, vol. 216, Apr. 2023, Art. no. 119446.

[30] P. Fortuna, J. Soler-Company, and L. Wanner, "How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets?" *Inf. Process. Manage.*, vol. 58, no. 3, May 2021, Art. no. 102524.

[31] D. Mody, Y. Huang, and T. E. Alves de Oliveira, "A curated dataset for hate speech detection on social media text," *Data Brief*, vol. 46, Feb. 2023, Art. no. 108832.

[32] O. de Gibert, N. Perez, A. García-Pablos, and M. Cuadros, "Hate speech dataset from a white supremacy forum," 2018, *arXiv:1809.04444*.

[33] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "Predicting the type and target of offensive posts in social media," 2019, *arXiv:1902.09666*.

[34] N. Ljubešić, D. Fišer, and T. Erjavec, "Offensive language dataset of Croatian, English and Slovenian comments FRENK 1.0," 2021. [Online]. Available: http://hdl.handle.net/11356/1433

[35] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. R. Pardo, P. Rosso, and M. Sanguinetti, "SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter," in *Proc. 13th Int. Workshop Semantic Eval.*, 2019, pp. 54–63.

[36] N. Ousidhoum, Z. Lin, H. Zhang, Y. Song, and D.-Y. Yeung, "Multilingual and multi-aspect hate speech analysis," 2019, *arXiv:1908.11049*.

[37] T. Mandl, S. Modha, P. Majumder, D. Patel, M. Dave, C. Mandlia, and A. Patel, "Overview of the HASOC track at FIRE 2019: Hate speech and offensive content identification in Indo–European languages," in *Proc. 11th Forum for Inf. Retr. Eval.*, Dec. 2019, pp. 14–17.

[38] T. Mandl, S. Modha, A. Kumar M, and B. R. Chakravarthi, "Overview of the HASOC track at FIRE 2020: Hate speech and offensive language identification in Tamil, Malayalam, Hindi, English and German," in *Proc. Forum for Inf. Retr. Eval.*, Dec. 2020, pp. 29–32.

[39] A. Gautam, P. Mathur, R. Gosangi, D. Mahata, R. Sawhney, and R. R. Shah, "#MeTooMA: Multi-aspect annotations of tweets related to the MeToo movement," in *Proc. Int. AAAI Conf. Web Social Media*, vol. 14, 2020, pp. 209–216.

[40] R. Agarwal. *Twitter Hate Speech*. Accessed: Nov. 20, 2023. [Online]. Available: https://www.kaggle.com/datasets/vkrahul/twitter-hate-speech

[41] M. Alberight. *Classified Tweets*. Accessed: Nov. 20, 2023. [Online]. Available: https://www.kaggle.com/datasets/munkialbright/classified-tweets

[42] S. Reddy. *Malignant Comment Classification*. Accessed: Nov. 20, 2023. [Online]. Available: https://www.kaggle.com/datasets/surekharamireddy/malignant-comment-classification

[43] A. Toosi. *Twitter Sentiment Analysis*. Accessed: Nov. 20, 2023. [Online]. Available: https://www.kaggle.com/datasets/arkhoshghalb/twitter-sentiment-analysis-hatred-speech

[44] Usharengaraju. *Dynamically Generated Hate Speech Dataset*. Accessed: Nov. 20, 2023. [Online]. Available: https://www.kaggle.com/datasets/usharengaraju/dynamically-generated-hate-speech-dataset

[45] A. Samshyn. *Hate Speech and Offensive Language Dataset*. Accessed: Nov. 20, 2023. [Online]. Available: https://www.kaggle.com/datasets/mrmorj/hate-speech-and-offensive-language-dataset

[46] A. Edwards, L. Edwards, and A. Martin, "Cyberbullying perceptions and experiences in diverse youth," in *Proc. Conf. Human Factors Cybersecurity (AHFE)*. New York, NY, USA: Springer, Jul. 2020, pp. 9–16.

[47] Mendeley. *Cyberbullying Dataset*. Accessed: Nov. 20, 2023. [Online]. Available: https://data.mendeley.com/datasets/jf4pzyvnpj/1

[48] B. Wei, J. Li, A. Gupta, H. Umair, A. Vovor, and N. Durzynski, "Offensive language and hate speech detection with deep learning and transfer learning," 2021, *arXiv:2108.03305*.

[49] C. Fellbaum, "WordNet," in *Theory and Applications of Ontology: Computer Applications*. Dordrecht, The Netherlands: Springer, 2010, pp. 231–243.

[50] X. Liu, Q. Tong, X. Liu, and Z. Qin, "Ontology matching: State of the art, future challenges, and thinking based on utilized information," *IEEE Access*, vol. 9, pp. 91235–91243, 2021.

