IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

HCovBi-Caps: Hate Speech Detection using Convolutional and Bi-Directional Gated Recurrent Unit with Capsule Network

SHAKIR KHAN[1], ASHRAF KAMAL[4], MOHD FAZIL[3], MOHAMMED ALI ALSHARA[1], VINEET KUMAR SEJWAL[2], REEMIAH MUNEER ALOTAIBI[1], ABDULRAUF BAIG[1] and SALIHAH ALQAHTANI[1],

[1]College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University, Riyadh, Saudi Arabia
[2]Department of Computer Science, Jamia Millia Islamia, New Delhi, India
[3]Department of Computer Engineering, Qatar University, Qatar
[4]ACL Digital, Bengaluru, India

Corresponding author: Shakir Khan (E-mail: Sgkhan@imamu.edu.sa)

ABSTRACT Adversaries and anti-social elements have exploited the rapid proliferation of computing technology and online social media in the form of novel security threats, such as fake profiles, hate speech, social bots, and rumors. The hate speech problem on online social networks (OSNs) is also widespread. The existing literature has machine learning approaches for hate speech detection on OSNs. However, the effectiveness of contextual information at different orientations is understudied. This study presents a novel *Convolutional*, *BiGRU*, and *Capsule* network-based deep learning model, HCovBi-Caps, to classify the hate speech. The proposed model is evaluated over two Twitter-based benchmark datasets – DS1(balanced) and DS2(unbalanced) with the best performance of 0.90, 0.80, and 0.84 respectively considering *precision*, *recall*, and *f-score* over unbalanced dataset. In terms of training and validation accuracy, the proposed model shows the best performance of 0.93 and 0.90, respectively, over the unbalanced dataset. In comparative evaluation, HCovBi-Caps demonstrates a significantly better performance than state-of-the-art approaches. In addition, HCovBi-Caps shows comparatively better performance over the unbalanced dataset. We also investigate the impact of different hyperparameters on the efficacy of HCovBi-Caps to ascertain the selection of their values. We observed that a higher value of *routing iterations* adversely affects the model performance, whereas a higher value of *capsule dimension* improves the performance.

INDEX TERMS Hate speech detection, Twitter data analysis, convolutional layer, capsule network, BiGRU, deep learning.

I. INTRODUCTION

In the last few decades, advancement in computing technology, especially in OSNs, has changed the users' communication behavior. Online social networking platforms, such as Facebook, Twitter, Weibo, and WhatsApp, are popular and part of peoples' routine life. On these platforms, users discuss the current trends and express views, sharing them over the virtual network of family and friends on OSNs. Users currently use either one or another OSN platform and generally more than one platform. Interactions among the large user base generate massive data, which we can mine to extract valuable insights. In the existing literature, researchers have used the OSN datasets in many domains, such as sentiment analysis [1], [2], social bot detection [3], [4], sarcasm detection [5] and its different specializations [6], [7], [8], humor detection [9], recommendation system [10], [11], and rumor detection [12].

The users' discourse on OSNs ranges from diverse themes, such as politics, democracy, economics, fashion, and science & technology, to sharing travel and nature experiences. Twitter, a microblogging platform, is one of the widely used OSN platforms that allows the users to express views within a limit of 280 characters. Politicians, celebrities, and public figures generally use Twitter to connect with their supporters

and followers. These high-profile people use it to announce events and professional updates as tweets, triggering the platform users whom reactions contain various emotions with different stances concerning the post. These OSNs have also attracted anti-social and ill-minded people, who generally respond to a tweet with hateful and abusive comments. The adversaries and anti-social elements generally target a specific community, race, gender, or socio-political group. Hateful and abusive content has adverse impacts and occasionally creates depression and anxiety issues in the targeted individual/community.

The existing literature has no universally accepted definition of hate speech, and even OSNs do not have a consensus. In the literature, researchers use hate speech and abusive speech interchangeably [13]. Hate speech is defined as offensive and aggressive content targeted toward specific groups based on ethnicity, religion, gender, sexual orientation, or other characteristics. Twitter defines a tweet as hate speech (HS) "*that promotes violence against or directly attack or threatens other people based on race, ethnicity, nationality, sexual orientation, gender, religious affiliation, age, disability, or serious disease*." Hate speech on OSN platforms can cause riots, resulting in communal disharmony in real life. For example, OSNs were recently overwhelmed with hate content and anti-social propaganda related to the Shaheen Bagh protest in Delhi (India) against the National Register of Citizens, Citizenship Amendment Act, and National Population Register[1]. Anti-Asia and sinophobic content became prevalent on OSN platforms during the COVID-19 pandemic, blaming the Asian people for the pandemic [14]. The OSN service providers have policies and methods to tackle such content. However, no usable and well-accepted solution exists to date. For example, Twitter occasionally deletes the tweets and comments or suspends the violating users.

Researchers from academia and industry are continuously introducing novel methods using techniques from statistical analysis to pattern mining and deep learning to tackle this growing problem of hate content on social media platforms [15], [16], [17]. Among the existing categories of approaches, machine learning methods are more effective than other categories of methods [18]. However, the current models still lag a satisfactory level of accuracy over a different set of datasets. To this end, this study presents HCovBi-Caps, a novel deep learning model incorporating contextual information at different orientations.

A. OUR CONTRIBUTIONS

The role of context is vital for detecting HS in online content. To extract the relevant hate speech-related information having context in different orientations is a challenging and notable research problem, especially in short texts, such as tweets. In this direction, this paper presents a capsule-based deep neural network model to detect HS on Twitter. The proposed model considers hate speech as a two-class (binary classification) problem, wherein the trained model classifies a tweet as either HS or non-hate speech (NHS). The proposed model, HCovBi-Caps, is a novel end-to-end deep learning model for HS detection. It integrates an input, embedding, convolutional, bi-directional gated recurrent unit (BiGRU), a capsule network, dense and outputs layers. The HCovBi-Caps extracts HS-related contextual information from the input text, considering different orientations and ordering of words. The HCovBi-Caps first converts the input text into a numeric vector and further convert it into an embedding vector at the embedding layer. The model transfers the embedding layer output to the convolutional layer to extract the low-level syntactic and semantic features. The BiGRU further retrieves the latent semantic features from sequences having contextual information. BiGRU is effective than a simple GRU because it considers the contextual information in the forward and backward directions. As a result, BiGRU extracts the preceding and succeeding contextual information-related sequences from the features generated by the convolutional layer. The capsule network further retrieves the contextual information of different orientations by maintaining the ordering of words in the input text. The capsule network considers the part-whole spatial/local relationship of HS-related words. This layer covers the hate-related context by considering various orientations of the input text. The dynamic routing algorithm used in the capsule network increases the weight values of the HS-related latent contextual information. HCovBi-Caps passes the output vector from the capsule network to the dense layer that is further given to a sigmoid function to classify the input text as either HS or NHS.

The problem of hate speech is non-trivial and prevalent in OSNs. Hateful content on social media may lead to large-scale violence and riots in real-life. To address the hate speech problem, researchers have presented various approaches to detect it. OSN service providers are also introducing in-build solutions and formulating policies to tackle this menace. HCovBi-Caps is a small contribution to the collaborative effort going around the world to eradicate the hateful and anti-social content from OSNs. In this direction, HCovBi-Caps detects the hate content written with different contextual orientations.

Overall, the main contributions of this paper can be summarized as follows.

- Introduce a novel deep neural network model, HCovBi-Caps, by integrating the BiGRU, Convolutional layer and Capsule network to incorporate the contextual information at different orientations for hate speech detection.
- Perform the comparative evaluation of HCovBi-Caps over two benchmark datasets to establish its efficacy.
- Investigate the impact of different hyper-parameters values on the efficacy of HCovBi-Caps performance to observe the best hyper-parameters values.

[1]https://www.aljazeera.com/news/2021/9/21/india-bihar-muslims-nrc-assam-citizenship-seemanchal

For reproducibility, we release our implementation code at GitHub[2] repository. The remainder of this paper is organized as follows. Section II presents a literature overview of a brief description of existing approaches on HS detection. Section III provides a detailed description of the introduced model, including functional details of the proposed HCovBi-Caps model. Section IV presents the experimental setup and evaluation results. This section further establishes the efficacy of the proposed model by performing its comparative evaluation with two state-of-the-art and six baseline methods. Section V investigates and discusses the effect of various hyperparameters on the efficacy of the proposed HCovBi-Caps model. Finally, Section VII concludes the paper with future research directions.

II. RELATED WORKS

This section presents a synopsis of existing literature on computational detection of hate content over OSN platforms. Researchers have studied hate speech and related research directions like rumor detection [12], credibility analysis [19] and presented approaches to track and curb this nuisance. Thus, researchers presented statistical analysis, pattern mining, and machine learning-based methods, wherein machine learning methods are prevalent and effective. This study classifies the existing approaches into two categories – (i) machine learning-based methods and (ii) deep learning-based methods. Finally, the section ends with a highlight on current status and limitations.

