IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 June 2023, accepted 17 July 2023, date of publication 25 July 2023, date of current version 3 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3298808

RESEARCH ARTICLE

A Deep Learning Framework for the Detection of Malay Hate Speech

KRISHANU MAITY[1], SHAUBHIK BHATTACHARYA[1], SRIPARNA SAHA[1], (Senior Member, IEEE), AND MANJEEVAN SEERA[2]

[1]CSE Department, Indian Institute of Technology Patna, Bihta 801106, India
[2]Department of Econometrics and Business Statistics, School of Business, Monash University Malaysia, Subang Jaya, Selangor Darul Ehsan 47500, Malaysia

Corresponding author: Krishanu Maity (krishanu_2021cs19@iitp.ac.in)

This work was supported by the Ministry of External Affairs (MEA) and the Department of Science and Technology (DST), India, under the ASEAN-India Collaborative Research and Development Scheme.

ABSTRACT Although social media can efficiently disseminate information, they also facilitate the dissemination of online abuse, harassment, and hate speech. In 2019, United Nations Secretary-General introduced the United Nations Strategy and Plan of Action on Hate Speech in response to the alarming global trend of rising hate speech. It is crucial to prevent hate speech because it can have severe negative effects on both individuals and society. While much research has been conducted on detecting online hate speech in English, little research has been conducted in other languages, such as Malay. In this paper, we present the first benchmark dataset *HateM* for detecting hate speech in Malay, comprised of over 4,892 annotated tweets. We created a two-channel deep learning model, *XLCaps*, to effectively manage noisy Malay language posts. One channel's input is the XLNet language model followed by the capsule network, while the other channel's input is the FastText embedding with Bi-GRU. Our proposed model surpasses the baseline models in terms of overall accuracy and F1 measurement, which are 80.69% and 80.41%, respectively. This work contributes to the prevention of hate speech in Malay and can serve as a basis for future study in this area. The approach to effectively managing noisy Malay posts can be also applied to other languages. The code and dataset are available at https://github.com/MaityKrishanu/Hate_Malay.

INDEX TERMS Hate speech, Malay, transformer, capsule network, FastText.

I. INTRODUCTION

Social media platforms have become an integral part of people's lives, allowing them to connect, express themselves, and exchange ideas with people from all over the world. Despite their many positive effects, these platforms are frequently beset by the prevalence of hate speech and offensive language. Hate speech not only violates the right to free speech and expression, but also has a deleterious effect on the mental health and well-being of individuals. Hate speech [1] is any communication intended to attack the dignity of a group on the basis of characteristics such as race, gender, ethnicity, sexual orientation, nationality, religion, or other characteristics. According to the Pew Research Center, 40% of social media users have experienced online harassment or intimidation [2]. According to the FBI, there were 8,263 reported incidents of hate crimes in 2020, a 13 percent increase from the 7,314 incidents reported in 2019.[1] Facebook detected and acted upon 22,3 million instances of hate speech content between July and September of 2021.[2] From December 2019 to March 2020, there was a 900% increase in the number of tweets containing hate speech directed at Chinese individuals and China, according to a study.[3] These presumably harmless social media posts incite violence and riots in the real world [2].

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

[1] https://www.fbi.gov/news/press-releases/fbi-releases-2019-hate-crime-statistics

[2] https://transparency.fb.com/data/community-standards-enforcement/hate-speech/facebook/

[3] https://https://l1ght.com/Toxicity_during_coronavirus_Report-L1ght.pdf

This justifies the requirement to detect and restrict hate speech. Significant research has been conducted over the past decade to develop models and datasets for the automatic detection of hate speech in the English language using traditional machine learning [3], [4], [5] and deep learning techniques [6], [7], [8]. For other languages, such as Italian [9], Indonesian [10], and Thai [11], there are fewer studies available. The detection of hatred speech in languages with limited resources presents a unique set of obstacles. The limited availability of annotated datasets, which are essential for training and evaluating machine learning models, is one of the primary obstacles. In addition, hate speech is strongly influenced by cultural norms, beliefs, and contextual factors, which can vary considerably across languages and regions [7]. This cultural sensitivity complicates the direct application of existing hate speech detection models from high-resource languages to low-resource languages such as Malay, as they may not be aligned with the cultural nuances of the target language. When constructing hate speech detection models for this low-resource language, it is crucial to account for the unique cultural characteristics of the Malay language and its context.

According to a recent report by the Centre for Independent Journalism Malaysia (CIJ), instances of online hate speech targeting marginalised groups, such as Rohingya refugees and the bisexual and transgender community, have increased significantly. According to The Malaysian Reserve (2020),[4] between January and June 2020, the Malaysian Communications and Multimedia Commission (MCMC) received a total of 11,235 complaints regarding various cyber offences such as harassment, cyberbullying, false news, and hate speech. These statistics demonstrate the urgent need to address the problem of hate speech in the Malay language. The lack of adequate measures for detecting and mitigating hate speech in Malay poses a significant challenge and necessitates further research and development of hate speech detection techniques in this low-resource language in order to effectively combat the expanding online threat of hate speech in Malaysia.

Utilizing advanced machine learning tools for hate speech detection and employing human content moderators to identify, demote, and remove problematic content are the two most prevalent strategies for addressing hate speech in contexts with limited resources. Both of these approaches, however, have significant limitations. Machine learning systems require annotated ground truth data and comprehensive data processing capabilities, which are frequently absent in languages with limited resources, such as Malay. According to our knowledge, there is no publicly accessible dataset of Malay hate speech. In addition, the vast volume of content generated daily on social media makes it impractical to rely solely on human moderators to detect all instances of hate speech, save for the most prominent ones. Even if feasible for a single context, scaling to additional contexts, languages, and countries is not straightforward. Moreover, the lexicon used in social media frequently deviates from standard literary language, posing challenges for natural language processing techniques [12]. For instance, Malay social media data is frequently noisy due to the intentional use of misspellings, abbreviations, and acronyms to obscure terms and elude automated detection. Such words may not be included in the pre-trained word embedding models, resulting in the loss of morphological information.

This spurred the development of a unique dataset and a more reliable model for online hate speech identification in the Malay language, with the goal of having the hate speech detection systems identify hate communications automatically. Our objective is to facilitate the automatic identification and flagging of hate messages, thereby assisting law enforcement agencies in taking appropriate action against those who engage in hate speech. This work is vital with three-fold contributions, as follows:

(i) Creation of *HateM*, consisting of a dataset for hate speech detection in the Malay language with over 4,892 manually annotated tweets with hate and non-hate classifications.

(ii) Development of *XLCaps*, a novel two-channel framework for efficient Malay data representation. The first channel utilises XLNet [13], a generalised autoregressive (AR) pretraining method that combines the benefits of AR and autoencoding (AE) methods by means of permutation language modelling. Using an iterative dynamic routing strategy, capsule networks [14] are then employed to encapsulate hierarchical relationships between successive layers. FastText embedding with Bi-GRU is used in the second channel, which takes advantage of character-level representations for word vectors, as opposed to word2vec and Glove, which use word-level representations.

