SEHC: A Benchmark Setup to Identify Online Hate Speech in English

Soumitra Ghosh, Asif Ekbal, Pushpak Bhattacharyya, Tista Saha, Alka Kumar, and Shikha Srivastava

***Abstract*— Thanks to the digital age, online speech and information may now be disseminated anonymously without regard for repercussions. Regulators face a unique problem with social media platforms because of the speed and volume of material and the lack of editorial supervision. The existing datasets on hate speech or offensive language identification lack diversity in the dataset's content. In this article, we create a *multi-domain hate speech corpus (MHC)* of English tweets that includes hate speech against religion, nationality, ethnicity, and gender in general and cover diverse domains, such as current affairs, politics, terrorism, technology, natural disasters, and human/drugs trafficking. Each instance in our dataset is manually annotated as *hate* or *non-hate*. We use the existing state-of-the-art models and present a *stacked-ensemble-based hate speech classifier (SEHC)* to identify hate speech from Twitter data. Our results indicate that the proposed method may serve as a strong baseline for future studies using this dataset. (The dataset is available at https://www.iitp.ac.in/~ai-nlp-ml/resources.html#MHC.)**

***Index Terms*— Annotated dataset, deep learning, ensemble learning, hate speech, neural networks, offensive language, social media, tweet classification.**

I. Introduction

MICROBLOGGING services have changed how individuals think, interact, behave, learn, and go about their daily lives. Due to a lack of regulation, the vast volume of user-generated information reflects the offline world more precisely than official news sources. As a result, social media has become an appealing medium for anybody looking for unbiased information. Crowds, riots, and mobs have a wide range of effects on societies. Crowd behavior may be a catalyst for social change. They may have severe short-term repercussions such as murders, slaughters, or material destruction, illustrating how fragmented our communities may be. Hate speech on the internet is a form of communication directed toward a person or group based on race, religion, ethnic origin, sexual orientation, or gender.

Manuscript received 29 July 2021; revised 26 December 2021 and 18 February 2022; accepted 20 February 2022. Date of publication 21 March 2022; date of current version 3 April 2023. This work was supported by the Centre for Development of Telematics (C-DOT), India. The work of Asif Ekbal was supported by the Young Faculty Research Fellowship (YFRF), through the Visvesvaraya Ph.D. Scheme for Electronics and IT, Ministry of Electronics and Information Technology (MeitY), Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia). *(Corresponding author: Asif Ekbal.)*

Soumitra Ghosh, Asif Ekbal, and Pushpak Bhattacharyya are with the Department of Computer Science and Engineering, IIT Patna, Patna 801106, India (e-mail: asif@iitp.ac.in).

Tista Saha, Alka Kumar, and Shikha Srivastava are with the Centre for Development of Telematics (C-DOT), New Delhi 110030, India.

Digital Object Identifier 10.1109/TCSS.2022.3157474

Since the Twitter Revolution in 2009, social media has become one of the most powerful forces in influencing society's operating environment by reducing the time it takes for information to disseminate and expanding the audience to whom it is available. Since its inception, the popularity of Twitter has been ever-increasing among other social media networks. Twitter is notably popular in USA, where the microblogging site had a 73.2 million strong user base as of April 2021. With 54.15 and 18.8 million users, Japan and India were placed second and third, respectively.[1] Social media is the primary means of instigating and organizing a crowd, as well as maintaining and fueling its intensity. They also extend the issue's reach beyond ideological, cultural, and geopolitical boundaries. As a result, the government must recognize the hazards of hate speech and ensure that proper rules are in place to combat the problem.

Combating hate speech and false news has become a significant problem for governments worldwide. But this is not simply a sociological problem; it is also a technological one. In Christchurch, New Zealand, in 2019, a terrorist opened fire in two mosques, killing at least 49 worshipers and injuring scores more.[2] Even before the event, a posting on the anonymous message board 4chan[3] seemed to foreshadow the catastrophe. The post directed readers to a Facebook page with a live stream of the event and an 87-page manifesto packed with anti-immigrant anti-Muslim sentiments.

Social media sites such as Facebook, Twitter, and WhatsApp have been frequently used to transmit false pictures and videos, and text messages that have instigated further violence globally. In August, two individuals were assaulted and burned to death in Mexico after reports of people kidnapping youngsters and harvesting their organs circulated on WhatsApp. In India, crowds killed two dozen individuals after a video made to seem like an abduction went viral on WhatsApp. In Sri Lanka, the government temporarily shut down WhatsApp and Facebook as part of an effort to stop ethnic conflicts.

Most of the existing datasets on hate speech suffer from the following limitations.

1) *Limited Diversity in Data Instances:* Majority of the datasets on hate speech are curated from a collection of social media posts over a certain period, primarily using specific hate keywords or phrases. This approach

[1] https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/

[2] https://www.bbc.com/news/world-asia-53861456

[3] https://www.4chan.org/

benefits from hitting hate posts correctly but at the cost of limited diversity in the domain of the collected data.

2) *Under-Represented Hate Class:* Although hate content is prevalent in social media in recent times, gathering random social media posts and annotating them for *hate* and *non-hate* occurrences frequently result in a highly skewed distribution across classes, with a sparse representation of the *hate* class.

Hate speech across different domains varies in the vocabularies [1], i.e., data distributions are different across domains. For instance, one set of hate speech posts revolving around the act of terrorism would contain adjectives such as "attack," "bombing," or "killing," and the other "buying," "selling," or "funding," etc. While mainstream machine learning techniques can be compelling in narrow domains, they typically suffer from poor transfer learning ability and system reusability. One solution could be to learn a different system for each domain. However, this would imply a considerable cost to annotate training data for many domains and prevent us from exploiting the information shared across domains. An alternative strategy is to learn a single unified system on a dataset created from multiple domains that will discover intermediate abstractions that are shared and meaningful across domains. To this end, our study attempts to address the lack of variation in the content of existing hate speech identification corpora.

This article introduces a multi-domain hate speech corpus (MHC) of English tweets with balanced distribution of instances over the *hate* and *non-hate* classes. Distinct from the existing hate speech datasets, our dataset is compiled from diverse domains such as politics, terrorism, technology, natural disasters, and human/drugs trafficking. Moreover, the dataset consists of 10 242 instances retrieved from Twitter with a balanced distribution of instances over the *hate* and *non-hate* classes (5121 instances in each category). We use the concept of entropy as defined by Shannon [2] to train our proposed system. Entropy defines the amount of information in a message (M where $M = m_1, m_2, \ldots, m_n$) as a function of the likelihood $[p(m_i)]$ of each conceivable message (m_i) occurring, and it is realized as follows:

$$\sum_{i=1}^{n} -p(m_i)\log_2(p(m_i)). \tag{1}$$

In addition, this work includes a thorough qualitative and quantitative analyses of the generated dataset many baseline experiments with various categorization methods.

The following are the key contributions of this work.

1) We introduced a manually labeled hate speech dataset (MHC) of 10 242 English Tweets with 5121 instances in each *hate* and *non-hate* classes.
2) We developed a stacked-ensemble-based hate speech classifier (*SEHC*) on our introduced dataset, built from several state-of-the-art deep learning networks.