A. MACHINE LEARNING-BASED METHODS

Warner and Hirschberg [20] used unigram, part of speech, and other template-based features in one of the early approaches to tackle the hate speech problem. The authors further trained the SVM^{light} model using linear kernel and evaluated it over two datasets from Yahoo and the American Jews Congress websites to classify the hate from non-hate content. In another approach, Kwok and Wand [21] used unigram features and further trained Naive Bayes classifier to segregate the racist tweets from ordinary ones with an accuracy of 76%. They experimentally concluded that bigram, trigram, and sentiment improve model performance. In another approach based on n-gram, Burnap and Williams [22] employed various n-gram features and trained three machine learning models: Bayesian logistic regression, SVM, and voted ensemble classifiers. They further evaluated the trained models over the crawled Twitter dataset and reported that *voted ensemble classifiers* show the best performance. Djuric et al. [23] utilized paragraph2vec [24] language model for the joint modeling of comments and words collected from the Yahoo Financial website. They further used the trained dense vector representation to learn a logistic regression model to classify the hate comment.

In a popular approach, Waseem and Hovy [25] open-sourced a benchmarked dataset of 16k tweets containing hate speech. The authors further used $1^{\sim}4$-gram features to train the logistic regression classifier to segregate the hate and ordinary tweets. The best model shows performance with an F1-score of 73.89. They also used location and gender features and gender with n-gram reports the best performance. The various categories of hateful content, such as hate, offensive, abusive, have subtle differences; however, it is understudied. Davidson et al. [18] investigated the difference between hate, abusive, spam, and genuine content and used unigram, bigram, POS tag-based n-grams, Flesch–Kincaid Grade Level, Flesch Reading Ease scores, sentiment score, and various linguistic features to train logistic regression classifier to segregate them. Malmasi and Zampieri [26] presented a similar approach using character and word n-gram features to train linear SVM classifier to classify the hate, offensive, and ordinary contents. They experimented using various feature combinations and reported that character 4-gram shows the best performance.

B. DEEP LEARNING-BASED METHODS

Classical machine learning shows good performance but requires feature engineering, a manual, time-consuming, and tedious task. As a result, these approaches depend on human intelligence, therefore, include human bias. Recently, researchers started exploiting the advancements in deep neural networks and presented various deep learning models for HS detection to avoid these limitations [27], [28], [15]. Badjatiya et al. [27] introduced one of the first deep learning-based approaches for hate speech detection. They evaluated various neural network architectures – CNN, LSTM, and DNN, for hate content detection by employing different word representation techniques, such as Glove, word2vec, and FastText. Park and Fung [28] presented a hybrid model integrating the logistic regression and CNN architecture to segregate abusive tweets from genuine tweets. On investigation, the authors found that the hybrid model performs better than the isolated machine and deep learning models. Zhang et al. [15] integrated the convolutional neural network and gated recurrent network to present a deep learning model to classify the hate content and reported the best performance with an F1- score of 0.94. The existing deep learning models used word embedding representations [27], [29], [15], [16] which is a static representation and does not include contextual information. Cao et al. [30] incorporated sentiment and topic-based contextual information using the word, sentiment, and topic-based representations and presented a hybrid deep learning model constructed using CNN, LSTM, and attention mechanism to classify the hate content. Roy et al. [31] introduced a deep convolutional neural network-based framework for hate content detection and reported the best performance with an accuracy of 92%. Researchers recently targeted hateful and abusive content written in code-mixed languages, such as *Hinglish*. In this research direction, Kamble and Joshi [32] compared three classical deep learning models (CNN, LSTM, and

[2]https://github.com/Ashraf-Kamal/Hate_Speech_Detection

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3143799, IEEE Access

BiLSTM) using a domain-specific word embedding to classify the code-mixed hate tweets from regular content. The authors found that the trained domain-specific code-mixed embedding provided better performance than pre-trained word embedding. Finding the labeled hateful content in various languages is tedious. Researchers recently started presenting approaches to detect hate speech in low-resource languages. Pamungkas et al. [33] presented a zero-shot learning approach for cross-lingual hate speech detection. Hate-reflecting words in textual content are used in different contexts depending on the situation. For example, certain words may be used in a derogatory manner in one context to target a race of people but could be slang in another context. The researchers exploited the contextual embedding trained using transformer-based language models, such as BERT for HS detection, to incorporate the contextual information of hate-inciting words [34], [35]. Researchers have also studied various aspects of hate speech, such as identification of hate-targeted vulnerable communities [36] to analyze the generalization of different categories of HS detection models across datasets [37].

C. CURRENT STATUS AND LIMITATIONS

The review of existing literature concludes the persistence of the hate speech problem and the need for globally accepted automatic hate content detection approaches. This problem is also evolving with different complexities. In the existing categories of methods, deep learning models are most effective for HS detection. Researchers also acknowledged that contextual information is crucial for HS detection, but extracting contextual information through different orientations of an instance is difficult. The concept of the capsule network presented by Hinton et al. [38] is prolific because it captures contextual information at different orientations. The existing literature has many approaches using capsule networks for NLP applications, such as text classification [39], clickbait detection [40], and sentiment classification [41]. Authors in [17], [42] have used capsule network for HS detection, but its utilization in this research direction is understudied. Integrating the capsule network with BiGRU and convolutional layer to retrieve the contextual information in different orientations for hate content detection is a fascinating, non-trivial, and significant research problem.

III. PROPOSED APPROACH

This section presents the proposed HCovBi-Caps model for hate content detection. It includes a detailed description of data crawling, pre-processing, and the proposed HCovBi-Caps model in the following subsections.

A. DATA CRAWLING

We perform the empirical evaluation of the HCovBi-Caps model over two Twitter-based benchmark datasets. To this end, a data crawler is developed in Python to retrieve the tweets by accessing them via `REST` API. We use the `Tweepy`[3], a Python library, to crawl the tweets and save them on a local machine to use in the later stages.

B. DATA PRE-PROCESSING

The HCovBi-Caps model first pre-processes the crawled tweets to filter the useless information. Algorithm 1 represents an algorithmic representation of pre-processing steps. In the pre-processing, `HCovBi-Caps` performs the following steps:

- Filtering of Twitter-related markers and symbols, such as hashtags, URLs, mentions, and retweets.
- Filtering of real numbers, stop words, ampersands, redundant white spaces, dots, single and double quotes, non-ASCII characters, commas, emoticons, exclamation marks, interjection, and punctuation marks.
- Removal of all redundant tweets.
- Finally, tweets are converted into lower-case.

C. PROPOSED HCOVBI-CAPS MODEL

Figure 1 shows the architecture of the proposed HCovBi-Caps model. This section presents a detailed description of various HCovBi-Caps components. The model comprises input, embedding, and convolutional layers integrated with BiGRU and capsule network followed by dense and output layers. Algorithm 2 presents an algorithmic representation of the proposed model. The following subsections discuss the functionality of these layers.

1) Input Layer

The HCovBi-Caps model passes the pre-processed tweet to the input layer, which tokenizes and maps each token into a unique number using the dictionary-based index value. Thus, the input layer maps the input text into a numeric vector. Mathematically, the input layer represents the input text as follows: if an input text (tweet) T comprises n tokens, then each token is replaced with its dictionary index such that $T \in \Re^{1\times n}$. Further, padding p is applied to input text to maintain the fixed-length size of the input vector. Finally, all the input texts are transformed into an input matrix $T \in \Re^{1\times p}$.