(iii) Design of experiments on *XLCaps*, by comparing with generic machine learning and deep learning approaches. This is compared across a variety of evaluation metrics.

II. RELATED WORK

Text mining and NLP paradigms have previously been used to examine a variety of topics related to hate speech detection, such as identifying online sexual predators, detecting internet abuse, and detecting cyberterrorism [15].

Detecting hateful and offensive speech presents challenges in understanding contextual nuances, addressing data bias, handling multilingual and code-switching text, adapting to the evolving nature of hate speech, dealing with subjectivity and ambiguity, countering evasion techniques, and considering ethical considerations [16]. These challenges necessitate robust and adaptable methodologies, including deep learning and user-centric approaches, to enhance hate speech detection systems. A common approach for hate speech detection involves combining feature extraction with classical machine learning algorithms. For instance, Dinakar et al. [4] utilized the Bag-of-Words (BoW) approach in conjunction with a Naïve Bayes and Support Vector Machines (SVMs) classifier. Deep Learning, which has demonstrated success in computer

[4] https://themalaysianreserve.com/2020/08/12/mcmc-addresses-over-11000-complaints-within-first-six-months-this-year/

vision, pattern recognition, and speech processing, has also gained significant momentum in natural language processing (NLP). One significant advancement in this direction was the introduction of embeddings [17], which have proven to be useful when combined with classical machine learning algorithms for hate speech detection [18], surpassing the performance of the BoW approach. Furthermore, other Deep Learning methods have been explored, such as the utilization of Convolutional Neural Networks (CNNs) [19], Recurrent Neural Networks (RNNs) [20], and hybrid models combining the two [21]. Another significant development was the introduction of transformers, particularly BERT, which exhibited exceptional performance in a recent hate speech detection competition, with seven out of the top ten performing models in a subtask being based on BERT [22].

The connected study mentioned below illustrates that while most current work is undertaken in English, hate speech identification in several low-resource languages should receive more attention.

A. WORKS ON ENGLISH DATA

The work by Watanabe et al. [23] introduced an approach that utilized unigrams and patterns extracted from the training set to detect hate expressions on Twitter, achieving an accuracy of 87.4% in differentiating between hate and non-hate tweets. Similarly, Davidson et al. [24] collected tweets based on specific keywords and crowdsourced the labeling of hate, offensive, and non-hate tweets, developing a multi-class classifier for hate and offensive tweet detection. In a separate study, a dataset of 4500 YouTube comments was used by authors in [4] to investigate cyberbullying detection, with SVM and Naive Bayes classifiers achieving overall accuracies of 66.70% and 63% respectively. A Cyberbullying dataset was created from Formspring.me in a study by authors in [5], and a C4.5 decision tree algorithm with the Weka toolkit achieved an accuracy of 78.5%. CyberBERT, a BERT-based framework created by [25], exhibited cutting-edge performance on Twitter (16k posts), Wikipedia (100k posts) and Formspring (12k posts) datasets. On a hate speech dataset of 16K annotated tweets, Badjatiya et al. [8] conducted extensive tests with deep learning architectures for learning semantic word embeddings, demonstrating that deep learning techniques beat char/word n-gram algorithms by 18% in terms of F1 score.

B. WORKS ON LOW RESOURCE LANGUAGES

The term "low-resource" refers to situations in which there are few technical resources available, such as labelled training data, linguistic tools for tasks such as semantic analysis, named-entity recognition, and parts of speech tagging, or digitised texts that can be used as supervised/unsupervised training data for language models. Prior research conducted in low-resource contexts includes studies by Mubarak et al. [26], who examined abusive language in Arabic tweets based on machine learning techniques. Similar investigations have been carried out in other languages like Amharic [27], Indonesian [28], and Vietnamese [29].

Pasupa et al. [11] constructed a benchmark Thai hate speech dataset from Facebook, Twitter, and YouTube postings, according to the authors. They got cutting-edge performance by fine-tuning the WangchanBERTa with the ordinal regression loss function. The authors of [21] achieved 79.28% accuracy in cyberbullying detection using CNN, BERT, GRU, and Capsule Networks on their provided code-mixed Indian language dataset. Based on the F1 score, the deep learning-based domain-specific word embedding model inciterwcm5 beats the standard model for recognizing hate speech from Hindi-English code mixed data by 13%. Authors in [30] compiled an aggression-annotated corpus of 21k Facebook comments and 18k tweets written in a Hindi-English code-mixed language.Karim et al. [31] developed an explainable hate speech detection approach (DeepHateExplainer) in Bengali based on different variants of transformer architectures (BERT-base, mMERT, XLM-RoBERTa).

Authors in [32] examine the effectiveness of textual features, acoustic features, and a combination of both in detecting hate speech in the Indonesian language using deep learning techniques. The experimental results reveal that the model leveraging textual features achieves the highest accuracy, achieving an Fl-score of 87.98%. Alfina et al. [28] address the scarcity of studies on hate speech detection in the Indonesian language. They created a comprehensive dataset that encompasses hate speech targeting various domains, including religion, race, ethnicity, and gender. By employing machine learning techniques, specifically word n-gram features with the Random Forest Decision Tree algorithm, they achieved an impressive F-measure of 93.5%. Moy et al. [33] explored various forms of toxicity across multiple languages, including hate speech, cyberbullying, obscenity, threats, and insults. Their study utilized the toxic comment classification dataset which contains Wikipedia comments highlighting the multilingual aspect of their research. In contrast, our study builds on the premise of developing and utilizing a dataset that has been directly collected from the Malay-speaking community on Twitter. This method has allowed us to capture the unique linguistic nuances and idiosyncrasies of hate speech as it naturally occurs in the Malay language. Additionally, our study specifically distinguishes between hate speech and non-hate instances through manual annotation.

Based on an extensive literature survey, our findings indicate that the majority of existing research on hate speech detection is focused on English language data. However, there is a growing recognition of the need to address hate speech in low-resource languages to effectively combat its negative impact on society.

III. *HateM:* HATE SPEECH CORPORA DEVELOPMENT

In this section, we have introduced a new benchmark hate speech dataset in Malay.

A. DATA COLLECTION

We leveraged the Twitter streaming API[5] and Twitter Search API[6] to collect tweets from Twitter, including both historical tweets based on specific keywords and real-time streaming data. Over a period of December 2022 to January 2023, we obtained approximately 20,000 raw tweets using a set of keywords such as (“bodoh” - stupid; “sial” - damn; “gila” - insane; “babi” - pig; “haram” - forbidden; “anjing” - dog; “mati” - dead; “setan” - devil; “celaka” - unfortunate; “bangsat” - bastard; “jahat” - evil; “hitam” - black; “pendek” - short; “lembab” - slow) as mentioned in [34]. The selection of the period from December 2022 to January 2023 for data collection was primarily driven by practical considerations and the availability of resources. However, we acknowledge that during this period, several significant events took place in Malaysia that could potentially have an impact on the hate speech landscape. In December 2022, a few noteworthy events occurred, including the appointment of Ahmad Zahid Hamidi as Deputy Prime Minister, despite his controversial past and corruption charges. This decision received criticism from the Coalition for Clean and Fair Elections (Bersih). In addition, a devastating landslide in Batang Kali, Selangor resulted in the death of at least 31 people. January 2023 was marked by political upheaval, with the Umno party undergoing a purge due to an inability to accept differing opinions or criticism.