The rest of the article is structured as follows: Section II contextualizes the work provided in the article by describing similar work. Section III describes the creation of the manually labeled hate speech dataset, including the annotation rules, inter-annotator agreement, and a quantitative description of the dataset. In Section IV, we discuss the various baseline experiments performed on the dataset using different classification models. Section V describes the results and analysis of the performance of the developed systems. Finally, we summarize the findings and future work in Section VII.

II. Related Work

Hate speech study has exploded in popularity in recent years, and several datasets have been introduced to facilitate research in this domain. The majority of the datasets are developed from Twitter data, labeled manually or by automated approaches, or using machine-learning-based filtering techniques such as taking help of frequently used negative keywords. Violent occurrences in the real world might lead to an uptick in hate speech on the internet [3]. Various approaches for hate speech identification have been proposed to combat this [4], [5].

In recent years, several studies have introduced various sentiment and offensive datasets. Poria *et al.* [6] introduced the Multimodal EmotionLines Dataset (MELD) that contains about 13 000 utterances from 1433 dialogs from the TV series Friends. Each utterance is annotated with emotion and sentiment labels and encompasses audio, visual, and textual modalities. Zhang *et al.* [7] presented a new conversational dataset for interactive sentiment analysis containing manually labeled 2214 multi-turn English conversations collected from various online communication service websites. The semantic evaluation (SEMEVAL) 2020 Task 8: Memotion Analysis [8] task released approximately 10 k memes manually annotated with sentiment, emotion, and their corresponding intensity labels. Zhang *et al.* [9] explored the use of quantum theory (QT) to model the multimodal sentiment analysis task. Extensive experiments were conducted on two large-scale datasets collected from Getty Images and Flickr photo-sharing platform. The Toxic Comment dataset introduced by Kaggle as a part of the Toxic Comment Classification Challenge consists of 150 k toxic behavior annotated Wikipedia comments. Waseem and Hovy [10] presented a collection of 16 k tweets that were categorized as racist, sexist, or neither. According to the degree of hostile intent, 9 k tweets were manually annotated for the presence of harmful speech and presented in [11]. Kumar *et al.* [12] introduced the second workshop on trolling, aggression and cyberbullying (TRAC2) shared task on aggression and misogyny identification. The dataset for this task contains 20 k YouTube comments annotated at two levels—aggression and misogyny.

Masud *et al.* [13] attempted to forecast the beginning and spread of hate speech on Twitter and found numerous critical elements such as exogenous information and user topic affinity that regulate the transmission of hate speech. The authors proposed retweeter identifier network with exogenous attention (RETINA), a neural framework that combines extra Twitter information (in the form of news) with an attention mechanism to anticipate possible retweeters for every given tweet. Golbeck *et al.* [14] presented a six-category fine-grained labeled dataset that focuses on online abuse on Twitter. Founta *et al.* [15] presented a large dataset of 100 k tweets; however, only 5% of the total tweets belonged to the *hate* class, which is a significant limitation of this dataset.

Chatzakou *et al.* [16] introduced a distinct dataset of hate tweets, which, unlike other hate datasets, focuses on the behavior of hate-related Twitter users rather than the content of the tweets. Based on the analysis of the users, 1.5 k users were classified as spammers (31.8%) or regular users (60.3%). Bullies (4.5%) and aggressors (4.5%) account for a tiny percentage of users (3.4%). Moon *et al.* [17] introduced 9.4 k manually annotated entertainment news comments collected from a widely used online news site in Korea for identifying Korean toxic speech.

Some research focuses on different forms of hate speech, such as the work in [18] focuses on identifying anti-Semitic postings against any other kind of hate speech. Kwok and Wang [19] used supervised learning methods to distinguish between racist and non-racist hate tweets. Similar works were proposed in [20] and [21] where deep-learning-based end-to-end systems were developed to racist and sexist tweets.

Basak *et al.* [22] offered a potential solution to the problem of online public shaming in Twitter by classifying shaming remarks into six categories, selecting relevant characteristics, and building a set of classifiers to identify them. Given the possible negative consequences of community tweets during disasters, Rudra *et al.* [23] developed an event-independent classifier to discriminate communal tweets from noncommunal tweets. Findings show that a high fraction of community tweets are made by famous individuals (with tens of thousands of followers), most of whom are involved in journalism and politics. Other ideas [24]–[26] focus on the annotation of hate speech rather than disparaging or offensive terms in texts. For hate speech identification in Twitter data, Ayo *et al.* [27] developed an end-to-end hate speech classifier that leveraged the effectiveness of word-level term frequency-inverse document frequency (TF-IDF) features and sentence-level features (extracted using recurrent neural networks) with an upgraded cuckoo search neural network. Ayo *et al.* [28] developed a probabilistic clustering approach for identifying hate speech in tweets. To solve the data sparsity problem that exists among the existing hate speech datasets, Kapil and Ekbal [29] leveraged multi-domain information to produce global representative features and proposed a convolution neural network (CNN)-based multi-task system for hate speech detection. Huang *et al.* [30] introduced a multilingual hate speech Twitter corpus with inferred author demographic labels such as sex, age, gender, and country. In a conversational setting, the datasets presented in [31] aims to generate automated responses to online hate speech conversations as an intervention measure. The HateXplain dataset introduced in [32] covers several aspects of hate speech, such as distinguishing among hate, offensive, and normal posts; identification of the target community; and the rationales behind labeling a post as hate/offensive/normal. More recently, Prasad *et al.* [33] proposed the character text image classifier (CTIC), a multimodal hate speech classifier based on bidirectional encoder representations from transformers (BERT), capsule network, and EfficientNet and incorporating four modalities: word embeddings, character embeddings, sentence embeddings, and pictures. The authors trained the model using various sampling strategies and selective training on the MMHS150K Twitter dataset, containing texts and related photographs. Castano-Pulgarín *et al.* [34] presented a systematic review of the previous studies on online hate speech and associated tasks and discussed the various methods to assess them.

Kennedy *et al.* [35] presented a dataset that includes posts from Twitter, Reddit, and The Guardian. The dataset contained 4136 instances categorized as harassment and 16 296 as non-harassment. According to experts, annotating hate speech is a challenging endeavor owing to the data annotation process. Waseem and Hovy [10] investigated the impact of annotator knowledge of hate speech on hate speech classifiers. Motivated by some of these works and the challenges posed due to the lack of diversity in the instances and skewed distribution of the hate speech datasets, we present a balanced dataset of English tweets annotated with *hate* and *non-hate* labels, covering diverse domains.

III. Corpus Creation

We briefly discuss the data generation process in this section, starting with data collection followed by pre-processing, annotation, and inter-annotator dependability.

A. Data Collection

To retrieve tweets from Twitter's standard search application programming interface (API),[4] we use the Twython[5] python package (wrapper). Between January 2018 and May 2020, tweets were retrieved using a set of domain-specific keywords[6] and their combinations. Hundreds of thousands of tweets were gathered first and then filtered using multiple lexicons to enhance the coverage of the affect-oriented tweets. To ensure diversity in the dataset, we retrieved tweets using the various combinations of keywords considered for scraping. Specifically, certain non-affective keywords such as *global, women, humanity, state, police, and world* are associated with some affective keywords such as *crime, terrorism, attack, protest, threat, and cyclone* to extract tweets that are diverse in terms of both demographics and domains. For example, extracting tweets with the words *global* and *terrorism*, *threat* and *women*, *world* and *technology*.