2) Embedding Layer

The embedding layer learns word representation as a low-dimensional dense vector by training over massive real-world corpora. This layer identifies semantically-related words having similar vector representations. The embedding layer has diverse applications from text classification, machine translation to recommender systems. In this paper, we use pre-trained (GloVe)[4] embedding, a well-known word representation technique based on word distribution statistics, for word representation. GloVe uses an unsupervised approach to train vector representation of words using a co-occurrence matrix of word pairs. In this study, we use Twitter-based GloVe embedding because the

[3]https://www.tweepy.org/

[4]https://nlp.stanford.edu/projects/glove/ [accessed on 04-Sep-2021]

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3143799, IEEE Access

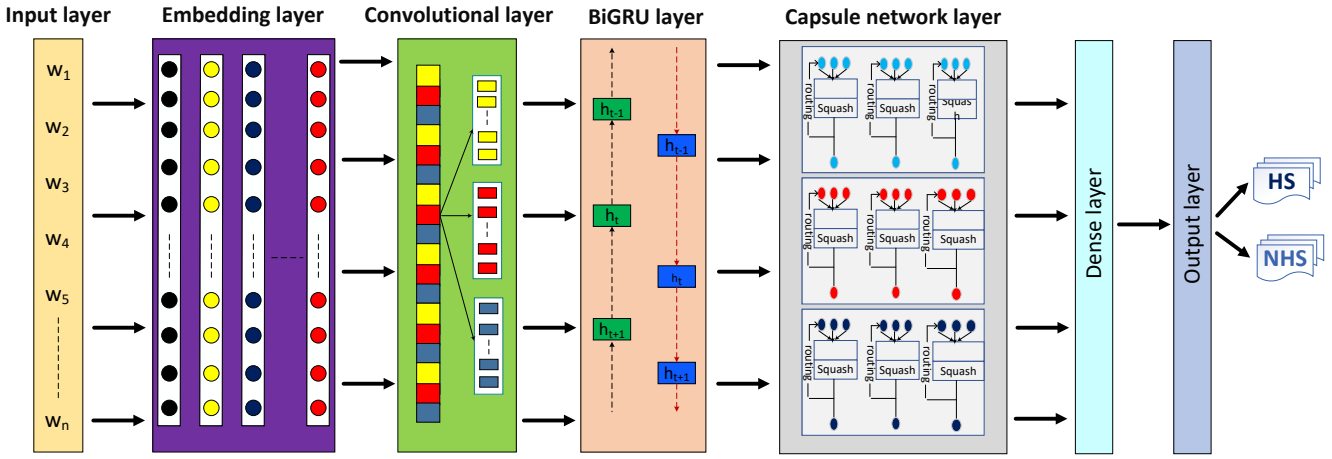


FIGURE 1: Architecture of the proposed HCovBi-Caps model for hate speech detection

used datasets are Twitter-based. Each token of the padded fixed-length input text is converted into a corresponding GloVe embedding of dimension d, transforming the input vector-matrix to $T \in \Re^{p \times d}$.

3) Convolutional Layer

The HCovBi-Caps applies the convolutional layer over the embedding vector to extract the spatial features. The proposed model uses the one-dimensional convolutional operation because the input embedding vector is a row vector. The convolutional layer uses 128 filters of three different filter sizes to extract the hate-related spatial and temporal features comprising 128 sequences. Equation 1 represents the n^{th} feature sequence, f_n, generated from word window x_t, where w_t, b, and f(·) represent the filter weight, bias, and *ReLU* (a popular nonlinear activation), respectively. The 128 filters execute the convolutional operation from top to bottom, extracting the feature sequence as f_s = [f_1, f_2, ..., f_{128}] from the input text. The max-pooling operation is applied to obtain the underlying feature map. Finally, HCovBi-Caps concatenate the output from each filter to extract the final feature vector, which is an input to the next layer.

$$f_n = f(w_t \cdot x_t + b) \tag{1}$$

4) BiGRU Layer

BiGRU, a type of RNN, is generally used in sequential modeling problems extracting the sequences from the forward and backward directions. In BiGRU, a forward ($\overrightarrow{GRU}$) and a backward GRUs ($\overleftarrow{GRU}$) are integrated to retrieve the succeeding (i.e., f_1 to f_{128}) and preceding feature sequences (i.e., f_{128} to f_1), respectively. The proposed HCovBi-Caps model retrieves the forward and backward sequences having contextual information by applying the BiGRU layer on the convolutional layer output. Equations 2 and 3 present the outcomes from BiGRU in forward and backward directions, respectively. The BiGRU output is the hate-incorporating representation of the input text by integrating the contexts from the forward & backward directions. The BiGRU-based representation for a given feature sequence f_s of the input text is the concatenation of the forward, $\overrightarrow{h_f}$, and backward, $\overleftarrow{h_b}$, hidden states. The two hidden states integrate the information collected around L_{f_s} to retrieve the hate incorporating contextual information-based sequences. Finally, Equation 4 represents the concatenated contextual information-incorporating sequence as a final hidden state h_t, which is passed to the capsule network layer.

$$\overrightarrow{h_f} = \overrightarrow{GRU}(L_{f_s}), n \in [1, 128] \tag{2}$$

$$\overleftarrow{h_b} = \overleftarrow{GRU}(L_{f_s}), n \in [128, 1] \tag{3}$$

$$h_t = [\overrightarrow{h_f}, \overleftarrow{h_b}] \tag{4}$$

5) Capsule Network Layer

Traditional CNN cannot extract salient features. It also losses crucial information because of the application of the pooling technique. Thus, extracted features lose important information generated from activation functions using Max, Min, or Average pooling techniques. Hinton et al. [43] introduced capsule network, a novel neural network architecture, to extract the syntactically enriched features considering different orientations and local ordering of words from the input data [44]. Recently, it has shown remarkable performance in text classification and information retrieval problems. Unlike CNN, it can identify the part-whole spatial relationship within features in textual data, therefore, effectively identifying semantic representation and hidden contextual information from the input text [39], [40], [41].

In a capsule network, a capsule contains multiple capsules. Furthermore, a capsule is made of neurons to extract semantic and syntactic information. This network uses the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3143799, IEEE Access

modulus of the capsule in the form of a vector to represent the classification probability and the capsule direction to describe different orientations of the text. Hence, the capsule network representation is more efficient and enriched than traditional neural network models, such as CNN. The capsule network generates a vector rather than a scalar value as obtained in the pooling layer of CNN. Furthermore, the dynamic routing algorithm [45], a principle component of the capsule network, adjusts the weight of latent features facilitating the extraction of additional features. As a result, the capsule network improves the classification performance of the underlying model.

The HCovBi-Caps model uses the capsule network because of its advantages discussed in the above paragraphs. To this end, the final hidden state h_t, representing the output of the BiGRU layer, is passed to the capsule network layer. The resultant output of the capsule network is obtained using equations 5, 6, 7, 8, and 9. In equation 5, the final hidden state h_t of BiGRU is first converted into a feature capsule u_i using a non-linear activation function. Furthermore, u_i determines the correlation between the input and output layers and generates the prediction vector $\hat{u}_{j|i}$, where W_{ij} represents the weight matrix.

$$\hat{u}_{j|i} = W_{ij} u_i \tag{5}$$

The dynamic routing process is applied to calculate the coupling coefficients c_{ij}. This process ignores the trivial and irrelevant hate-related words from the input text. The weight of HS-related features is proportional to the coupling coefficient c_{ij}. For example, a feature with a high c_{ij} value has a higher weight and vice versa. This correlation helps in encoding vital HS-related contextual representations of the input text considering different orientations. The output of a capsule, s_j, is calculated as the summation of all the prediction vectors using equation 6.

$$s_j = \sum_{i=1}^{n} c_{ij} \hat{u}_{j|i} \tag{6}$$

The coupling coefficient c_{ij} is calculated by the *softmax* function using equation 7.

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_u \exp(b_{iu})} \tag{7}$$

Equation 8 updates b_{ij} through until the iteration requirements are met. It represents the higher layers of the capsule network.

$$b_{ij} \leftarrow b_{ij} + V_j^T \, f(u_i, \theta_j) \tag{8}$$

Equation 9 normalizes the final output vector v_j via squash function (a nonlinear activation function), which includes different orientations and local ordering of words/tokens of the input text.

$$v_j = \frac{||s_j||^2}{1 + ||s_j||^2} \frac{s_j}{||s_j||} \tag{9}$$

Algorithm 1 DataPreparation_Algo

Input: T ← Set of Tweets, L← labels,
l_m ← maximum number of word in a tweet
Output: E ←Embedding matrix

for *t in T* **do**
  $T_t \leftarrow filterTwitterMarkers(t)$
  $T_i \leftarrow filterIrrelevant(T_t)$
  $T_d \leftarrow filterDuplicate(T_i)$
  $T_c \leftarrow convertCase(T_d)$
end
Encoding:
  $tok \leftarrow Tokenizer()$
  $tok.fit\_on\_texts(T_c)$
  $T_{cs} \leftarrow tok.texts\_to\_sequence(T_c)$
  $w_i \leftarrow tok.word\_index$
  $w_{count} \leftarrow len(word\_index)$
  $l \leftarrow labelEncoder()$
  $L_e \leftarrow l.fit\_transform(L)$
Padding:
  $T_p \leftarrow pad\_Sequences(T_{cs}, l_m)$
Glove Embedding Matrix:
  $G_e \leftarrow GloVe()$
for *word, index in word_index.items()* **do**
  $W_e \leftarrow G_e.get(word)$
  $E[index] \leftarrow W_e$
end

Algorithm 2 HCovBi_Caps

Input: E, W
Output: $classified\ tweets$
Model Construction:
  model ← Sequential()
  model ← model.add(Embedding(w_c, e_d, l_m, E))
  model ← model.add(Conv1D(128, 3, a_c='relu'))
  model ← model.add(BiGRU(128))
  model ← model.add(DropOut(0.4))
  model ← model.add(Capsule(3, 5, 4))
  model ← model.add(Flatten())
  model ← model.add(Dense(1, a_c='sigmoid'))
Model Compilation:
  model.compile('binary_crossentropy', 'Adam', 'acc')
Training & Evaluation:
  hist ← model.fit(T_p, 128, 50, 0.20)

6) Dense and output Layers

The output vector v_j generated from the capsule network passes to the fully connected layer. Finally, the output layer classifies the hate speech by applying the *sigmoid* function

on the dense layer output. The proposed HCovBi-Caps model uses the *binary cross-entropy* loss function.