After pre-processing the scraped tweets, we narrowed down the dataset to 4,892 tweets written in Malay, which were then subjected to manual annotation.

B. DATA PRE-PROCESSING

Because each language has a significant level of ambiguity, data pretreatment is crucial for any NLP work. The raw data is made up of countless irrelevant tweets. To make the annotation work easier, we created a filter that removes unnecessary tweets that meets the following requirements:

- If a tweet is repeated.
- If the tweet just has a URL.
- If the tweet is written in other than Malay language.
- If the tweet is less than 10 characters.

C. DATA ANNOTATION

1) HATE SPEECH DEFINITION

In order to ensure consistency and reliability in data annotation, we faced the challenge of defining hate speech, as there is no universally agreed-upon definition, and contextual variations exist. To address this, we sought to adopt a definition that was broad enough to encompass relevant instances of hate speech in Malaysia, yet specific enough to avoid ambiguity for annotators. We reviewed definitions provided by the United Nations Strategy and Plan of Action on Hate Speech [35], social media platforms, and existing hate speech research [24].

For instance, Facebook defines hate speech as “a direct attack against people rather than concepts or institutions on the basis of protected characteristics such as race, ethnicity, national origin, disability, religious affiliation, caste, sexual orientation, sex, gender identity, and serious disease.” Similarly, Twitter defines it as “promoting violence against, directly attacking, or threatening others based on race, ethnicity, national origin, caste, sexual orientation, gender, gender identity, religious affiliation, age, disability, or serious disease.” Since there was no national definition for hate speech in Malaysia, we developed an annotation guideline by adapting the UN’s definition and incorporating protected characteristics mentioned by Twitter.

2) ANNOTATION TRAINING

The annotation process was led by two proficient professors with extensive expertise in hate speech and offensive content, and executed by three undergraduate students who were proficient in Malay. Initially, a group of master’s students in linguistics were voluntarily recruited through the department email list and compensated with gift vouchers and honorarium. For annotation training, gold standard samples annotated with hate speech labels (Hate or Non-hate) were required. Our expert annotators randomly selected 300 samples (tweets) and annotated them with either a hate or non-hate label. During the annotation training, emphasis was placed on the definition of hate speech, which was clarified to include dehumanizing or demeaning sentiment expressed towards a target based on protected characteristics. It was also highlighted that hate speech can be directed towards individuals or groups [36], and can be explicit or implicit, with the latter requiring contextual understanding. Subsequently, expert annotators engaged in discussions to resolve differences and create 300 gold-standard samples with hate speech annotation. These annotated examples were divided into three sets, each containing 100 samples, to facilitate three-phase training.

After the completion of each phase, expert annotators collaborated with novice annotators to rectify any mis-annotations, and the annotation guidelines were updated accordingly. Following the conclusion of the third round of training, the top three annotators were selected to annotate the entire dataset.

3) MAIN ANNOTATION

For the main annotation process, we utilized the open-source platform Docanno,[7] which was deployed on a secure Heroku instance. Each qualified annotator was provided with a dedicated account to annotate and track their progress. Initially, a small batch of 100 samples was used to initiate the annotation process, and subsequently, the batch size was increased to 500 as the annotators gained proficiency in the task. To maintain consistency, errors made by

[5] https://developer.twitter.com/en/docs/tutorials/consuming-streaming-data

[6] https://developer.twitter.com/en/docs/twitter-api/v1/tweets/search/api-reference/get-search-tweets

[7] https://github.com/doccano/doccano

TABLE 1. Samples from annotated dataset.

Tweets	Class
T1: persatuan bodoh mana yang cadang sampai 65 tahun tu?? **Translation**: Which stupid association proposes that the age of retirement be extended to 65 years old?	Hate
T2: dah macam cina dari china, vietnam makan anjing **Translation**: It's like the Chinese from China, where Vietnamese eating dogs	Hate
T3: terima kasih untuk hari ini akan ku kenang sampai mati **Translation**:Thank you for today, I will remember it until I die.	Non-Hate
T4: seksa la gila bila batuk sampai nak terkeluar anak tekak. **Translation**:It's really torturous coughing till the uvula is about to come out.	Non-Hate

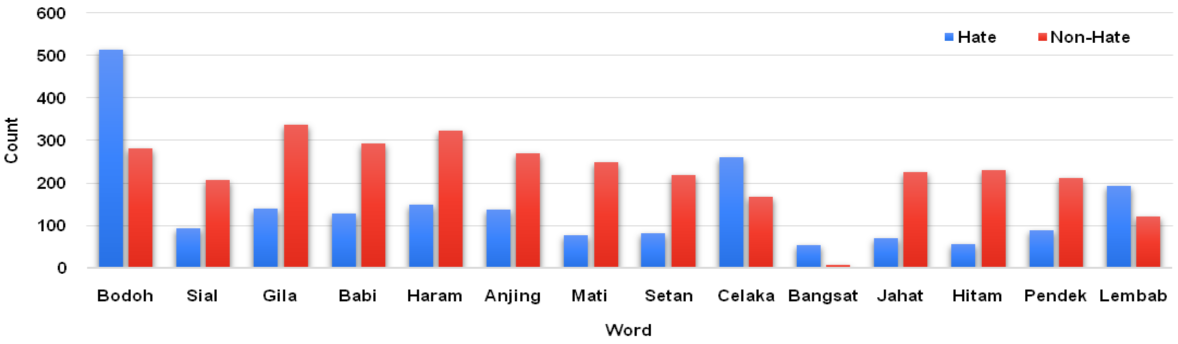

FIGURE 1. Keyword-wise distribution of hate and non-hate tweets in the *HateM* dataset.

annotators in previous batches were corrected during the annotation process. To determine the final hate speech labels, we employed a majority voting method after the completion of each set of annotations. In cases where the selections of the three annotators varied, we sought the assistance of an expert annotator to break the tie. Moreover, annotators were instructed to annotate the posts without consideration for any specific demographic, religion, or other factors. To ensure the quality of annotations, we calculated the inter-annotator agreement (IAA) using Fleiss' Kappa score [37]. The obtained IAA score for the hate speech detection (HSD) task was 0.85, indicating that the dataset was of acceptable quality.