B. Data Preprocessing

We conduct some simple pre-processing steps to eliminate noise from data before using it for our experiments (removing URLs,[7] mentions, non-American Standard Code For Information Interchange (ASCII) characters, punctuation except "." "!," and "?," converting to lower case, etc.). Smileys are substituted with their meanings (e.g., *:-(* is replaced with *sad*, and *<3* is replaced with *Love*). Frequently used contractions are replaced with their full forms (e.g., *can't* is replaced with *cannot*, and *he's* is substituted with *he is*). Each tweet is sentence tokenized considering "." "!," and "?" as the sentence delimiters.

[4]https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets

[5]https://pypi.org/project/twython/

[6]The exhaustive list of keywords are: *casteism, counter-terrorism, cyber-crime, cyclone, terrorism, communal dispute, crime, cyber-security, human trafficking, narcotics, technology, weapons, terror, insurgency, attack, bombing, threat, cyber-attack, earthquake, tsunami, business, and humanity.*

[7]Beautifulsoup Python library: https://pypi.org/project/beautifulsoup4/

TABLE I

SAMPLES OF SOME ANNOTATED PRE-PROCESSED TWEETS FROM MHC

Tweet	Label
italian congress and marxist maoist communist terrorist party are funded by pakistani isi and chinese mss to divide hindus and india	*hate*
america will be legally clean it will also help minimize crime and terrorism every one	*non-hate*
they also say grow some balls you are such a girl or in goes india it like stop effin gender inequality.	*hate*
The border patrol narcotics units need bullet proof vests and night vision goggles	*non-hate*
okay he did a crime and he was hanged but in india rapists are protected just like you guys cant even punish the culprits	*hate*

TABLE II

FIFTEEN MOST FREQUENTLY OCCURRING WORDS IN *Hate* AND *Non-Hate* CLASSES (NOT CONSIDERING STOPWORDS)

non-hate		*hate*	
Words	**Freq.**	**Words**	**Freq.**
caa	1021	india	1994
terrorism	963	terrorism	1606
india	865	caa	1022
terrorist	832	terrorist	864
threat	604	people	815
kashmir	602	crime	802
people	592	kashmir	674
protest	555	technology	648
nrc	519	human	646
attack	498	threat	627
terror	470	protest	566
pakistan	405	like	564
like	381	attack	544
muslims	337	terror	534
crime	310	nrc	521

C. Data Annotation

Three annotators manually annotate the whole dataset, labeling each tweet as *hate* or *non-hate*. Keeping in mind that annotators may disagree on the labels for some tweets, we assigned three annotators[8] for the task so that a clear majority vote can be done to determine the final labels. Also, we advised the annotators not to leave any tweet unmarked, which may lead to a tie during the majority vote for some instances. We create the categories *hate* and *non-hate* based on the existing research on the issue, modifying the defining structures as needed to meet the paradigm of our dataset. We define *hate* and *non-hate* as follows.

1) *Hate:* A *hate* tweet is one that contains a directed insult(s), vulgar language to denigrate a target, or words that instigate or support violence. Furthermore, the simple use of offensive language such as slang and slurs on does not automatically result in a tweet of the type *hate*.
2) *Non-Hate:* All other tweets that do not fall in the *hate* category are *non-hate* tweets.

The annotators skip any code-mixed texts. A few annotation samples are shown in Table I.

[8]Two linguists and one graduate student, sufficiently acquainted with labeling tasks and knowing the concepts of hate speech detection.

D. Inter-Annotator Agreement

Finding agreement among the various annotators is crucial in any annotation effort requiring many raters to create a trustworthy annotated dataset. Cohen's kappa coefficient [36] is one such metric that is often used to evaluate the inter-annotator agreement. It is defined as

$$K = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)}$$

where $\Pr(a)$ and $\Pr(e)$ denote the observed and chance agreement among raters, respectively. Our dataset has an average kappa agreement of 0.74, suggesting that it is good quality. A comparable dataset presented in [37] has a kappa score of 0.62 on average. A majority vote decided the final annotations for the dataset on the annotations of the three annotators.

E. Data Statistics

The dataset consists of 10 242 instances with 5121 instances in each *hate* and *non-hate* classes. We studied the corpus' content and observed that several terms appear often in both classifications, *hate* and *non-hate*, but their distribution frequency varies substantially depending on the polarity of the word and the class type. We also observed that some sets of words occur more frequently in one class than in the other. Table II displays the top 15 commonly occurring terms for each *hate* and *non-hate* classes. The gathered dataset has 230 251 words in total, with 22 320 distinct words. The average sequence length is 22.48.

IV. METHODOLOGY

On top of pre-trained word embeddings, we develop and train three basic and two stacked deep learning-based models, *viz.,* CNN [38], bidirectional long short-term memory (BiLSTM) network [39], bidirectional gated recurrent unit (BiGRU) network [40], stacked CNN_BiGRU, and stacked CNN_BiLSTM. We use the GloVe [41] embedding model to train our word embeddings since it captures syntactic and semantic connections between words. By passing the CNN representation via an attention layer [42], which maps the important and relevant terms from the input text and gives larger weights to these words, the output prediction accuracy is enhanced. Finally, we arrive at our final predictions using an ensemble system comprising the five models mentioned above.

A. Problem Definition

Given a tweet as input, our proposed ensemble framework aims to classify it as *hate* or *non-hate*. Formally, let D be our hate speech dataset containing N samples, then $D = (w_1, y_1), (w_2, y_2), \ldots, (w_n, y_n)$, where each tuple in this set consists of the tweet text (w_i) and the manually annotated *hate* or *non-hate* label (y_i) associated with the tweet. Then

$$D = \{(w_i, y_i)\}_{i=1}^{N} \quad (2)$$

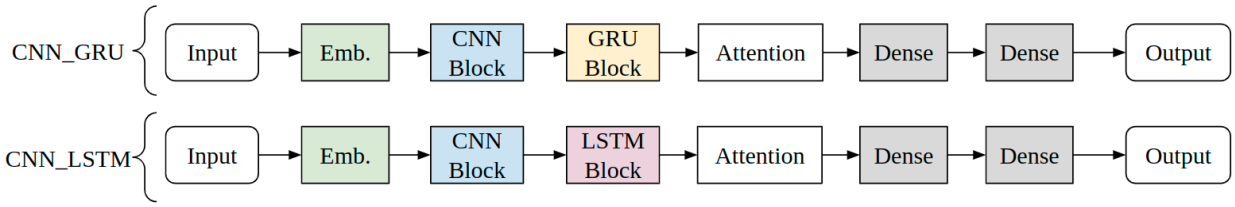

Fig. 1. Overview of the stacked architectures. Emb. represents the GloVe embedding layer.

where w_i represents the ith tweet in the dataset, and the associated label is denoted by y_i. The task's goal is to maximize the value of the following function:

$$\underset{\theta}{\operatorname{argmax}}\left(\Pi_{i=1}^{N} P(y_i|x_i;\theta)\right). \tag{3}$$

The model parameters that we want to optimize are denoted by θ. The classifier is developed by learning on a training set and then tested on the test set of the developed data, modeling the task as a binary classification problem. In particular, the input text is converted into a machine-readable format using a text representation approach that ideally captures and keeps relevant properties in the input text. The representation data are fed into a deep learning algorithm, which assigns the input to two classes with high certainty. This discriminating information is used to build the classifier during the training phase.