IV. EXPERIMENTAL SETUP AND RESULTS

This section presents the experimental details of the proposed HCovBi-Caps model. It includes the description of the datasets, experimental settings, hyperparameter settings, evaluation metrics, evaluation results, and comparative analysis.

A. DATASETS

This study uses two Twitter-based datasets for the empirical evaluation of the proposed model. Among the two datasets, one is relatively balanced and one unbalanced to evaluate the HcovBi-Caps on two types of datasets. Table 1 presents a brief statistics of the datasets. A brief description of the datasets is provided in the following paragraphs.

- Founta et al. [46] (DS1): This dataset contains $80,000$ tweets, which are labeled as either one of the four classes – *abusive*, *hateful*, *normal*, or *spam*. We consider only two labels – *hateful* and *normal* because the proposed approach is a two-class problem. This dataset is a relatively balanced containing 2615 hate and 5385 non-hate tweets as shown in the first row of Table 1.
- Kaggle (DS2): The kaggle dataset consists of $31,962$ tweet. Out of which, 1421 hate and 9575 non-hate tweets are selected to create the final dataset DS2 as shown in the second row of Table 1. This dataset is unbalanced.

TABLE 1: Statistics of the datasets

Datasets	#HS	#NHS	Total	Type
DS1	2615	5385	8000	Relatively Balanced
DS2	1421	9579	10000	Unbalanced

B. EXPERIMENTAL SETTINGS

The HCovBi-Caps model is implemented using Python language. We performed all the experimental evaluations using a Windows-10 (64-bit) machine having Intel i-3 6006 processor and 8 GB RAM. We use Tweepy[5], an in-built library, to crawl the tweets from Twitter. Finally, Keras[6], a popular neural network library, is used to implement the proposed model.

C. HYPERPARAMETER SETTINGS

The proposed model splits both the datasets into the training and validation parts with a distribution of 80% and 20%, respectively. Furthermore, *dropout*, *batch size*, *verbose*, *epoch*, and *optimizer* are 0.4, 128, 2, 50, and *Adam*, respectively. In the CNN layer, *filter width*, *number of filters*, and the *pool size* are adjusted to 3, 128, and 2, respectively. The BiGRU layer uses 128 neurons for information processing. The *number of routing iterations*, *number of capsules*, and the *capsule dimension* in the capsule network layer are 3, 5, and 4, respectively. Table 2 presents various hyperparameters and their values used in the proposed HCovBi-Caps model.

0[h]

TABLE 2: Hyperparameters and their values used in HCovBi-Caps model

Hyperparameter	Value
Glove embedding dimension	50
Padding	25
Filter size (CNN layer)	3
Number of CNN Filters	128
Number of neurons (BiLSTM layer)	128
Dropout	0.4
Routing	3
Number of capsules	5
Dimension of capsule	4
Optimizer	Adam

D. EVALUATION METRICS

The performance of HCovBi-Caps model is evaluated using four standard information retrieval metrics, namely *Precision*, *Recall*, *F-score*, and *Accuracy*. These metrics are mathematically defined in Equations 10, 11, 12, and 13 respectively using True Positive, False Positive, True Negative, and False Negative. True Positive refers to the total number of correctly classified hate tweets. False Positive refers to the total number of non-hate tweets classified as hate tweets. True Negative refers to the total number of correctly classified non-hate tweets. Finally, False Negative refers to the total number of hate tweets classified as non-hate tweets.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{10}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{11}$$

$$\text{F-score} = \frac{2 \times \text{Precsion} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\#tweets} \tag{13}$$

E. EVALUATION RESULTS AND COMPARATIVE ANALYSIS

This section presents the evaluation results of the HCovBi-Caps model for HS detection over the two datasets. We also perform a comparative evaluation of the proposed model with two deep learning-based state-of-the-art and six baseline methods. Table 3 shows that proposed model demonstrates better performance results over both the

[5]https://www.tweepy.org/ [accessed on 04-Sep-2021]
[6]https://keras.io/ [accessed on 04-Sep-2021]

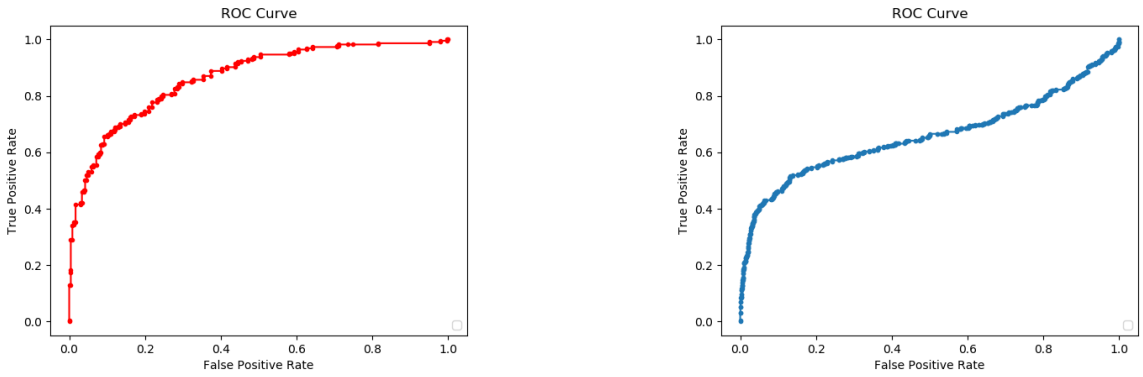
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3143799, IEEE Access

TABLE 3: Performance evaluation results of HCovBi-Caps over DS1 and DS2 datasets in terms of *precision*, *recall*, and *f-score*

Datasets →	**DS1 (balanced)**			**DS2 (unbalanced)**		
Methods ↓	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
HCovBi-Caps	0.80	0.73	0.76	0.90	0.80	0.84
Ding et al. [42]	0.64	0.60	0.61	0.65	0.61	0.62
Roy et al. [31]	0.60	0.40	0.48	0.61	0.58	0.59
CNN	0.55	0.32	0.40	0.65	0.35	0.45
LSTM	0.37	0.34	0.35	0.72	0.45	0.55
GRU	0.48	0.41	0.44	0.75	0.53	0.62
BiLSTM	0.64	0.38	0.47	0.83	0.67	0.74
BiGRU	0.68	0.44	0.53	0.87	0.76	0.81
DNN	0.45	0.37	0.40	0.69	0.31	0.42

TABLE 4: Training and validation accuracy over DS1 and DS2 datasets

Datasets →	**DS1 (balanced)**		**DS2 (unbalanced)**	
Methods ↓	**Train Acc**	**Valid Acc**	**Train Acc**	**Valid Acc**
HCovBi-Caps	0.87	0.80	0.93	0.90
Ding et al. [42]	0.67	0.65	0.88	0.88
Roy et al. [31]	0.73	0.67	0.87	0.87
CNN	0.67	0.66	0.89	0.86
LSTM	0.66	0.65	0.91	0.87
GRU	0.68	0.64	0.90	0.86
BiLSTM	0.80	0.70	0.92	0.89
BiGRU	0.77	0.61	0.91	0.84
DNN	0.67	0.65	0.90	0.88



(a) (b)

FIGURE 2: ROC curve of the proposed HCovBi-Caps model over (a) DS-1 (b) DS-2 datasets

datasets – DS1 (balanced) and DS2(unbalanced) considering *precision*, *recall*, and *f-score*. The analysis of Table 4 also reveals the improved performance of the proposed model considering training and validation accuracy. We also evaluate the performance of the proposed model using the receiver operating characteristics (ROC) curve. It is a metric to evaluate the performance measurement of a classification model at various thresholds. Figure 2 presents the ROC curve of the proposed model for both the datasets – DS1 and DS2. We can observe from the figure that the proposed model shows significantly good results over both datasets. Interestingly, the proposed model shows better results on the DS2 (unbalanced) dataset than DS1 (balanced) dataset.

1) Comparison with State-of-the-Art Methods

We perform the comparative evaluation of the HCovBi-Caps model with two state-of-the-art deep learning methods for hate speech detection. Consequently, we implement the comparison methods from scratch using the instructions defined in the respective papers.

- Ding et al. [42]: In this paper, authors uses a stack of BiGRU and capsule network layers to detect HS on tweets.