D. DATASET STATISTICS

Our developed *HateM* dataset comprises 4892 tweets, with 3002 labeled as non-hate and the remaining 1890 marked as hate. The average post length is 22.35. The distribution of tweets, categorized as either hate or non-hate, based on keywords is illustrated in Figure 1. Upon examining Figure 1, we can observe that the keyword "bodoh" contributes to the highest number of hate tweets. Additionally, an important observation is that, despite all the keywords having offensive or profane connotations, they are often labeled as non-hate. This observation highlights the inherent challenges involved in hate speech detection, where context plays a crucial role. Table 1 provides examples of tweets collected and labeled for the study. The label indicates whether the tweet was classified as "Hate" or "Non-Hate". For instance, Tweet T1 is a statement critiquing a proposal from an unspecified association and is labeled as "Hate". On the other hand, Tweet T4, which expresses personal discomfort, is labeled as "Non-Hate". This table offers a glimpse into the nature of the data used in our study and illustrates the diversity of expressions that were classified and analyzed.

IV. METHODOLOGY

The suggested approach for detecting hate speech in Malay is explained in this section.

Problem Defination: The task is to develop a binary text classification model for automatically categorizing tweets into hate or non-hate categories. Let $I = [X_t, h_t]_{t=1}^{N}$, denote a set of N instances, where X_t represents the input sentence and h_t represents the corresponding hate label for the t-th instance. The objective of our proposed framework is to learn the optimal model parameters denoted by θ that maximize the probability of predicting the correct hate label for each input sentence, as expressed in Equation (1):

$$\underset{\theta}{\operatorname{argmax}} \left(\prod_{t=1}^{N} P(b_t|X_t, \theta) \right) \tag{1}$$

where X_t represents the input sentence for which the predicted hate label (b_t) is to be determined.

A. TEXT EMBEDDING GENERATION

Text embedding, which represents textual data as numerical vectors, is a crucial step in preparing text data for machine learning models. In this study, we explored two different approaches for generating text embeddings.

i) **XLNet** is a state-of-the-art language model that incorporates bidirectional context, autoregressive modeling, and permutation-based training to generate contextualized text embeddings. Unlike traditional BERT, XLNet employs a permutation-based training approach, where the model is trained on all possible permutations of the

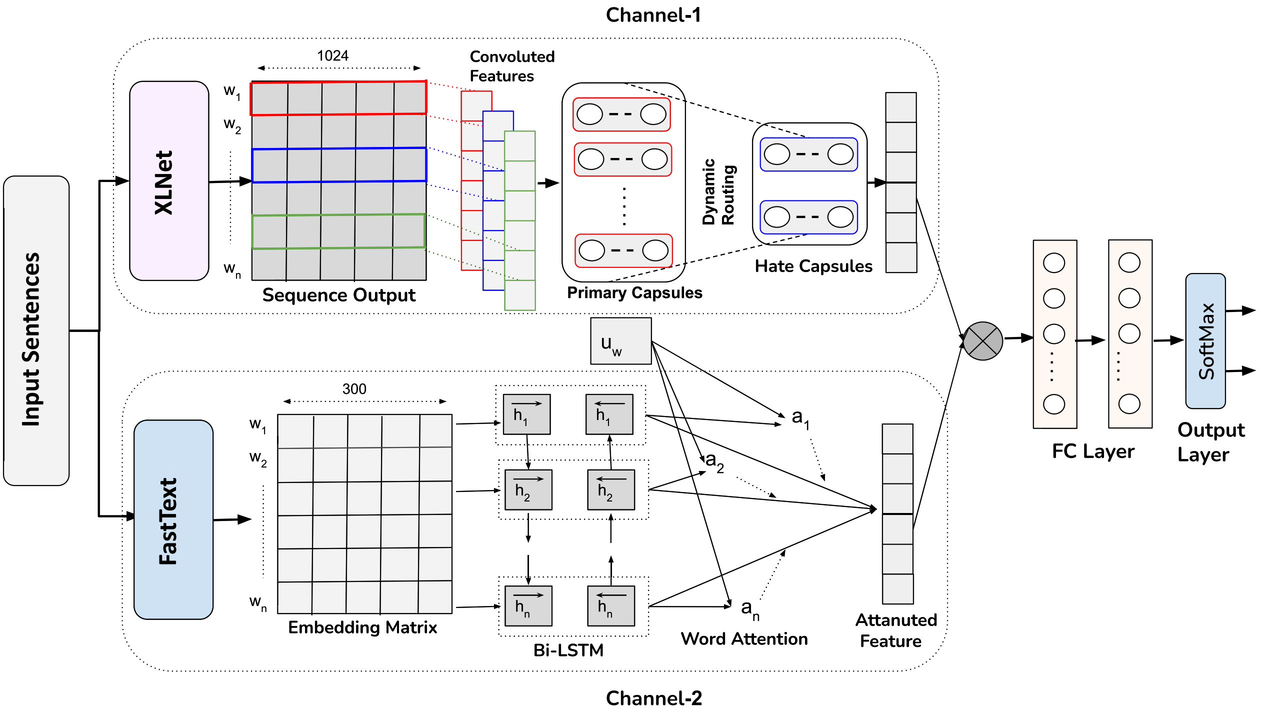

FIGURE 2. ***XLCaps*** **architecture.**

words in a sentence, rather than just masked words. This enables the model to learn more robust and contextually-aware representations for each word.

ii) **FastText** [38] is an unsupervised, lightweight text embedding approach that represents words as continuous bag of character n-grams, capturing subword information and allowing for out-of-vocabulary (OOV) word representations. This makes FastText more robust to handle noisy social media data, which often contain misspellings, abbreviations, and other non-standard language usage. Unlike Word2Vec or GloVe, which rely on word-level representations, FastText can handle OOV words, learn representations for rare words, and capture morphological information, making it more suitable for handling diverse and noisy text data typically found in social media, online forums, and other user-generated content.

XLNet and FastText capture text representations from different perspectives. XLNet, being a contextualized language model, captures contextual information and relationships between words in a sentence, while FastText captures subword information and is robust to handle OOV words and rare words. By combining both approaches, we can leverage the complementary information they provide, potentially leading to better overall performance.

B. PROPOSED XLCaps MODEL

The proposed *XLCaps* model incorporates two distinct channels to capture different aspects of input sentences. The overall architecture of our proposed *XLCaps* model is illustrated in Figure 2. The first channel employs XLNet followed by a capsule network, while the second channel uses pre-trained FastText embedding followed by Bi-GRU with attention. Given an input sentence $W = \{w_1, w_2, \ldots w_n\}$ comprising n words, both channels process the input using a series of operations as follow.