B. Word Embeddings

Word embeddings that have been pre-trained on large datasets help capture both the syntactic and semantic information of words. The embedded representation of the words in a text instance is provided to the models. The embeddings are fetched from GloVe (300-dimension) pre-trained word embedding, which was trained on the Common Crawl corpus (840 billion tokens). We pad input arrays with zeros to make input sequences the same length. On our training set, we create an embedding matrix (E) that contains the embedding vectors (x_{ij}) for the words (w_{ij}). The embedding layer passes the word vectors ($x_{1:n}$) to the word sequence encoders (CNN/BiGRU/BiLSTM), which extract the contextual information from the sentence.

C. Convolution Neural Network

CNN [38] is well-known for its ability to produce results equivalent to other state-of-the-art classifiers in image classification problems, but it is also frequently used for text classification tasks. Akhtar *et al.* [43] and Singhal and Bhattacharyya [44] developed CNN-based automated systems for sentiment analysis.

A convolutional layer is a deep learning method that takes a text and assigns importance (learnable weights and biases) to different text parts to differentiate them. Our CNN-based classification method uses two convolutional layers in sequence, each having 150 filters of sizes 2–4. The layers' outputs are combined (merged) to form a single result with the same shape as the individual layers' outputs. After each convolution layer, the convoluted output is subjected to a max-pooling process (pool size = 2). The pooling method produces a feature map that contains the most popular features from the previous feature map, given to the word attention layer. The output of the attention layer is routed via two dense layers (each with 100 neurons and ReLU activation), followed by the final output layer (with softmax activation).

D. GRU Network

Gated recurrent units (GRUs) [40], which are a type of bidirectional recurrent neural network, collect contextual information from both forward $[\overrightarrow{\text{GRU}}(x_{it})]$ and backward $[\overleftarrow{\text{GRU}}(x_{it})]$ time steps in its input and forget gates. The BiGRU layer generates a hidden representation (h_{it}) of each word (w_{it}) in the sentence (s_i): where the sentence number is i, and the t denotes the word number in the sentence i. The forward and backward hidden representations of x_{it} are denoted as $\overrightarrow{h_{it}}$ and $\overleftarrow{h_{it}}$, respectively, which are combined to create h_{it}. We use a 256-neuron BiGRU layer, followed by a 128-neuron GRU layer that accepts input from the prior embedding layer and passes the output through a word attention layer. The output of the attention layer is transmitted via two dense layers (each with 100 neurons and ReLU activation), followed by the final output layer (with softmax activation).

E. LSTM Network

The long short-term memory (LSTM) [39] approach has shown to be one of the most effective methods for learning long-range relationships in text and therefore avoiding the vanishing gradient problem. For controlling the amount of information it wishes to keep in its cell state (memory), the LSTM uses three gates (forget, input, and output). We use a BiLSTM layer with 256 neurons followed by an LSTM layer with 128 neurons, and the output is sent via a word attention layer, similar to the GRU network mentioned in Section IV-D. The output of the attention layer is transmitted via two dense layers (each with 100 neurons and ReLU activation), followed by the final output layer (with softmax activation).

F. Stacked Deep Learning Models

We develop two architectures based on the combinations of CNN, GRU, and LSTM deep learning networks (discussed in Sections IV-C–IV-E). Fig. 1 shows the system architectures of the stacked models. Previous works have observed significant performance improvements with stacked models compared with the individual networks.

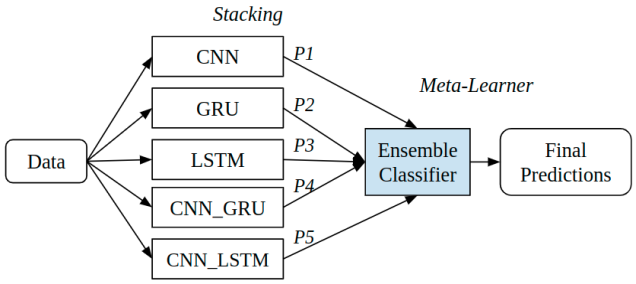

Fig. 2. Overview of the stacked ensemble. $P1$–$P5$ denote the predictions from the constituent systems.

1) *CNN_LSTM:* We develop *CNN_LSTM* by stacking CNN layers on the LSTM block, where the CNN block is responsible for feature extraction on input data, and the LSTM block provides sequence prediction. The CNN model for feature extraction and the LSTM network for feature interpretation over time steps can be considered two sub-models defined by this architecture. The LSTM output is passed through a word attention layer.
2) *CNN_GRU:* Next, we replace the LSTM block with GRU block in the aforesaid stacked model to develop another variant of the stacked model, *CNN_GRU*.

G. Word Attention

On the CNN/GRU/LSTM output, we use an attention mechanism [42] to focus on the most significant words in an input sentence. The attention layer creates a distribution across the input representations to be attended to and outputs a weighted combination of the inputs that is fed into a multilayer perceptron (MLP) layer. Specifically

$$u_{\text{it}} = \tanh(W_w h_{\text{it}} + b_w) \tag{4}$$

$$\alpha_{\text{it}} = \frac{\exp\left(u_{\text{it}}^{\text{T}} u_w\right)}{\sum_t \exp\left(u_{\text{it}}^{\text{T}} u_w\right)} \tag{5}$$

$$s_i = \sum_t (\alpha_{\text{it}} * h_{\text{it}}). \tag{6}$$

The hidden representation of h^{it} is u^{it} (word representation from word encoding layer). The context vector is u^w, and the attention weight for a word is α^{it}. The output phrase vector from the attention layer is s^i.

H. Stacked-Ensemble-Based Hate Speech Classifier

Ensemble learning enables a strategic combination of several models to increase the efficacy of the constituent learning algorithms [45]. By combining the advantages of the various models, the ensemble of models usually improves the prediction accuracy.

We use the stacking ensemble method to load the independent deep learning models, generate predictions for each model, and then average the predictions. The ensemble model must not be trained as it does not take any parameters. However, we create an ensemble model and save it as a single model to reuse and/or deploy it quickly afterward. We use the *layers.average* function from the python-based Keras[9] library to develop our ensemble model. We load the previously discussed five deep learning models into a list and ensure each is assigned a different name. We also ensure that each model's input is a tensor with the same shape as the input data. The stacking ensemble makes use of the same input layer that has been used in the constituent models. Using the *Average()* merge layer in the top layer, the ensemble computes the average of five models' outputs. The ensemble model is expected to have a lower error rate than any single model. The overall architecture of the stacked ensemble system is illustrated in Fig. 2.

I. Calculation of Loss

We use categorical cross-entropy as the loss function and Adam optimizer [46] to train our models through backpropagation. The loss function is realized as

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{7}$$

where N is the total number of instances in the dataset, y_i is the ground-truth label, and $\hat{y}_i$ is the predicted label.

V. Experiments

To evaluate the introduced dataset, a series of baseline experiments were conducted (i.e., whether the two annotated classes can be distinguished only based on the text of the labeled instances). No external resources, such as lexicons, heuristics, or rules, are used in these trials.