- Roy et al. [31]: Authors use deep CNN for detecting HS or NHS on tweets.

Tables 3 and 4 also show that Ding et al. [42], employing the capsule network layer, performs better in comparison to Roy et al. [31], which uses only deep CNN. Therefore, the proposed model integrates capsule network and CNN with BiGRU to gain the advantage of various deep learning components. We can infer from both the tables that the proposed model remarkably outperforms the two state-of-the-art methods. On analysis, we observed that the inclusion of convolutional layers in the proposed model significantly enhances the performance because it helps in encoding spatial and temporal features. The proposed model outperforms the Ding et al. [42] by 16, 13, and 15 points considering *precision*, *recall*, and *f-score* over DS1 (balanced) dataset. HCovBi-Caps also outperforms the Ding et al. [42] by 25, 19, and 22 points considering *precision*, *recall*, and *f-score* over DS2 (unbalanced) dataset. Additionally, we can observe from the fifth row of Table 3 that the HCovBi-Caps model outperforms the Roy et al. [31]. Similarly, 3-5 rows of Table 4 presents the performance evaluation results of the HCovBi-Caps model in comparison to the state-of-the-art approaches considering training and validation accuracy over the two datasets – DS1 and DS2. We can observe from the table that the proposed model performs significantly better than state-of-the-art approaches.

2) Comparison with Baseline Methods

HCovBi-Caps is compared with six baseline methods in this paper. A short detail of hyperparameter values of these baselines is given below:

- CNN: It is used in this paper as one of the baseline methods, in which filter width size and number of filter are 3 and 128, respectively.
- LSTM: It is used in this paper as one of the baseline methods, in which 128 neurons are utilized.
- GRU: It is used as one of the baseline methods, in which 128 neurons are considered.
- BiLSTM: It is a special type of RNN whose functionality lies in both forward and backward directions. In this paper, it is used as one of the baseline methods, in which 128 neurons are used.
- BiGRU: Like BiLSTM, it is also a special type of RNN which is functional in both directions. In this paper, it is used as one of the baseline methods, in which 128 neurons are used.
- DNN: It contains many hidden nodes, and it uses the input data and weights of these nodes. In this paper, it is used as one of the baseline methods, in which two dense layers having 128 neurons in each layer are used.

Tables 3 and 4 show that the bi-directional RNN models, such BiLSTM and BiGRU, perform effectively across all the baseline methods. Such a performance is due to its efficient retrieval of latent sequential features from forward and backward directions. The proposed model outperforms significantly better than all the six baseline methods. For example, the proposed HCovBi-Caps model outperforms CNN baseline by 12, 29, and 23 points considering *precision*, *recall*, and *f-score*, respectively over the DS1(balanced) dataset. Similarly, additional comparative results with other baselines can be observed from tables Tables 3 and 4. Among the baselines, BiGRU shows the best performance over both the datasets DS1 and DS2, considering precision, recall, and f-score, whereas, in terms of training and validation accuracy, BiLSTM shows the best performance over the datasets DS1 and DS2.

V. DISCUSSION

We found in the experimental evaluation that the HCovBi-Caps model significantly outperforms the state-of-the-art and baseline methods over both DS1 and DS2 datasets. In the proposed model, we use BiGRU because it shows the best performance among the baselines. Interestingly, the proposed and comparison approaches show relatively better performance on the unbalanced dataset. Furthermore, the performance difference between the proposed and comparison approaches is also down over the unbalanced dataset. In this section, we present the effects of different neural network hyperparameters on the performance of the HCovBi-Caps model. We perform this empirical investigation to find the optimal value of each hyperparameter so that the final empirical evaluation results are optimal.

A. EFFECT OF NEURAL NETWORK HYPERPARAMETERS

The selection of hyperparameter values is crucial for any deep learning-based model because it affects the model performance. This section experimentally analyzes the impact of four hyperparameters – *activation functions*, *CNN filter size*, *number of BiGRU hidden units*, and *optimization algorithms*, on the performance of the proposed HCovBi-Caps model over DS1 and DS2 datasets considering *f-score* and *accuracy*.

1) Activation Functions

An activation function is crucial considering the activation or non-activation of neurons in any deep learning-based model. We perform the experiments using sigmoid and softmax activation functions to analyze their impact on the classification performance of the HCovBi-Caps model over DS1 and DS2 datasets in terms of *f-score* and *accuracy*. The underlying evaluation results are shown in Figure 3 . We can observe from the figure that the sigmoid performs significantly better than the softmax function over both datasets. One of the key reasons behind such a performance is that sigmoid is effective for binary classification problems. Thus, the proposed model uses the sigmoid activation function.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3143799, IEEE Access

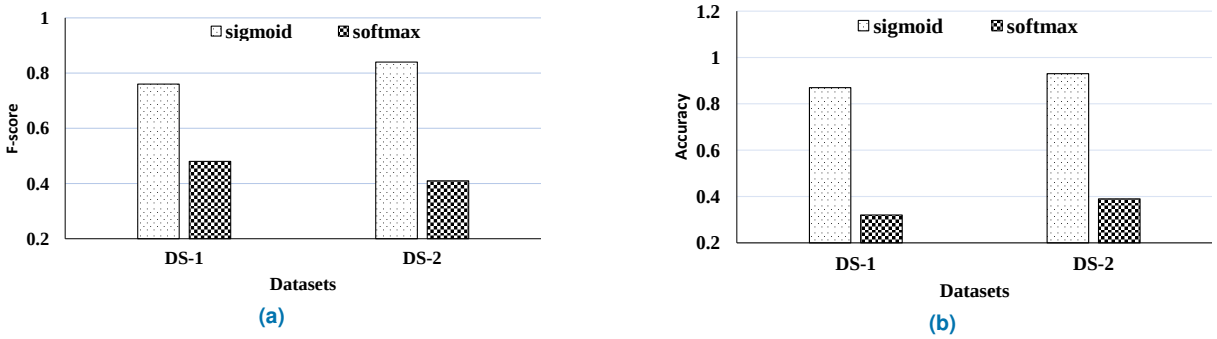


FIGURE 3: Performance evaluation results of HCovBi-Caps model using sigmoid and softmax activation functions over DS1 and DS2 datasets in terms of (a) *f-score* (b) *accuracy*

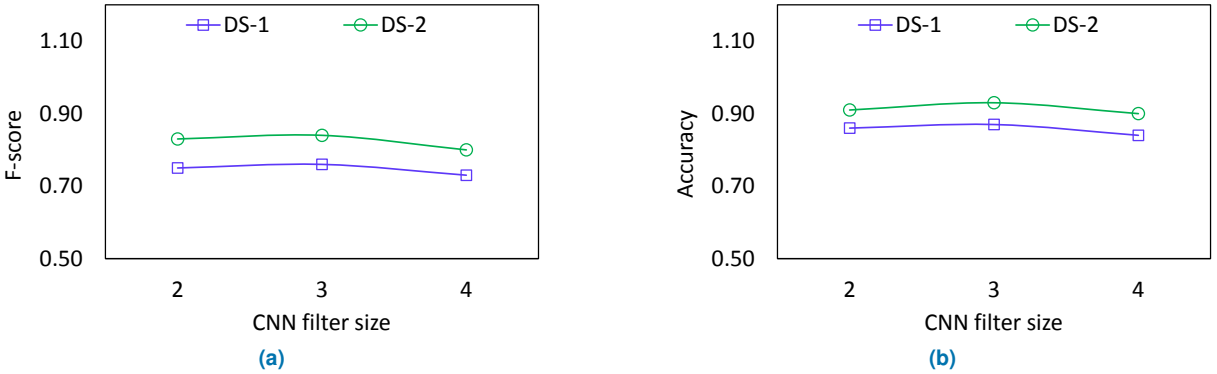


FIGURE 4: Performance evaluation results of HCovBi-Caps model using different CNN filter size – 2, 3, and 4 over DS1 and DS2 datasets in terms of (a) *f-score* (b) *accuracy*

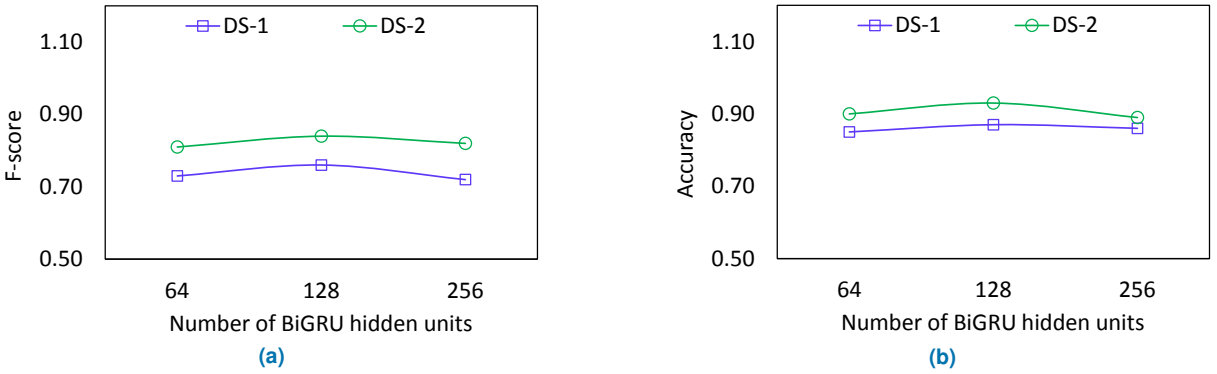


FIGURE 5: Performance evaluation results of HCovBi-Caps model using different BiGRU hidden units – 64, 128, and 256 over DS1 and DS2 datasets in terms of (a) *f-score* (b) *accuracy*