1) CHANNEL-1

a: XLNet

The XLNet model processes the input sentence W and generates a sequence output $E_B \in \mathbb{R}^{n \times d}$ of dimensions $max\_sequence\_length \times 768$. Our experimentation revealed that setting n to 128 produces better results. The output E_B is then fed through Convolutional neural network (CNN) layers for the purpose of abstract feature extraction.

b: CNN [39]

Let the output feature map of this CNN layer be denoted as F. The element-wise dot product is performed between the output sequence E_B and different filters c_i of size $h \times d$ in the CNN layer. This produces the feature map f_i which corresponds to the presence of a particular n-gram in the input sentence. The dimension of F is given by $(n - h + 1) \times k$, where h is the filter size and k is the number of filters used. Therefore, F is a collection of k feature maps obtained by sliding the filters over the entire input sequence.

$$F = [F_1, F_2, F_3, \ldots, F_k]. \tag{2}$$

In our proposed *XLCaps* model, instead of applying a pooling operation to the feature maps, we have used a capsule network [14] to retain the special features that are often lost during pooling. This capsule network helps to preserve the spatial relationships between the features and enables our model to effectively model complex hierarchical structures within the input text.

c: PRIMARY CAPSULE LAYER

The capsule network's first layer combines the convolutional features produced by the CNN and creates primary capsules that represent each element in the feature maps using a group of neurons, thereby preserving local word order and semantic representations of words as instantiation parameters, rather than scalar values. This technique facilitates encoding intricate information about spatial relationships among input features and capturing nuanced semantic features. To generate a set of capsules, denoted as $p_i \in \mathbb{R}^l$, a kernel, denoted as K_i, is applied over the feature maps F, where l is the number of neurons in a capsule. Within the main capsule layer, a channel C_i consisting of a collection of capsules p_i is defined as follows:

$$C_i = \imath(F * K_i + b_i), \tag{3}$$

where $\imath$ refers to a non-linear activation function known as the squashing function, and b_i is a bias term. This process produces a concise and informative representation of the input data in a hierarchical manner, which facilitates capturing complex relationships between input features.

d: DYNAMIC ROUTING BETWEEN CAPSULES

In this layer, each capsule in the previous layer sends its output vector to all capsules in the next layer. The coupling coefficient between capsule i in the previous layer and capsule j in the next layer, denoted as $c_{i,j}$, is determined by a softmax function over all capsules in the next layer, and is calculated as follows:

$$p_{i,j} = \frac{\exp(b_{i,j})}{\sum_k \exp(b_{i,k})}, \tag{4}$$

where $b_{i,j}$ is the log prior probability that capsule i should be coupled with capsule j. The output of each capsule in the next layer is then calculated as a weighted sum of the predictions from all capsules in the previous layer, weighted by the coupling coefficients:

$$s_j = \sum_i p_{i,j}\hat{a}_{j|i} \ \ and \ \ \hat{a}_{j|i} = W_{ij}a_i, \tag{5}$$

where $\hat{a}_{j|i}$ is the prediction vector of capsule i for the presence of an entity of class j and is defined as the dot product between the output vector of capsule i and a transformation matrix $W_{i,j}$, which learns to represent the instantiation parameters between capsule i and class j. Finally, the output vector of each capsule j is passed through a non-linear squashing function to ensure that its length is between 0 and 1.

e: HATE CAPSULE LAYER

The final layer of the proposed capsule network is the hate capsule layer, which consists of k capsules with 16-dimensional instantiated parameters. In this layer, each capsule is dedicated to identifying a specific type of hate speech. The hate capsules are generated by routing the output of the previous layer to the final layer. The output of the hate capsule layer is a flattened 1D vector of dimension ($k \times 16$). This vector is concatenated with the attenuated features generated by channel-2, which captures the long-term dependencies and the context of the input text.

C. CHANNEL-2

1) BI-GRU

To enhance the contextual representation of the input sequence, we have integrated a Bi-GRU layer that takes the embedding vector generated by the fastText model as its input. By processing the input sequence in both forward and backward directions, the Bi-GRU layer can capture contextual information in both directions. At each time step, the hidden state h_t of the Bi-GRU is obtained by concatenating the hidden state of the forward GRU $\overrightarrow{h_t}$ and the hidden state of the backward GRU $\overleftarrow{h_t}$. Consequently, the output of the Bi-GRU layer is a sequence of hidden states H_e that includes all the hidden states of the input sequence. This representation can be expressed as $H_e = [h_1, h_2, h_3, \ldots, h_n]$.

2) ATTENTION LAYER

We incorporate a word attention layer after the Bi-GRU layer, which allows the model to selectively focus on important words in the sentence. The word attention mechanism uses a weighted sum to compute a sentence representation based on the attention scores assigned to each word. Formally, given the hidden state h_i of the Bi-GRU at time step i and the weight vector u_a, the attention score α_i for the i-th word is computed as

$$\alpha_i = \frac{\exp(u_a^T h_i)}{\sum_{j=1}^{n} \exp(u_a^T h_j)}, \tag{6}$$

where n is the length of the input sentence. The sentence representation is then obtained as the weighted sum of the Bi-GRU hidden states:

$$s = \sum_{i=1}^{n} \alpha_i h_i. \tag{7}$$

The resulting sentence representation is then concatenated with the output of the hate capsule layer.

3) FC LAYERS

The concatenated outputs of the XLNet+Capsule and FastText+GRU+Attention models form a combined representation, denoted as J, for the input sentence X. Subsequently, this representation J is fed into fully connected layers, consisting of FC_1 with 200 neurons and FC_2 with 100 neurons. Finally, a softmax output layer is applied to predict the probabilities of the sample belonging to the target classes.

D. LOSS FUNCTION

For the purpose of parameter optimization and backpropagation of loss, the categorical cross-entropy loss function $L_{CE}(\hat{Y}, Y)$ has been utilized in this study. It is defined as follows:

$$L_{CE}(\hat{Y}, Y) = -\frac{1}{N}\sum_{j=1}^{M}\sum_{i=1}^{N} Y_i^j log(\hat{Y}_i^j), \quad (8)$$

where $\hat{Y}_i^j$ represents the predicted label and Y_i^j represents the true label. The term N denotes the number of tweets in the dataset, while M represents the number of classes.

E. INTUITION BEHIND TWO CHANNELS APPROACH

Our proposed model has two channels: one consists of XLNet+Capsule and the other consists of FastText+RNN (Bi-GRU).

1) FastText AND XLNet

The initial hurdle in handling Malay text data is the high level of noise it contains, which includes short words, abbreviations, and misspelled words. To address this challenge, we utilized two distinct embeddings, namely XLNet and FastText, to effectively represent the Malay text. Furthermore, social media texts often include Out-of-Vocabulary (OOV) words, making it challenging to capture the morphological information lost by pre-trained word embedding models. In comparison, FastText's vectors incorporate character-level representations, which is not the case for word2vec [40] and GloVe [41] that use word-level representations. Meanwhile, XLNet is a bidirectional language model that considers both preceding and following words when generating word embeddings, enabling it to capture contextual relationships between words more accurately than unidirectional models. Our approach demonstrates that the combination of XLNet and FastText embeddings can more effectively handle noisy Malay data.