A. Baselines

The state-of-the-art techniques that have been tested on our dataset are as follows.

1) *CNNs* [47]*: With 100 feature maps and filter sizes of 3–5, this is a single-layer CNN. The feature maps are maxpooled and presented to a softmax classifier to obtain the output.
2) *Hierarchical Attention Network (HAN) [48]*: The attention mechanism in HAN considers the hierarchical structure of texts and identifies the most relevant words in a sentence and most relevant sentences in a document while taking contextual information into account.
3) *BERT [49]:* We use the recent state-of-the-art BERT_BASE model, which uses 12 transformer [50] encoders, each with 12 self-attention heads and a 768 hidden dimension size. To get the most out of the GPU memory, we set the batch size to 16. To allow for larger batch size, we restrict the document length to 50 tokens.

B. Experimental Setting

The dataset is divided into train, test, and validation sets in the proportions of 70%, 20%, and 10%, respectively. For each *hate* and *non-hate* classes, the resultant train, test, and validation sets have 3592, 1015, and 514 tweets, respectively. The BERT_BASE model is implemented using the python-based *keras_bert*[10] package. During the development of our

[9]A high-level neural network API: https://keras.io/

[10]https://github.com/CyberZHG/keras-bert

TABLE III
DETAILS OF VARIOUS HYPER-PARAMETERS RELATED TO OUR EXPERIMENTS

Parameters	Details
CNN Block	2 successive convolutional layers 150 filters of size 2,3 and 4 Maxpool size: 2
GRU/LSTM Block	1 BiGRU/BiLSTM layer (256 units) 1 GRU/LSTM layer (128 units)
Attention Dimension	100
Output layers	2 neurons
Hidden Activation	ReLU (Dense layers)
Output Activation	Softmax
Batch size	64
Epochs	20
Dropout	0.25 [51]
Loss *Optimizer*	Categorical CrossEntropy Adam [46]

models, we use Keras and Scikit-learn [52] libraries. In each of the developed models, we use a 25% dropout [51] after the attention layer and the dense layers to combat overfitting. The experiments were carried out on an NVIDIA GeForce GTX 1080 Ti GPU. We present the accuracy and macro-$F1$ scores for each model for a complete study when evaluating our models. We report the various hyper-parameters for our implemented systems in Table III.

C. Evaluation Results

Instead of showing the overall precision (P), recall (R), and $F1$ (F) scores of the constituent systems, we report the P, R, and F scores of both the complimentary classes to emphasize the classifiers' unique performance. The assessment results of the five deep learning constituent models and their ensemble are shown in Tables IV and V, respectively. The performance of the constituent models is comparable and very close to one another, as seen in Table IV. The developed GRU and LSTM baseline systems share similar architecture and hyper-parameters settings with only differences in the inherent gating mechanism of the GRU and LSTM units. This may be one of the primary reasons behind the attained comparable scores by the GRU and LSTM systems and their stacked variants. While both GRU and LSTM are popularly known for their capability for processing sequence data, it is established from past studies that notable performance differences between GRU and LSTM are observed for certain dataset characteristics, such as GRU tends to outperform LSTMs on long texts and small datasets [53]. In our study, the dataset is substantially large but not very large, and also, the dataset comprises tweets that are inherently considered as short texts. Hence, this may be another main reason why we do not observe significant differences in the attained scores by GRU and LSTM. The CNN baseline was built on top of the CNN* [47] state-of-the-art system, where we added a convolutional layer and used larger feature maps (150) for varying filter sizes (2–4). Our main objective was to improve the performance of the CNN* system that could produce comparable results, if not better, than the developed GRU and LSTM baseline systems. We achieved this goal through CNN, and as we did not try exhaustive hyperparameter optimization, there is scope for improving the results further. This explains why the CNN and the stacked CNN variants produce similar results with other developed systems. Empirical evaluation shows that despite comparable performances among the developed baseline systems, the proposed stacked ensemble *SEHC* system attains an accuracy of 88.87%, an improvement of 1.62% over the attained average accuracy (87.25%) of the constituent models.

The results also reveal that despite the good per-class performance from all the developed systems, the performance of the *non-hate* class is slightly better than that of the *hate* class. Two constituent systems (CNN and CNN_LSTM) and the ensemble system attained a higher $F1$ score for the *non-hate* class than the *hate* class, whereas only one constituent system (GRU) performed better for the *hate* class than the *non-hate* class. Table VI shows some sample instances which are correctly classified by our proposed ensemble system but misclassified by the baseline methods. The *SEHC* ensemble classifier approximately takes only 14% of the size of the BERT model when saved in the local system, which increases its usability in various situations with resource constraints.

We have conducted the experiments ten times and conducted a Student's t-test with 5% (0.05) significance level to illustrate that the accuracy value obtained by the proposed approach has not happened by chance. We attained a p-value of 0.036, which indicates that the results achieved by *SEHC* are statistically significant over the next best performing system.

D. Comparison With State-of-the-Art

Among the considered baselines, the BERT [49] model attained the best score of 87.88% accuracy. However, it could not outperform the proposed ensemble method. The benchmark CNN* [47] system was the least performing system with 81.87% accuracy which may be attributed to the shallow convolutional network for feature extraction and no attention mechanism, unlike our developed CNN-variant, which used two convolutional layers with multiple filters and sizes along with word attention mechanism. Although the HAN system was able to outperform the CNN* system, it could only attain an accuracy of 84.43%, which is 4.44% less than our proposed *SEHC* system. The low scores of the existing benchmark methods indicate the complexity of identifying hate speech in tweets or social media in general. The superior performance of the stacked ensemble system than the developed constituent baselines is due to its ability to capture the diversity of the models in terms of feature representation for the same input instance.

1) On Unseen Test Data: We tested the effectiveness of our technique on unseen test cases to the standard BERT model using the test set of the TRAC2 dataset [12]. Our *SEHC* model achieved an accuracy of 83.17%, which was 3.25% higher than the result obtained by BERT. However, we found that the BERT model performed marginally better for the *hate* class, whereas the *SEHC* model performed better than BERT for the *non-hate* class. With some trivial post-processing operations, we were able to improve the accuracy

TABLE IV

CLASS-WISE PERFORMANCE OF THE FIVE DEEP LEARNING CONSTITUENT MODELS AND THE PROPOSED *SEHC* MODEL ON THE TEST DATASET. THE VALUES IN BOLD REPRESENT THE HIGHEST OBTAINED F1-SCORE

Models	**CNN**			**GRU**			**LSTM**			**CNN_GRU**			**CNN_LSTM**			***SEHC***		
Classes	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
non-hate	0.85	0.90	0.88	0.89	0.86	0.87	0.86	0.89	0.87	0.87	0.88	0.87	0.86	0.89	0.88	0.87	0.91	**0.89**
hate	0.90	0.84	0.87	0.86	0.89	**0.89**	0.88	0.86	0.87	0.88	0.86	0.87	0.89	0.85	0.87	0.90	0.87	**0.89**

TABLE V

MACRO-MEASURES OF P, R, AND F AND OVERALL ACCURACY OF THE DEVELOPED BASELINES. THE VALUES IN BOLD REPRESENT THE HIGHEST OBTAINED SCORE FOR A GIVEN MEASURE ACROSS ALL MODELS