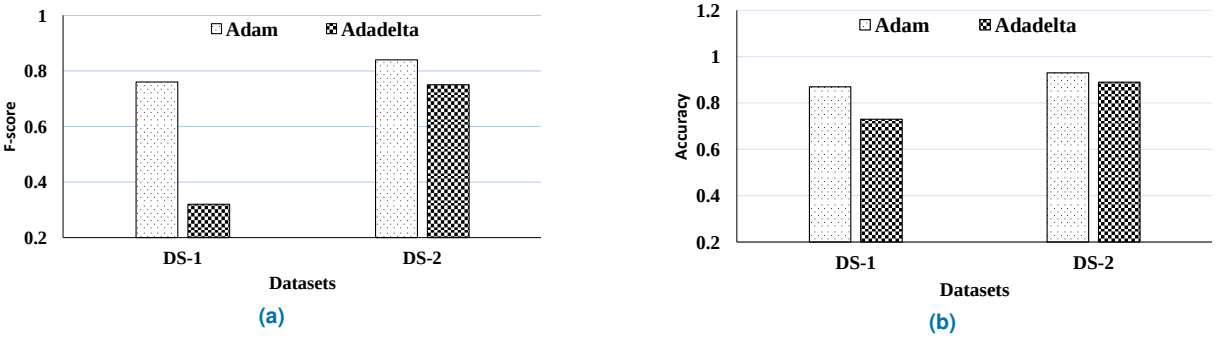


FIGURE 6: Performance evaluation results of HCovBi-Caps model using Adam and Adadelta optimization algorithms over DS1 and DS2 datasets in terms of (a) *f-score* (b) *accuracy*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3143799, IEEE Access

2) CNN Filter Size

The CNN filter size is vital in retrieving important features from CNN layers. Thus, the filter size is important to analyze the classification performance of a deep learning-based model. In this paper, we performed the experimental evaluation using three CNN filter sizes to observe its impact on the performance proposed HCovBi-Caps model. Figure 4 presents the underlying evaluation results using different CNN filter sizes – 2, 3, and 4 over DS1 and DS2 datasets considering *f-score* and *accuracy*. We can observe from the figure that as the CNN filter size increases, the model performance goes down. Furthermore, the model shows the best performance on filter size 3 over both the DS1 and DS2 datasets.

3) BiGRU Hidden Units

The selection of the number of hidden units is another important hyperparameter impacting the classification performance of a deep neural network model. To this end, we performed the experimental evaluation with a different number of hidden units in BiGRU to find its optimal value. Figure 5 presents the impact of different BiGRU hidden units – 64, 128, and 256 on the performance of proposed model over both the DS1 and DS2 datasets considering *f-score* and *accuracy*. The figure demonstrates that the BiGRU with 128 hidden units shows significantly better performance across all datasets. Therefore, the proposed model uses 128 BiGRU hidden units.

4) Optimization Algorithms

Like the already discussed hyperparameters, an optimization algorithm also has a significant effect on the performance of a deep learning-based classification model. We performed the experiments using different optimization algorithms to analyze their impact on the classification performance of the HCovBi-Caps model over both DS1 and DS2 datasets. Figure 6 presents the underlying results using Adam and Adadelta algorithms over both the datasets considering *f-score* and *accuracy*, respectively. We can observe from the figure that *Adam* performs better than Adadelta across all datasets. Thus, the proposed model uses the *Adam* as an optimization algorithm.

B. EFFECT OF CAPSULE NETWORK HYPERPARAMETERS

Accurate functioning of capsule network relies on various hyperparameters. This section presents the effect of three hyperparameters – *number of capsules*, *dimension of capsule*, and n*umber of routing iterations* on the performance of HCovBi-Caps model over all datasets considering *f-score* and *accuracy*.

1) Number of Capsules

The total number of capsules in a capsule network highlights the role of neurons at each layer and affects the performance of the underlying model. To this end, the performance of the HCovBi-Caps model is investigated using the different number of capsules over both datasets. Figure 7 presents experimental evaluation results representing the effect of number of capsules – 4, 5, and 6 on the performance of HCovBi-Caps model over both the datasets considering *f-score* and *accuracy*. We found on analysis that the model performance is inversely proportional to the number of capsules. Furthermore, we can observe from the figure 7 that HCovBi-Caps performs significantly better using 4 capsules over both datasets. This result justifies the selection of 4 capsules in the proposed model.

2) Dimension of Capsule

The capsule dimension is another critical hyperparameter in a capsule network. It controls the length of the output vector of a capsule. We perform experiments using different capsule dimensions to analyze the classification performance of the HCovBi-Caps model. Figure 8 presents the effect of various capsule dimensions – 2, 4, and 8) on HCovBi-Caps model over both datasets considering *f-score* and *accuracy*. The figure demonstrates that the proposed model performs best on capsule dimension 4. Further observation is that neither small nor larger capsule dimension is effective.

3) Number of Routing Iterations

Routing iterations connect the capsules of consecutive layers in a capsule network. We performed the experimental evaluation with the different number of routing iterations to analyze the classification performance of the HCovBi-Caps model over both datasets. Figures 9 present the results with different number of routing iterations – 2, 3, and 4 representing its impact on the classification performance of HCovBi-Caps over both datasets considering *f-score* and *accuracy*. We can observe from the figure that HCovBi-Caps performs significantly better with 3 routing iterations.

VI. CONCLUSION

This study has presented a novel deep neural network model, HCovBi-Caps, integrating the convolutional, BiGRU, and capsule network layers for hate speech detection. Unlike existing models, the HCovBi-Caps incorporates the contextual information at different orientations using the capsule network. We evaluated the proposed model over two Twitter-based benchmark datasets – DS1(balanced) and DS2(unbalanced) to classify hate speech from general text. The proposed model shows the best performance over DS2(unbalanced) with values of 0.90, 0.80, and 0.84 considering *precision*, *recall*, and *f-score*, respectively. The proposed model shows the best performance on the unbalanced dataset considering accuracy. The proposed model has shown significantly improved performance over state-of-the-art and baseline methods. We have further investigated the impact of various hyperparameters of neural and capsule networks to analyze the efficacy of our proposed HCovBi-Caps model.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3143799, IEEE Access

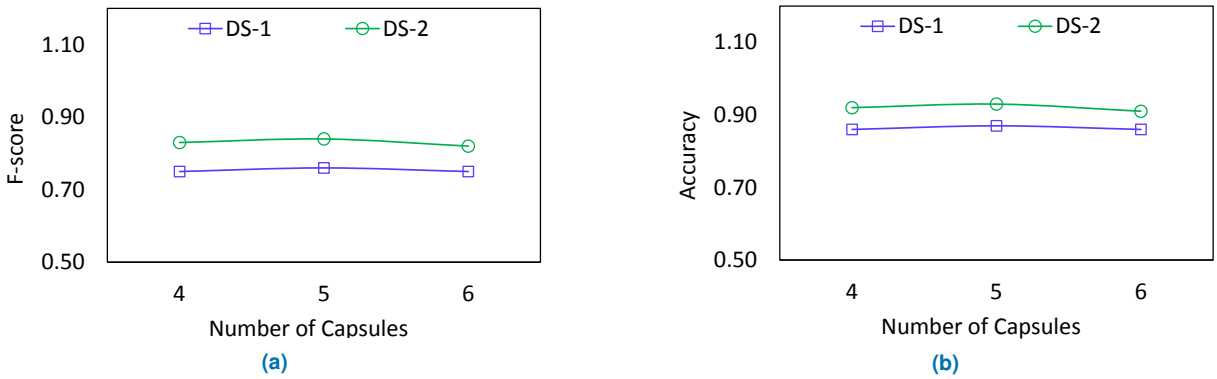


FIGURE 7: Performance evaluation results of HCovBi-Caps model using different number of capsules – 4, 5, and 6 over DS1 and DS2 datasets in terms of (a) *f-score* (b) *accuracy*