2) CAPSULE AND GRU

To effectively extract features that are both position-invariant and local, CNNs are typically employed [42]. However, for tasks that require long-range semantic dependencies such as language modeling and machine translation, Recurrent Neural Networks (RNNs) tend to perform better than certain local key phrases. In our study, since the average length of posts in our developed dataset is 124.36 (which is quite lengthy) and we are performing classification, we require both local key phrase features and long-range dependency. Therefore, our proposed architecture includes both Channel-1 (Capsule) and Channel-2 (GRU) to address these requirements.

3) WHY CAPSULE INSTEAD OF CNN

Traditional CNNs can be limited in their ability to preserve spatial features, such as object position and rotation in images or word order in the text, due to the pooling operation. The capsule network, introduced by Sabour et al. [14], is a novel architecture that aims to overcome this limitation. Capsule networks utilize an iterative routing process to determine the attribution of credit between nodes in different layers. In the case of Malay text, where word order is critical due to its unique sentence structure, Capsule networks can effectively handle noisy data by encoding spatial relationships between features. As a result, Capsule networks have gained popularity in the field of text classification [43], [44]. Hence, we have included Capsule networks in our proposed model to improve the performance of our model on Malay text data.

V. EXPERIMENTAL RESULTS AND ANALYSIS

This section discusses the results of multiple baseline models and our suggested model, both of which were tested on the *HateM* dataset.

A. EXPERIMENTAL SETTINGS

This section details our experimental setup, manifesting our work's various hyperparameters and experimental settings. All our experiments are performed on a machine with an AMD EPYC 7552 48-Core Processor, 512 GB DDR4 RAM, and 5x Nvidia Ampere A100 GPUs totaling 200 GB of graphics memory. To prepare for the experiments, we partitioned the dataset into testing, validation, and training sets, with ratios of 10%, 10%, and 80%, respectively. The models were trained ten times with different random splits to ensure robustness, and the average performance was reported. Various network configurations were tested, and we achieved the best results with a batch size of 16, the learning rate of $1e^{-4}$, and 30 epochs. All models were implemented using Scikit-Learn and PyTorch.

B. BASELINES SETUP

To ensure a comprehensive evaluation of our proposed model, we established a suite of machine and deep learning-based baselines. We considered four commonly used machine learning models, namely Naive Bayes, SVM, Decision Tree, and Random Forest, with various embedding techniques. For machine learning baselines, we utilized the 768-dimensional pooled output of XLNet and mBERT. For FastText embedding, we employed a pre-trained Malay FastText model to extract the embedding of each token and computed the average to obtain a 300-dimensional vector representing the entire sentence.

We established various variants of single-channel and double-channel deep learning baselines by varying the input embedding models followed by different deep learning models such as CNN, Bi-GRU, and Capsule network. In the case of single-channel baselines, we first passed the input tweet through XLNet or FastText to generate a 2-d embedding matrix (E_m) of dimension (max-sequence length × d), where d = 300 for FastText and d = 1024 for XLNet. We then passed this E_m through different deep learning models as follows:

(i) **Capsule network:** The input embedding E_m was fed into a 1D CNN with 64 window size 2 filters. The convulted feature was then transferred via the Capsule network, and the hatred capsule layer's output was flattened and routed through an FC layer. Finally, for the final prediction, a softmax layer was used.

(ii) **CNN:** Here, E_m was passed through a 1D CNN with 64 filters of window size 2. We then performed Average Pooling on convoluted features followed by a softmax output layer.

(iii) **Bi-GRU:** Input embedding E_m was fed through Bi-GRU with 128 hidden states. The output of the Bi-GRU layer was then passed through a fully connected (FC) layer with 100 neurons. Finally, a softmax layer was employed for the final prediction.

(iv) **XLM-RoBERTa** [45]: We fine-tuned the XLM-RoBERTa model by adding a softmax layer to the top [CLS] token. XLM-RoBERTa, a multilingual version of RoBERTa, was pre-trained on a filtered CommonCrawl dataset of 2.5TB, which includes 100 languages, including Malay.

Overall, these single-channel baselines served as a foundation for the double-channel baselines, where we retained the same structure as the proposed model, varying the channel configurations such as embedding generation techniques followed by different deep learning models. For instance, we experimented with variants such as XLnet+BiGRU; Fasttext+BiGRU, XLnet+caps; Fasttext+caps, XLnet+BiGRU; Fasttext+caps, XLnet+ CNN; Fasttext+BiGRU, etc. Ultimately, the final prediction was made by passing the joint representation vector (J) to two FC layers, each with 100 neurons, followed by a softmax layer.

C. FINDINGS FROM EXPERIMENTS

Table 2 presents the evaluation results of our proposed model, *XLCaps*, and other baselines in terms of accuracy, precision, recall, and macro F1 score. The following observations can be made from the table 2:

(i) SVM consistently outperforms the other machine learning baselines in terms of F1 score, achieving the best F1 score of 64.68% among different embedding settings, i.e., XLNET, mBERT, and FastText.

(ii) Our proposed model *XLCaps* significantly outperforms the best machine learning baseline (XLNet+SVM) with an improved F1 score of 15.73%.

(iii) In terms of single-channel deep learning baselines, Capsule network outperforms BiGRU and CNN with XLNet embedding, with improvements of 2.74% and 1.97% in F1 score, respectively. Conversely, BiGRU performs better than the others with FastText embedding. XLNet+Caps achieved the best F1 score of 77.48% among the single-channel-based deep learning baselines, surpassing XLNet+SVM by 12.80% in F1 score. This finding supports the efficacy of deep learning models over machine learning models for hate speech detection in noisy social media data.

TABLE 2. Results of different baselines and proposed frameworks for hate speech detection (HD) task; Pre - Precision, Rec - Recall, Acc - accuracy, bold — best.

Embedding	Model	Hate			
		Pre	Rec	F1	ACC
Machine Learning Baselines					
XLNet	Naive Bayes	**56.78**	**58.13**	**56.54**	**58.47**
	SVM	64.27	65.32	**64.68**	65.24
	Decision Tree	56.17	57.36	57.14	57.38
	Random forest	62.11	63.19	56.78	63.26
mBERT	Naive Bayes	56.21	52.34	53.54	52.37
	SVM	62.16	63.42	**62.38**	64.15
	Decision Tree	56.11	57.27	57.08	57.60
	Random forest	62.11	63.19	56.78	63.23
FastText	Naive Bayes	58.16	53.27	53.47	52.50
	SVM	62.21	63.56	**61.28**	63.22
	Decision Tree	56.13	57.39	57.23	56.58
	Random forest	64.36	65.41	59.88	64.55
Deep Learning Baselines (Single Channel)					
XLNet	Capsule	79.64	76.35	**77.48**	76.25
	BiGRU	76.86	74.56	75.53	75.57
	CNN	73.63	75.58	74.89	74.83
FastText	Capsule	74.89	72.91	73.67	73.26
	BiGRU	75.27	73.34	**74.67**	74.72
	CNN	69.81	72.67	71.92	72.13
XLM-RoBERTa		68.28	67.01	67.37	67.11
Deep Learning Baselines (Two Channel)					
XLnet+BiGRU; Fasttext+BiGRU		79.46	78.61	78.85	78.65
XLnet+Capsule; Fasttext+Capsule		78.34	77.63	77.81	77.63
XLnet+CNN; Fasttext+CNN		77.91	76.91	77.16	76.91
XLnet+BiGRU; Fasttext+Capsule		80.14	79.46	**79.65**	79.46
XLnet+BiGRU; Fasttext+CNN		78.44	77.73	77.93	77.73
XLnet+CNN; Fasttext+BiGRU		78.60	78.03	78.21	78.03
Proposed Model (XLCaps)					
XLnet+Capsule; Fasttext+BiGRU		**82.21**	**81.13**	**80.41**	**80.69**