	Metrics			
Models	**P (%)**	**R (%)**	**F (%)**	**Accuracy (%)**
Developed Baselines				
CNN	87.48	87.34	87.33	87.34
GRU	87.33	87.29	87.29	87.29
LSTM	87.22	87.19	87.19	87.19
CNN_GRU	87.20	87.19	87.19	87.19
CNN_LSTM	87.31	87.24	87.24	87.24
State-of-the-Art Baselines				
CNN* [47]	82.11	81.87	81.84	81.87
HAN [48]	84.56	84.35	84.38	84.43
BERT [49]	88.18	87.88	87.56	87.88
Proposed System				
SEHC	**88.92**	**88.87**	**88.86**	**88.87**
Ablation Experiment				
$SEHC_{Bemb}$	84.77	84.74	84.73	84.74

of the *SEHC* system by 4%. In post-processing, we performed two operations: 1) for example, if the precise forecast was made by both the *SEHC* and BERT algorithms, we consider that label to be the final prediction and 2) for a particular class (say *hate*), in the conflicting predictions, we prioritize the system's predictions that perform better[11] for that particular class (*BERT*) and vice versa. Similar approaches can enable the proposed system to be used on other relevant datasets while also leveraging the efficacy of the existing state-of-the-art methodologies.

E. Ablation Study

To study the role of each model, we modified the suggested stacked ensemble system, *SEHC*, by deleting one of the five constituent models at a time and recording the performance changes for each scenario. When any constituent model is omitted from *SEHC*, the accuracy score falls by 0.39% (on average) throughout the five ablation studies. The CNN_GRU (88.37%) and CNN_LSTM (88.62%) systems had the biggest and lowest accuracy drops, respectively. We do three further ablation experiments to evaluate the overall impact of a certain deep learning approach such as CNN, GRU, or LSTM by deleting the individual variation of a method (say GRU) and its stacked variant (CNN_GRU) from *SEHC*. When we exclude the CNN, CNN_GRU, and CNN_LSTM systems and simply use the GRU and LSTM systems, we get a 2.12% drop in accuracy from the proposed *SEHC* score. This demonstrates the features that CNN variations contribute to the ensemble approach. The drop in accuracy is minimal (0.8%) when LSTM and CNN_LSTM from *SEHC* are removed, but large (1.3%) when GRU variants are removed.

To investigate whether BERT embeddings (Bemb) can improve the proposed model, we have implemented the five constituent models in our work, replacing GloVe with Bemb. We use only the embeddings, and no fine-tuning of the BERT layers is done because that would result in the constituent systems being more dependent on the features generated from BERT and less on the following deep learning methods of CNN/GRU/LSTM. We observe empirically that using only Bemb, the average accuracy attained by the five models is 82.96%, and the ensemble model ($SEHC_{\text{Bemb}}$) attains an accuracy of 84.74%, which is an improvement of 1.78%. While it is clear that the ensembling of multiple models is helping to achieve significant improvement in performance than the constituent models, usage of Bemb has produced significantly lower scores when compared with the GloVe-based systems. This is due to the different input segmentation used by these two approaches. GloVe uses typical word-like tokens, but BERT divides its input into subword units called word-pieces. BERT assures that there are no out-of-vocabulary tokens, and completely unfamiliar words are broken into characters so that it is unlikely to understand them. BERT learns its unique word-piece embeddings alongside the overall model (which we did not allow in this scenario). Word fragments in BERT do not contain the same semantic information as GloVe, and BERT must make meaning of them in the later layers.

F. Error Analysis

A manual assessment of a set of incorrectly categorized sentences was undertaken to understand the performance of the trained classifiers better. There are two primary categories of mistakes that have been identified.

1) *False Negatives:* Sentences those were manually marked as *hate* but were categorized as *non-hate* by the algorithm, generally owing to a lack of context or the required world knowledge to comprehend the meaning of the statement.
   a) Heard the climate change crowd wants to blame climate change for Hurricane Michael! A hurricane Camille was stronger in 1969, and b I hope we don't drag petite women out of SUVs and beat them since we can't get to the big offenders India and China.

[11]A manual analysis of few sample predictions can help us learn the performance of a model on the respective classes on unseen data.

TABLE VI
SAMPLE INSTANCES THAT WERE CORRECTLY CLASSIFIED BY THE ENSEMBLE SYSTEM. WORDS IN ITALICS SHOWS THE TRUE LABELS

Sentence	CNN*	HAN	BERT	*SEHC*
Mahesh, what can we call these people? Dogs or Humans?	*non-hate*	*non-hate*	*non-hate*	*hate*
That is what the English said about India, Malta, Ireland and over 100 nations, 'you are the disease and we are working on the cure.'	*non-hate*	*non-hate*	*non-hate*	*hate*
This is no different than what some rogue police officers do. Domestic terrorism, only without a badge this time.	*non-hate*	*non-hate*	*non-hate*	*hate*
Indeed he has policy of zero-tolerance towards terrorism.	*hate*	*hate*	*hate*	*non-hate*
Kashmiri protests is a matter of Terrorism.	*hate*	*hate*	*hate*	*non-hate*

b) Why should I believe this data? This data is based on your country census. You guys go to the UN forum and tell blatant lies about terrorism, etc. If Hindus r safe and population increasing in Pak why are some of them coming to India and taking refugee in India?

From the above examples, we find that long sequences seem a significant limitation for our proposed model to predict correctly. Tweets with too much information or additional context seem to act as noise and negatively impact the system's performance in such cases.

2) *False Positives:* Sentences those were manually labeled as *non-hate* but were automatically categorized as *hate* owing to the usage of common inflammatory language with non-hateful intent.
   a) We have had 27 300 homicide offenses and 120 300 sex offenses over 397 000 assault offences and 1100 terrorism offenses.
   b) 15 years for a hate crime.

   Our model finds difficulty in some cases where common words related to hate (homicide, crimes, terrorism) are explicitly mentioned in the tweet even if the overall tweet does not signify *hate*. Insufficient context information (as in the second sentence) causes another type of limitation for our model to predict instances correctly.

G. Discussions

Several elements of the introduced dataset and hate speech annotation, in general, are discussed in this section.

1) Dataset Curation: This study outlines the key factors for determining what constitutes hate speech and what does not. In any event, despite our best attempts to make the annotation standards as clear, rational, and thorough as possible, the annotation process has proven to be difficult and time-consuming. Even though we have leveraged offensive words and phrases to develop this dataset, not all instances with traces of profanity or a hateful undertone are construed as hate speech.

2) Annotation: The annotation rules were established in stages, starting with a review of the hate speech annotation literature. After the initial round of manual labeling, the criteria were checked for inconsistencies among human annotators, and the rules were updated. The annotation criterion of *hate* speech being a planned assault is unclear and subject to numerous interpretations since it lacks robustness and a firm definition. The other annotation criterion, *hate* speech directed toward a particular group of people, was also a source of contention among human annotators.