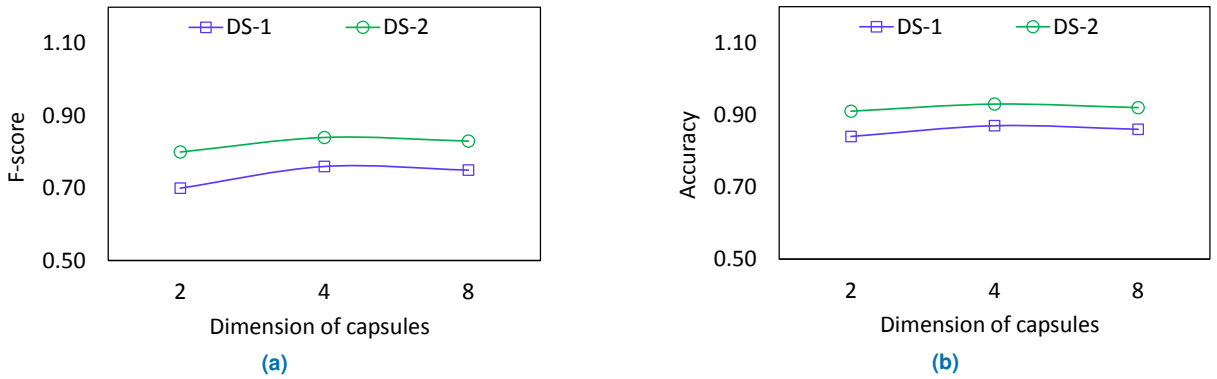


FIGURE 8: Performance evaluation results of HCovBi-Caps model using different capsule dimensions – 4, 8, and 16 over DS1 and DS2 datasets in terms of (a) *f-score* (b) *accuracy*

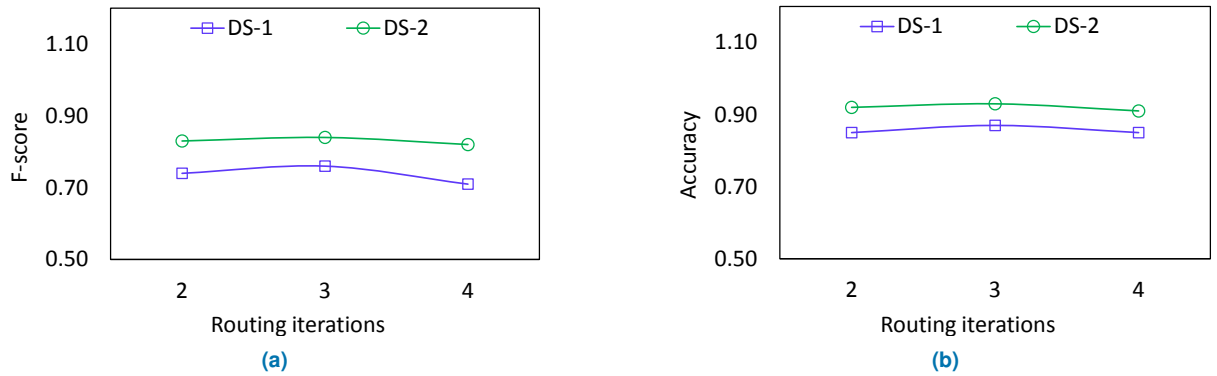


FIGURE 9: Performance evaluation results of HCovBi-Caps model using different routing iterations – 2, 3, and 4 over DS1 and DS2 datasets in terms of (a) *f-score* (b) *accuracy*

VII. LIMITATIONS AND FUTURE WORKS

The proposed HCovBi-Caps model detects hateful content with different contextual orientations. However, we can further improve it in detecting hate content considering different contextual semantic. Further, HCovBi-Caps does not exploit the sentiment and users' profile-related features, which may be effective. We can also evaluate HCovBi-Caps over more diverse datasets. The HCovBi-Caps model detects the hate propagated in text only. Therefore, it can be extended to a multi-model approach for hate speech detection. The extension of HCovBi-Caps to classify the hateful multi-lingual and code-mixed content is also another direction of research. The contextual information, which triggers controversy and hates on OSNs, will also be investigated.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University for funding this work through Research Group no. RG-21-07-08.

REFERENCES

[1] P. K. Jain, V. Saravanan, and R. Pamula, "A hybrid cnn-lstm: A deep learning approach for consumer sentiment analysis using qualitative user-generated contents," ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 20, no. 5, pp. 1–15, 2021.

[2] P. K. Jain, R. Pamula, and G. Srivastava, "A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews," Computer Science Review, vol. 41, no. 1, 2021.

[3] M. Fazil and M. Abulaish, "A hybrid approach for detecting automated spammers in twitter," IEEE Transactions on Information Forensics and Security, vol. 13, no. 11, pp. 2707–2719, 2018.

[4] M. Abulaish and M. Fazil, "Socialbots: Impacts, threat-dimensions, and defense challenges," IEEE Technology and Society Magazine, vol. 39, no. 3, pp. 52–61, 2020.

[5] M. Abulaish, A. Kamal, and M. J. Zaki, "A survey of figurative language and its computational detection in online social networks," ACM Transaction on the Web, vol. 14, no. 1, pp. 1–52, Jan. 2020.

[6] M. Abulaish and A. Kamal, "Self-deprecating sarcasm detection: An amalgamation of rule-based and machine learning approach," in Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI' 18), Santiago, Chile. IEEE, Dec. 2018, pp. 574–579.

[7] A. Kamal and M. Abulaish, "An lstm-based deep learning approach for detecting self-deprecating sarcasm in textual data," in Proceedings of the 16th International Conference on Natural Language Processing (ICON' 19), Hyderabad, India. NLPAI, 2019, pp. 201–210.

[8] A. Kamaal and M. Abulaish, "Cat-bigru: Convolution and attention with bi-directional gated recurrent unit for self-deprecating sarcasm detection," Cognitive Computation, pp. 1–19, 2021.

[9] A. Kamal and M. Abulaish, "Self-deprecating humor detection: A machine learning approach," in Proceedings of the 16th International Conference of the Pacific Association for Computational Linguistics (PACLING' 19), Hanoi, Vietnam. Springer, 2019, pp. 483–494.

[10] P. K. Jain, R. Pamula, and E. A. Yekun, "A multi-label ensemble predicting model to service recommendation from social media contents," The Journal of Supercomputing, vol. 66, no. 1, pp. 1–20, 2021.

[11] P. K. Jain, E. A. Yekun, R. Pamula, and G. Srivastava, "Consumer recommendation prediction in online reviews using cuckoo optimized machine learning models," Computers Electrical Engineering, vol. 95, no. 10, pp. 1–10, 2021.

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

[13] E. Wulczyn, N. Thain, and L. Dixon, "Ex machina: Personal attacks seen at scale," in Proceedings of the 26th International Conference on World Wide Web. Perth, Australia: ACM, 2017, pp. 1391–1399.

[14] A. R. Gover, S. B. Harper, and L. Langton, "Anti-asian hate crime during the covid-19 pandemic: Exploring the reproduction of inequality," American Journal of Criminal Justice, vol. 45, no. 7, pp. 647—-667, 2020.

[15] Z. Zhang, D. Robinson, and J. Tepper, "Detecting hate speech on twitter using a convolution-gru based deep neural network," in Proceedings of the European Semantic Web Conference. Heraklion, Greece: Springer, Cham, 2018, pp. 745–760.

[16] A.-M. Founta, D. Chatzakou, N. Kourtellis, J. Blackburn, A. Vakali, and I. Leontiadis, "A unified deep learning architecture for abuse detection," in Proceedings of the 11th International Conference on Web Science. Massachusetts USA: ACM, 2019, pp. 105–114.

[17] B. Wang and H. Ding, "Ynu nlp at semeval-2019 task 5: Attention and capsule ensemble for identifying hate speech," in Proceedings of the 13th International Workshop on Semantic Evaluation, (SemEval' 19), Minneapolis, Minnesota, USA. ACL, 2019, pp. 529–534.

[18] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in Proceedings of the 11th International AAAI Conference on Web and Social Media, (ICWSM' 17), Montréal, Canada. AAAI, May 15–18, 2017, pp. 512–515.

[19] P. K. Jain, R. Pamula, and S. Ansari, "A supervised machine learning approach for the credibility assessment of user-generated content," Wireless Personal Communications, vol. 118, no. 4, pp. 2469–2485, 2021.

[20] W. Warner and J. Hirschberg, "Detecting hate speech on the world wide web," in Proceedings of the 2012 Workshop on Language in Social Media. Montreal, Canada: ACL, 2012, pp. 19–26.

[21] I. Kwok and Y. Wang, "Locate the hate: Detecting tweets against blacks," in Proceedings of the 27th AAAI Conference on Artificial Intelligence. Bellevue, USA: AAAI, 2013, pp. 1621–1622.