(iv) The singular results of channel 1 (XLNet+Caps) and channel 2 (FastText+BiGRU) are 76.25% and 74.72% in terms of F1 score, respectively. However, combining both channels achieves an F1 score of 80.69%, indicating the efficiency of the combination of XLNet and FastText embeddings for handling noisy text.

(v) The Capsule network performs better than Bi-GRU when embedded with XLNet, and the reverse occurs for the FastText embedding, which is why we keep XLNet+Caps for channel 1 and FastText+BiGRU for channel 2 in our proposed model.

(vi) We evaluated other variants of the proposed model and concluded that *XLCaps* (XLnet+Caps; Fasttext+ BiGRU) achieved the best performance with an F1 score of 80.41, significantly outperforming all the baselines.

(vii) It is noted that XLNet performs better than mBERT in our problem statement, which is why we only reported XLNet results in deep learning-based baselines.

(viii) When comparing XLNet with FastText, we observe that XLNet embedded with any deep learning models always performs better than FastText. A similar trend is also observed in the case of machine learning baselines, except for Random Forest. This observation indicates the advantage of the transformer-based pre-trained language model XLNet over FastText in terms of efficient embedding generation of noisy social media text data.

(ix) Our proposed model *XLCaps* significantly outperforms the XLM-RoBERTa finetuning model.

D. ERROR ANALYSIS

We meticulously examined the data instances misclassified by the proposed model to conduct a thorough error analysis. In our investigation, we looked at the following examples.

(i) T1: *cuba share lagi kat komen, apa lagi yang kita buat, ambil hak orang lain tanpa kita sedar atau sedar tapi buat bodoh*,

Translation: Try to share it in the comments, what else do we do? Taking away someone else's rights without realizing it or realizing it but acting foolish.

The predicted classification label of this tweet as hate speech was incorrect, as a closer examination reveals that it lacks such characteristics. The model's identification of the term "bodoh" (meaning "stupid" or "foolish" in Malay) as a profanity, likely due to its association with other previously trained tweets, could explain the misclassification. Notably, "bodoh" is a commonly used Malay swear word, but it also has a dual meaning, often used to describe someone as foolish.

(ii) T2: *@user Bodoh macam anjing nyusah kn org*,

Translation: @user Stupid like a dog, annoying people

Despite being predicted as non-hate speech, this tweet demonstrates hate speech characteristics. The language used in this case includes derogatory terms directed at the rail provider in Malaysia, specifically the term "anjing" (which means "dog" in Malay). It is important to note that the misclassification could have occurred as a result of the occasional use of "anjing" as a swear word, emphasising the importance of context when analysing such instances.

(iii) T3: *houseman dulu dulu takde sistem shift kerja jam nonstop houseman sekarang kerja shift kerja jam je sehari itupun nak bising lembik*,

Translation: In the past, housemen didn't have a shift system, they worked non-stop for hours. Now they have shift work, just a few hours a day, and they still complain and feel weak.

Despite the lack of any hostile intent, this tweet is mistakenly labelled as hate speech. It is, in reality, a sarcastic remark using the term "lembik" (Malay for "weak"). Despite having set work schedules, the tweet criticises today's housemen (junior doctors) for moaning and lacking energy at work. This statement is not directed towards any one person, but rather just an observation. The misclassification happened as a result of a misunderstanding of the term "lembik" as hate speech.

VI. CONCLUSION

This paper addresses the critical issue of hate speech detection in the Malay language, recognizing its significant impact on individuals and society. We introduce the first benchmark dataset for hate speech detection in Malay, comprising over 4,892 annotated tweets, providing a valuable resource for future research. Our two-channel deep learning model, *XLCaps*, effectively handles the noise in Malay language posts by combining XLNet with Capsule networks in one channel and FastText with Bi-GRU in the other. *XLCaps* surpasses the baseline models, achieving impressive accuracy and F1 scores of 80.69% and 81.41%, respectively, demonstrating the efficacy of the two-channel approach. Our findings highlight the superiority of the XLNet+Capsule combination in a single channel over mBERT and FastText, generating efficient features for hate speech detection in deep learning and machine learning models.

In future work, we aim to enhance the *HateM* dataset by incorporating sentiment and emotion labels to investigate their impact on hate speech detection in Malay. Our study also explores the model's performance in detecting hate speech by considering specific types of noise, such as misspellings and abbreviations.

REFERENCES

[1] J. T. Nockleby, "Hate speech in context: The case of verbal threats," *Buff. L. Rev.*, vol. 42, p. 653, Jan. 1994.

[2] T. K. H. Chan, C. M. K. Cheung, and R. Y. M. Wong, "Cyberbullying on social networking sites: The crime opportunity and affordance perspectives," *J. Manage. Inf. Syst.*, vol. 36, no. 2, pp. 574–609, Apr. 2019.

[3] M. Dadvar, D. Trieschnigg, and F. de Jong, "Experts and machines against bullies: A hybrid approach to detect cyberbullies," in *Proc. Can. Conf. Artif. Intell.* Cham, Switzerland: Springer, 2014, pp. 275–281.

[4] K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the detection of textual cyberbullying," in *Proc. Int. Conf. Weblog Social Media*. Princeton, NJ, USA: Citeseer, 2011, pp. 1–7.

[5] K. Reynolds, A. Kontostathis, and L. Edwards, "Using machine learning to detect cyberbullying," in *Proc. 10th Int. Conf. Mach. Learn. Appl. Workshops*, vol. 2, Dec. 2011, pp. 241–244.

[6] S. Agrawal and A. Awekar, "Deep learning for detecting cyberbullying across multiple social media platforms," in *Proc. Eur. Conf. Inf. Retr.* Cham, Switzerland: Springer, 2018, pp. 141–153.

[7] 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.

[8] 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.

[9] F. D. Vigna, A. Cimino, F. Dell'Orletta, M. Petrocchi, and M. Tesconi, "Hate me, hate me not: Hate speech detection on Facebook," in *Proc. 1st Italian Conf. Cybersecurity (ITASEC)*, 2017, pp. 86–95.