[51] B. Liang, Q. Liu, J. Xu, Q. Zhou, and P. Zhang, "Aspect-based sentiment analysis based on multi-attention CNN," *J. Comput. Res. Development. Chin.*, vol. 54, no. 8, pp. 1724–1735, 2017.
[52] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2015, vol. 29, no. 1, pp. 2267–2273.
[53] J. Cai, J. Li, W. Li, and J. Wang, "Deeplearning model used in text classification," in *Proc. 15th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process. (ICCWAMTIP)*, Dec. 2018, pp. 123–126.
[54] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[56] Y. Zhou, Y. Yang, H. Liu, X. Liu, and N. Savage, "Deep learning based fusion approach for hate speech detection," *IEEE Access*, vol. 8, pp. 128923–128929, 2020.
[57] H. S. Alatawi, A. M. Alhothali, and K. M. Moria, "Detecting white supremacist hate speech using domain specific word embedding with deep learning and BERT," *IEEE Access*, vol. 9, pp. 106363–106374, 2021.
[58] B. Gambäck and U. K. Sikdar, "Using convolutional neural networks to classify hate-speech," in *Proc. 1st Workshop Abusive Lang.*, 2017, pp. 85–90.
[59] 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.
[60] A. 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. Int. AAAI Conf. Web Social Media*, 2018, vol. 12, no. 1, pp. 491–500.
[61] C. Baydogan and B. Alatas, "Metaheuristic ant lion and moth flame optimization-based novel approach for automatic detection of hate speech in online social networks," *IEEE Access*, vol. 9, pp. 110047–110062, 2021.
[62] Kaggle. *Detecting Insults in Social Commentary*. Accessed: Nov. 20, 2023. [Online]. Available: https://www.kaggle.com/competitions/detecting-insults-in-social-commentary/data
[63] M. Gaikwad, S. Ahirrao, K. Kotecha, and A. Abraham, "Multi-ideology multi-class extremism classification using deep learning techniques," *IEEE Access*, vol. 10, pp. 104829–104843, 2022.
[64] K. A. Qureshi and M. Sabih, "Un-compromised credibility: Social media based multi-class hate speech classification for text," *IEEE Access*, vol. 9, pp. 109465–109477, 2021.
[65] Z. Waseem, "Are you a racist or Am I seeing things? Annotator influence on hate speech detection on Twitter," in *Proc. 1st Workshop NLP Comput. Social Sci.*, Nov. 2016, pp. 138–142.
[66] M. ElSherief, S. Nilizadeh, D. Nguyen, G. Vigna, and E. Belding, "Peer to peer hate: Hate speech instigators and their targets," in *Proc. Int. AAAI Conf. Web Social Media*, 2018, vol. 12, no. 1, pp. 52–61.
[67] P. Mathur, R. Sawhney, M. Ayyar, and R. Shah, "Did you offend me? Classification of offensive tweets in Hinglish language," in *Proc. 2nd Workshop Abusive Lang. (ALW2)*, 2018, pp. 138–148.
[68] S. D. Swamy, A. Jamatia, and B. Gambäck, "Studying generalisability across abusive language detection datasets," in *Proc. 23rd Conf. Comput. Natural Lang. Learn. (CoNLL)*, 2019, pp. 940–950.
[69] M. Karan and J. Šnajder, "Cross-domain detection of abusive language online," in *Proc. 2nd Workshop Abusive Lang. Online (ALW2)*, 2018, pp. 132–137.

WAQAS SHARIF received the master's degree in computer science from The Islamia University of Bahawalpur, Punjab, Pakistan, in 2018, where he is currently pursuing the Ph.D. degree in computer science. He is also a Lecturer with the Department of Computer Science, The Islamia University of Bahawalpur. He has over six years of professional experience, with four years dedicated to teaching, alongside two years in software development. His research interests include natural language processing, machine learning, and bioinformatics. His expertise extends to serving as a Reviewer for various journals, including IEEE Access and *IJIST*.

SAIMA ABDULLAH received the Ph.D. degree from the Department of Computer Science and Electronic Engineering, University of Essex, U.K. Currently, she holds the position of an Assistant Professor with the Department of Computer Science and Information Technology, The Islamia University of Bahawalpur, Pakistan. As a member of the Multimedia Research Group, DCS, she focuses on efficient and secure communication of multimedia data over future generation network technologies. Her primary research interests include wireless networks and communications, future internet technology, and network performance analysis. She has authored ten articles in these areas. She serves as a reviewer for various international journals.

SAMAN IFTIKHAR (Member, IEEE) received the M.S. and Ph.D. degrees in information technology from the National University of Sciences and Technology (NUST), Islamabad, Pakistan, in 2008 and 2014, respectively. She is currently an Assistant Professor with Arab Open University, Saudi Arabia. She has published 12 research articles in various reputed journals on her credit. She has presented nine research papers in prestigious conferences in Pakistan, Dubai, Japan, Malaysia, and America. She also published one book chapter. Her research interests include networking, information security, cyber security, machine learning, data mining, distributed computing, and semantic web. She was a member of IEEE WIE, IEEE IAS, IEEE Computer Society, and IEEE Communication Society. She was with the IEEE Academic Pakistan Initiative, as a Speaker and a Coordinator.

DANIAH AL-MADANI received the master's degree in computer science from Ryerson University, Toronto, Canada. Currently, she is with Arab Open University, Jeddah, Saudi Arabia, as a Lecturer. She has several publications in reputed journals and conferences. Her research interests include the IoT, data science, and data mining.

SHAHZAD MUMTAZ received the master's degree in computer science from The Islamia University of Bahawalpur, Pakistan, in 2005, and the Ph.D. degree in computer science from Aston University, U.K., in 2015. He was the Assistant Director (computer) with the National Highway Authority, Pakistan, from December 2005 to October 2007. Recently, he has worked in other areas, such as natural language processing/analytics and high-performance computing. His research projects including Probabilistic Modeling of Blood Glucose Through Eye Parameters, An Analysis of the Protein Family of Major Histocompatibility Complex, Predictive Modeling of Accidents and Emergency Arrivals and Admissions, Patient-Specific Recommendation Systems for HIP Joint Patients, and Predictive Modeling of Extreme Content from Twitter in the Context of Afghanistan. His research interests include the areas of machine learning and data mining and their application to health informatics domains.