3) Choosing a Suitable Sequence Length: We noted a trade-off while performing error analysis on the predictions of our proposed method. In many cases, lengthy tweets were misclassified by the classifier due to excess information or additional context. Again, all the developed systems misclassified the short tweets in a few cases. Considering a suitable sequence length of the input while training the models seems to be a critical factor in achieving good performance by the system.

4) Role of External Knowledge: Human annotators usually use external information or common sense to fill in the necessary context while annotating a sentence. Because automatic classifiers lack access to this additional knowledge, they are prone to making errors in such instances.

VI. LIMITATIONS

The indicators of hate speech fluctuate depending on the issue, time, cultural demography, target demographic, etc., which makes it vastly dynamic and challenging to recognize. As a corpus created by mining a social media platform, the created corpus has a few limitations. While this corpus is of present importance, it may lose its relevance or be supplanted by future trends in public discussions. Another issue with online content, in general, is that supplementary material, such as URL links contained in postings, may be deleted, rendering the context for the content contributions inadequate or just inaccessible. Unfortunately, this is due to the fragile nature of online information and data, a prevalent issue in this study.

VII. CONCLUSION AND FUTURE WORK

This study attempts to address the lack of variation in the existing hate speech detection corpora content. We discuss a hate speech dataset manually tagged and acquired from Twitter, encompassing diverse topics such as terrorism, cyberbullying, natural disasters, and politics. The dataset includes 10 242 tweets classed as hate speech or not. Because the idea of hate speech is so nuanced, the manual annotation criteria and standards are established in this work. Several dataset features have been investigated, such as the annotators' need for extra information to decide or the distribution of vocabulary used in hate speech samples. Furthermore, automated classifiers have been used in a variety of baseline investigations, emphasizing an ensemble deep learning system that exceeds the individual performances of the automatic classifiers. The ability of the stacked ensemble system to capture the variety of existing deep neural models in terms of feature representation for the same

input instance is why it outperforms the considered state-of-the-art systems. In addition, the proposed ensemble approach is less resource-hungry than popular transformer-based models and yet produces superior results when compared with the same. What is evident is that hate speech is a complex subject, and to create better resources and effective hate speech classifiers, we need to do a lot more fine-tuning, debate, and consensus on what hate speech even is.

In the future, it would be intriguing to look at methods to include global knowledge and/or the context of an online conversation (e.g., past and subsequent postings, forum subject title) into a future study to build more robust hate speech automated classifiers. To identify online hate speech, features such as social graph mining, network analysis, and user centrality assessment can be used. Furthermore, rephrasing the goal as developing a regression model to predict the precise number of likes, retweets, and reactions would be preferable and more informative, potentially leading to a better understanding of how hate speech operates on Twitter.

ACKNOWLEDGMENT

The authors would like to thank the linguists (Saroj Jha, Akash Bhagat, and Swati Srivastava) for labeling the tweets.

REFERENCES

[1] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proc. ICML*, 2011, pp. 513–520.

[2] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.

[3] A. Olteanu, C. Castillo, J. Boy, and K. Varshney, "The effect of extremist violence on hateful speech online," in *Proc. Int. AAAI Conf. Web Social Media*, vol. 12, no. 1, 2018, pp. 221–230.

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

[5] J. Qian, M. ElSherief, E. M. Belding, and W. Y. Wang, "Hierarchical CVAE for fine-grained hate speech classification," in *Proc. EMNLP*, 2018, pp. 3550–3559.

[6] S. Poria, D. Hazarika, N. Majumder, G. Naik, E. Cambria, and R. Mihalcea, "MELD: A multimodal multi-party dataset for emotion recognition in conversations," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 527–536.

[7] Y. Zhang, Z. Zhao, P. Wang, X. Li, L. Rong, and D. Song, "ScenarioSA: A dyadic conversational database for interactive sentiment analysis," *IEEE Access*, vol. 8, pp. 90652–90664, 2020.

[8] C. Sharma *et al.*, "SemEval-2020 task 8: Memotion analysis-the visuo-lingual metaphor!" in *Proc. 14th Workshop Semantic Eval.*, 2020, pp. 759–773.

[9] Y. Zhang *et al.*, "A quantum-inspired multimodal sentiment analysis framework," *Theor. Comput. Sci.*, vol. 752, pp. 21–40, Dec. 2018.

[10] 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.

[11] S. Sharma, S. Agrawal, and M. Shrivastava, "Degree based classification of harmful speech using Twitter data," in *Proc. 1st Workshop Trolling, Aggression Cyberbullying (TRAC)*, 2018, pp. 106–112.

[12] R. Kumar, A. K. Ojha, S. Malmasi, and M. Zampieri, "Evaluating aggression identification in social media," in *Proc. 2nd Workshop Trolling, Aggression Cyberbullying*, Marseille, France, 2020, pp. 1–5.

[13] S. Masud *et al.*, "Hate is the new infodemic: A topic-aware modeling of hate speech diffusion on Twitter," in *Proc. IEEE 37th Int. Conf. Data Eng. (ICDE)*, Apr. 2021, pp. 504–515.

[14] J. Golbeck *et al.*, "A large labeled corpus for online harassment research," in *Proc. ACM Web Sci. Conf.*, 2017, pp. 229–233.

[15] A. M. Founta *et al.*, "Large scale crowdsourcing and characterization of Twitter abusive behavior," in *Proc. 12th Int. AAAI Conf. Web Social Media*, 2018, pp. 491–500.

[16] D. Chatzakou, N. Kourtellis, J. Blackburn, E. De Cristofaro, G. Stringhini, and A. Vakali, "Mean birds: Detecting aggression and bullying on Twitter," in *Proc. ACM web Sci. Conf.*, 2017, pp. 13–22.

[17] J. Moon, W. I. Cho, and J. Lee, "BEEP! Korean corpus of online news comments for toxic speech detection," in *Proc. 8th Int. Workshop Natural Lang. Process. Social Media*. Stroudsburg, PA, USA: Association for Computational Linguistics, Jul. 2020, pp. 25–31. [Online]. Available: https://aclanthology.org/2020.socialnlp-1.4, doi: 10.18653/v1/2020.socialnlp-1.4.

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

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

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

[21] B. Gambäck and U. K. Sikdar, "Using convolutional neural networks to classify hate-speech," in *Proc. 1st Workshop Abusive Lang. Online*, 2017, pp. 85–90.

[22] R. Basak, S. Sural, N. Ganguly, and S. K. Ghosh, "Online public shaming on Twitter: Detection, analysis, and mitigation," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 2, pp. 208–220, Apr. 2019.

[23] K. Rudra, A. Sharma, N. Ganguly, and S. Ghosh, "Characterizing and countering communal microblogs during disaster events," *IEEE Trans. Computat. Social Syst.*, vol. 5, no. 2, pp. 403–417, Jan. 2018.

[24] 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*, vol. 11, no. 1, 2017, pp. 512–515.

[25] S. Malmasi and M. Zampieri, "Challenges in discriminating profanity from hate speech," *J. Experim. Theor. Artif. Intell.*, vol. 30, no. 2, pp. 187–202, Mar. 2018.

[26] H. Watanabe, M. Bouazizi, and T. Ohtsuki, "Hate speech on Twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection," *IEEE Access*, vol. 6, pp. 13825–13835, 2018.