[22] 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.

[23] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, "Hate speech detection with comment embeddings," in Proceedings of the 24th International Conference on World Wide Web Companion. Florence, Italy: ACM, 2015, pp. 29–30.

[24] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," Proceedings of Machine Learning Research, vol. 32, no. 2, pp. 1188–1196, 2014.

[25] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? predictive features for hate speech detection on twitter," in Proceedings of the NAACL-HLT. California, USA: ACL, 2016, pp. 88–93.

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

[27] P. Badjatiya, S. Gupta1, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," in Proceedings of the 26th International Conference on World Wide Web Companion. Perth Australia: ACM, 2017, pp. 759–760.

[28] J. H. Park and P. Fung, "One-step and two-step classification for abusive language detection on twitter," in Proceedings of the First Workshop on Abusive Language Online. Vancouver, Canada: ACL, 2017, pp. 41–45.

[29] B. Gamback and U. K. Sikdar, "Using convolutional neural networks to classify hate-speech," in Proceedings of the First Workshop on Abusive Language Online. Vancouver, Canada: ACL, 2017, pp. 85–90.

[30] R. Cao, R. K.-W. Lee, and T.-A. Hoang, "Deephate: Hate speech detection via multi-faceted text representations," in Proceedings of the 12th International Conference on Web Science. Southampton, UK: ACM, 2020, pp. 11–20.

[31] P. K. Roy, A. K. Tripathy, T. K. Das, and X. Gao, "A framework for hate speech detection using deep convolutional neural network," IEEE Access, vol. 8, pp. 204 951–204 962, 2020.

[32] S. Kamble and A. Joshi, "Hate speech detection from code-mixed hindi-english tweets using deep learning models," in Proceedings of 15th International Conference on Natural Language Processing. Patiala, India: ACL, 2018, pp. 155–160.

[33] E. W. Pamungkas, V. Basile, and V. Patti, "A joint learning approach with knowledge injection for zero-shot cross-lingual hate speech detection," Information Processing & Management, vol. 58, no. 4, pp. 1–19, 2021.

[34] M. Mozafari, R. Farahbakhsh, and N. Crespi, "A bert-based transfer learning approach for hate speech detection in online social media," in Proceedings of International Conference on Complex Networks and their Applications. Lisbon, Portugal: Springer, Cham, 2019, pp. 928–940.

[35] K. Miok, B. Skrlj, D. Zaharie, and M. Robnik-Sikonja, "To ban or not to ban: Bayesian attention networks for reliable hate speech detection," Cognitive Computation, vol. 21, no. 1, pp. 1–19, 2021.

[36] Z. Mossie and J.-H. Wang, "Vulnerable community identification using hate speech detection on social media," Information Processing & Management, vol. 57, no. 3, pp. 1–16, 2020.

[37] P. Fortuna, J. Soler-Company, and L. Wanner, "How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets?" Information Processing & Management, vol. 58, no. 3, pp. 1–17, 2021.

[38] G. E. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with em routing," in Proceedings of the International Conference on Learning Representations (ICLR' 18), Vancouver, BC Canada, April 30–May 03, 2018, pp. 44–51.

[39] D. K. Jain, R. Jain, Y. Upadhyay, A. Kathuria, and X. Lan, "Deep refinement: Capsule network with attention mechanism-based system for text classification," Neural Computing and Applications, vol. 32, no. 7, pp. 1839–1856, 2020.

[40] U. Bhattacharjee, "Capsule network on social media text: An application to automatic detection of clickbaits," in Proceedings of the 11th International Conference on Communication Systems & Networks (COMSNETS' 19), Bengaluru, India. IEEE, January 7–11, 2019, pp. 473–476.

[41] Y. Du, X. Zhao, M. He, and W. Guo, "A novel capsule based hybrid neural network for sentiment classification," IEEE Access, vol. 7, pp. 39 321–39 328, 2019.

[42] 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 Proceedings of the 13th International Workshop on Semantic Evaluation, (SemEval' 19), Minneapolis, Minnesota, USA. ACL, June 6–7, 2019, pp. 535–539.

[43] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in Artificial Neural Networks and Machine Learning, (ICANN' 11), T. Honkela, W. Duch, M. Girolami, and S. Kaski, Eds. LNCS, Springer, 2011, pp. 44–51.

[44] G.-A. Vlad, M.-A. Tanase, C. Onose, and D.-C. Cercel, "Sentence-level propaganda detection in news articles with transfer learning and bert-bilstm-capsule model," in Proceedings of the 2nd Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda. ACL, 2019, pp. 148–154.

[45] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in Proceedings of the Advances in Neural Information Processing Systems (NIPS' 17), Long Beach, CA, USA, December 4–9, 2017, pp. 3856–3866.

[46] A.-M. Founta, C. Djouvas, D. Chatzakou, I. Leontiadis, J. Blackburn, G. Stringhini, A. Vakali, and M. S. and Nicolas Kourtellis, "Large scale crowdsourcing and characterization of twitter abusive behavior," in Proceedings of the 12th International AAAI Conference on Web and Social Media, (ICWSM' 18), Stanford, California, USA. AAAI, June 25–28, 2018, pp. 491–500.

SHAKIR KHAN received his BSc, MSc and PhD in computer science in 1999, 2005 and 2011 respectively. He is member of the International Association of Online Engineering (IAOE) and IEEE; He is currently working as Associate Professor at College of Computer and Information Sciences in Imam Mohammad Ibn Saud Islamic University, Riyadh (Saudi Arabia). His research interest is Big Data, Data Science, Data Mining, Machine Learning, Internet of Things (IoT), and e-learning, Artificial Intelligence, Emerging Technology, Open-Source Software, Library Automation and Mobile / Web Application. He published many research papers in international journals and conferences in his research domain. He has around 15 years of teaching, research and IT experience in India and Saudi Arabia. Dr. Khan is teaching bachelor and master degree courses in the college of computer at Imam University. He is reviewer for many international journals

ASHRAF KAMAL has received the Ph.D. degree in Computer Science from Jamia Millia Islamia, New Delhi. He is currently working as Data Scientist at ACL Digital, Bengaluru. He has qualified UGC-NET in 2014 and his research interests include text mining, machine learning, data science, social computing, and information retrieval. He has published research papers in reputed SCI-indexed journals, including ACM Transactions and CORE ranking conferences.

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 Postdoctoral Research Associate at Department of Comptuter Engineering, Qatar University, Qatar. He has published over 14 research articles including two papers in IEEE Transactions on Information Forensics and Security. His research interests include data science, social computing, and data-driven cyber security.

MOHAMMED ALI ALSHARA received BSc from Imam Mohammad Ibn Saud Islamic University in 2002 and MSc in 2008 and PhD in 2016 from University of Texas. He is currently working as Assistant Professor and vice dean in College of Computer and Information Sciences (CCIS), Imam Mohamad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia. His research interest includes: Data and content Analysis, Knowledge Management, Information Retrieval etc.

VINEET SEJWAL has received his PhD degree in Computer Science from Jamia Millia Islamia (A Central University), New Delhi, India. He has qualified one of most prestigious Indian exams in Computer Science and Engineering, GATE. His research interests include Recommender System, Text Mining, Machine Learning, and Information Retrieval. He has published research papers in reputed SCI-indexed journals.

REEMIAH MUNEER ALOTAIBI is currently an Assistant Professor in College of Computer Sciences and Information at Imam Muhammad bin Saud Islamic University, Saudi Arabia. She received her BSc from Imam University, MSc, and PhD in "Information Management" from Leeds Beckett University, UK. Moreover, she has some courses such as Professional development diploma in project management in Midlands Academy of Business and Technology.UK. She has also authored articles published in scientific journals, book chapters, and conference materials, and has acted as reviewer for International Journal of Civic Engagement and Social Change (IJCESC), IGI Global.

ABDUL RAUF BAIG (Member, IEEE) received the B.E. degree in electrical engineering from the NED University of Engineering and Technology, Karachi, Pakistan, in 1987, the Diplôme de Spécialisation degree in computer science from Supélec, Rennes, France, in 1996, and the Ph.D. degree in computer science from the University of Rennes 1, Rennes, in 2000. He was with the National University of Computer and Emerging Sciences, Islamabad, Pakistan, as a Faculty Member, an Assistant Professor, an Associate Professor, and a Professor, from 2001 to 2010. Since 2010, he has been a Professor with Imam Mohammad Ibn Saud Islamic University, Riyadh, Saudi Arabia. He has more than 100 publications in journals and international conferences. His research interests includes machine learning, data mining, and evolutionary algorithms. He is a member of the IEEE Computational Intelligence Society.

SALIHAH ALQAHTANI Salihah Alqahtani is currently pursuing master degree in Information Management at College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University, Riyadh, Saudi Arabia. Her research interest are data science and big data.