[27] F. E. Ayo, O. Folorunso, F. T. Ibharalu, and I. A. Osinuga, "Hate speech detection in Twitter using hybrid embeddings and improved cuckoo search-based neural networks," *Int. J. Intell. Comput. Cybern.*, vol. 13, no. 4, pp. 485–525, Nov. 2020.

[28] F. E. Ayo, O. Folorunso, F. T. Ibharalu, I. A. Osinuga, and A. Abayomi-Alli, "A probabilistic clustering model for hate speech classification in Twitter," *Expert Syst. Appl.*, vol. 173, Jul. 2021, Art. no. 114762.

[29] P. Kapil and A. Ekbal, "Leveraging multi-domain, heterogeneous data using deep multitask learning for hate speech detection," 2021, *arXiv:2103.12412*.

[30] X. Huang, L. Xing, F. Dernoncourt, and M. Paul, "Multilingual Twitter corpus and baselines for evaluating demographic bias in hate speech recognition," in *Proc. 12th Lang. Resour. Eval. Conf.*, 2020, pp. 1440–1448.

[31] J. Qian, A. Bethke, Y. Liu, E. Belding, and W. Y. Wang, "A benchmark dataset for learning to intervene in online hate speech," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 4755–4764.

[32] B. Mathew, P. Saha, S. M. Yimam, C. Biemann, P. Goyal, and A. Mukherjee, "Hatexplain: A benchmark dataset for explainable hate speech detection," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 17, 2021, pp. 14867–14875.

[33] N. Prasad, S. Saha, and P. Bhattacharyya, "A multimodal classification of noisy hate speech using character level embedding and attention," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–8.

[34] S. A. Castano-Pulgarín, N. Suárez-Betancur, L. M. T. Vega, and M. H. andH López, "Internet, social media and online hate speech. systematic review," *Aggression Violent Behav.*, vol. 58, May 2021, Art. no. 101608.

[35] G. Kennedy, A. McCollough, E. Dixon, A. Bastidas, J. Ryan, C. Loo, and S. Sahay, "Technology solutions to combat online harassment," in *Proc. 1st Workshop Abusive Lang.*, 2017, pp. 73–77.

[36] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, 1960.

[37] O. de Gibert, N. Perez, A. García-Pablos, and M. Cuadros, "Hate speech dataset from a white supremacy forum," in *Proc. 2nd Workshop Abusive Lang. Online (ALW)*, 2018, pp. 11–20.

[38] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*.

[39] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[40] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014, *arXiv:1409.1259*.
[41] 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.
[42] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*.
[43] M. S. Akhtar, A. Kumar, A. Ekbal, and P. Bhattacharyya, "A hybrid deep learning architecture for sentiment analysis," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers*, 2016, pp. 482–493.
[44] P. Singhal and P. Bhattacharyya, "Borrow a little from your rich cousin: Using embeddings and polarities of English words for multilingual sentiment classification," in *Proc. COLING 26th Int. Conf. Comput. Linguistics, Tech. Papers*, 2016, pp. 3053–3062.
[45] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL, USA: CRC Press, 2019.
[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[47] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Doha, Qatar, Aug. 2014, pp. 1746–1751.
[48] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 1480–1489.
[49] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, 2019, pp. 4171–4186.
[50] A. Vaswani *et al.*, "Attention is all you need," in *Proc. NIPS*, 2017, pp. 5998–6008.
[51] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.
[52] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
[53] S. Yang, X. Yu, and Y. Zhou, "LSTM and GRU neural network performance comparison study: Taking yelp review dataset as an example," in *Proc. Int. Workshop Electron. Commun. Artif. Intell. (IWECAI)*, Jun. 2020, pp. 98–101.

Soumitra Ghosh received the B.Tech. degree from Notabilis, Sociatrix, Humanus and Maxime (NSHM), Durgapur, India, in 2014, and the M.Tech. degree from the Birla Institute of Technology (BIT), Mesra, India, in 2016.

He is currently a Senior Research Fellow with the Department of Computer Science and Engineering, IIT Patna, Patna, India. He has published papers in various peer-reviewed conferences and journals of international repute. His main areas of research are natural language processing (NLP) and sentiment analysis.

Asif Ekbal is currently an Associate Professor with the Department of Computer Science and Engineering, IIT Patna, Patna, India. He has been pursuing research in natural language processing, information extraction, text mining, and machine learning applications for the past 14 years. He is involved with different sponsored research projects in broad areas of artificial intelligence (AI) and machine learning technologies, funded by government and private agencies.

Dr. Ekbal was a recipient of the Visvesvaraya Young Faculty Research Fellowship (YFRF), Government of India; the Japan Society for the Promotion of Science (JSPS) Fellowship Award from the Goverment of Japan; and the Innovative Project Award from the Indian National Academy of Engineering (INAE).

Pushpak Bhattacharyya is currently the Director of IIT Patna, Patna, India, and a Professor of computer science and engineering with IIT Patna and IIT Bombay, Mumbai, India. He is an Outstanding Researcher in natural language processing and machine learning. He has contributed to all areas of natural language processing, encompassing machine translation, sentiment and opinion mining, cross-lingual search, and multilingual information extraction.

Prof. Bhattacharyya held the office of the Association for Computational Linguistics (ACL), the highest body of natural language processing (NLP), as the President. He received several awards and fellowships, such as the Fellow of Indian National Academy of Engineering, the International Business Machine (IBM) Faculty Award, and the Yahoo Faculty Award.

Tista Saha has been working as a Research Engineer with the Centre for Development of Telematics (C-DOT), New Delhi, India, since 2016. It is a research organization in the telecom sector. She has worked in open source intelligence systems, majorly in gender, sentiment analysis, and hate speech detection (social media data source) to identify path-breaking standards in artificial intelligence (AI). Her skills include algorithm development, data analysis, and machine learning.

Alka Kumar received the bachelor's and master's degrees in computer science from Delhi University, New Delhi, India, in 1991 and 1994, respectively.

She is currently the Team Leader with the Centre for Development of Telematics (C-DOT), New Delhi, a premier Government of India research and development organization. She has been associated with this organization for more than 25 years with rich experience in telecom projects, including developing and deploying intelligent network system service control and switching points, Third Generation Partnership Project (3GPP) network customized applications for mobile network enhanced logic (CAMEL) solution and services, and gigabit passive optical network, which is a backbone of Bharatnet Project. She is currently working in text-based analytic solutions with a keen interest in building high-score deep learning artificial intelligence (AI) models that can be integrated successfully with industrial requirements.

Shikha Srivastava received the B.Tech. degree in electronics engineering from IIT-BHU, Varanasi, India, in 1991, and the M.Res. degree from IIT Delhi, New Delhi, India, in 2007.

She has recently been appointed as the Director of Member Board with the Centre for Development of Telematics (C-DOT), New Delhi. She has more than 25 years of hands-on and supervisory experience in communication systems analysis, design, development, and support. Her work areas include digital switching, fault-tolerant systems, signaling system 7 (SS7), fixed line short message service (SMS), global system for mobile communication (GSM) core network elements, universal mobile telecommunications system (UMTS), voice over internet protocol (VoIP), chat protocol, lawful interception, internal navigation, text analytics, video analytics, mobile application development, machine learning, and artificial intelligence.

Ms. Srivastava received the Gold Medal.