[10] M. O. Ibrohim and I. Budi, "Multi-label hate speech and abusive language detection in Indonesian Twitter," in *Proc. 3rd Workshop Abusive Lang. Online*, 2019, pp. 46–57.

[11] K. Pasupa, W. Karnbanjob, and M. Aksornsiri, "Hate speech detection in Thai social media with ordinal-imbalanced text classification," in *Proc. 19th Int. Joint Conf. Comput. Sci. Softw. Eng. (JCSSE)*, Jun. 2022, pp. 1–6.

[12] M. Choudhury, R. Saraf, V. Jain, A. Mukherjee, S. Sarkar, and A. Basu, "Investigation and modeling of the structure of texting language," *Int. J. Document Anal. Recognit. (IJDAR)*, vol. 10, nos. 3–4, pp. 157–174, Dec. 2007.

[13] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–18.

[14] S. Sabour, N. Frosst, and G. E Hinton, "Dynamic routing between capsules," 2017, *arXiv:1710.09829*.

[15] D. A. Simanjuntak, H. P. Ipung, C. lim, and A. S. Nugroho, "Text classification techniques used to faciliate cyber terrorism investigation," in *Proc. 2nd Int. Conf. Adv. Comput., Control, Telecommun. Technol.*, Dec. 2010, pp. 198–200.

[16] P. Fortuna and S. Nunes, "A survey on automatic detection of hate speech in text," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–30, Jul. 2019.

[17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 3111–3119.

[18] Y. Mehdad and J. Tetreault, "Do characters abuse more than words? in *Proc. 17th Annu. Meeting Special Interest Group Discourse Dialogue*, 2016, pp. 299–303.

[19] S. Zimmerman, U. Kruschwitz, and C. Fox, "Improving hate speech detection with deep learning ensembles," in *Proc. 11th Int. Conf. Lang. Resour. Eval. (LREC)*, 2018, pp. 1–8.

[20] H. T.-T. Do, H. D. Huynh, K. Van Nguyen, N. L.-T. Nguyen, and A. G.-T. Nguyen, "Hate speech detection on Vietnamese social media text using the bidirectional-LSTM model," 2019, *arXiv:1911.03648*.

[21] K. Maity and S. Saha, "Bert-capsule model for cyberbullying detection in code-mixed Indian languages," in *Proc. Int. Conf. Appl. Natural Lang. Inf. Syst.* Cham, Switzerland: Springer, 2021, pp. 147–155.

[22] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval)," 2019, *arXiv:1903.08983*.

[23] 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.

[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. Paul and S. Saha, "CyberBERT: BERT for cyberbullying identification," *Multimedia Syst.*, vol. 28, pp. 1897–1904, Nov. 2020.

[26] H. Mubarak, K. Darwish, and W. Magdy, "Abusive language detection on Arabic social media," in *Proc. 1st Workshop Abusive Lang. Online*, 2017, pp. 52–56.

[27] Z. Mossie and J.-H. Wang, "Social network hate speech detection for amharic language," *Comput. Sci. Inf. Technol.*, pp. 41–55, Apr. 2018.

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

[29] T. Van Huynh, V. D. Nguyen, K. Van Nguyen, N. L.-T. Nguyen, and A. G.-T. Nguyen, "Hate speech detection on Vietnamese social media text using the Bi-GRU-LSTM-CNN model," 2019, *arXiv:1911.03644*.

[30] R. Kumar, A. N. Reganti, A. Bhatia, and T. Maheshwari, "Aggression-annotated corpus of Hindi-english code-mixed data," 2018, *arXiv:1803.09402*.

[31] M. R. Karim, S. K. Dey, T. Islam, S. Sarker, M. H. Menon, K. Hossain, M. A. Hossain, and S. Decker, "DeepHateExplainer: Explainable hate speech detection in under-resourced Bengali language," in *Proc. IEEE 8th Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Oct. 2021, pp. 1–10.

[32] T. L. Sutejo and D. P. Lestari, "Indonesia hate speech detection using deep learning," in *Proc. Int. Conf. Asian Lang. Process. (IALP)*, Nov. 2018, pp. 39–43.

[33] T. X. Moy, M. Rahem, and R. Logeswaran, "Multilingual hate speech detection," *Int. J. Multidisciplinary Res. Publications*, vol. 4, no. 10, pp. 19–28, 2022.

[34] Z. Zainol, S. Wani, P. N. Nohuddin, W. M. Noormanshah, and S. Marzukhi, "Association analysis of cyberbullying on social media using Apriori algorithm," *Int. J. Eng. Technol.*, vol. 7, pp. 72–75, Dec. 2018.

[35] A. Guterres. (2019). *United Nations Strategy and Plan of Action on Hate Speech*. Taken From. [Online]. Available: https://www.un.org/en/genocideprevention/documents/U

[36] Mai ElSherief, Vivek Kulkarni, Dana Nguyen, William Yang Wang, and Elizabeth Belding, "Hate lingo: A target-based linguistic analysis of hate speech in social media," in *Proc. Int. AAAI Conf. Web Social Media*, vol. 12, 2018.

[37] J. L. Fleiss, "Measuring nominal scale agreement among many raters," *Psychol. Bull.*, vol. 76, no. 5, pp. 378–382, Nov. 1971.

[38] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," 2018, *arXiv:1802.06893*.

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

[40] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1188–1196.

[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] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of CNN and RNN for natural language processing," 2017, *arXiv:1702.01923*.

[43] J. Kim, S. Jang, E. Park, and S. Choi, "Text classification using capsules," *Neurocomputing*, vol. 376, pp. 214–221, Feb. 2020.

[44] Y. Cheng, H. Zou, H. Sun, H. Chen, Y. Cai, M. Li, and Q. Du, "HSAN-capsule: A novel text classification model," *Neurocomputing*, vol. 489, pp. 521–533, Jun. 2022.

[45] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," 2019, *arXiv:1911.02116*.

KRISHANU MAITY received the M.Tech. degree in information technology from the Kalyani Government Engineering College, Nadia, India. He is currently a Research Scholar with the Department of Computer Science and Engineering (CSE), Indian Institute of Technology Patna, India. His research interests include natural language processing specifically code-mixed languages, deep learning, and user behaviors' analysis in social media.

SHAUBHIK BHATTACHARYA received the B.Tech. degree in computer science and engineering (CSE) from the Indian Institute of Information Technology Kalyani, Nadia, India, in 2019. He is currently pursuing the M.Tech. degree in computer science and engineering (CSE) with the Indian Institute of Technology Patna, Patna. His research interests include natural language processing specifically hate speech detection and deep learning.

SRIPARNA SAHA (Senior Member, IEEE) received the M.Tech. and Ph.D. degrees in computer science from the Indian Statistical Institute, Kolkata, India. From September 2009 to June 2010, she was a Postdoctoral Research Fellow with the University of Heidelberg, Germany. From September 2010 to January 2011, she was a Postdoctoral Research Fellow with the Department of Information Engineering and Computer Science, University of Trento, Italy. She is currently an Associate Professor with the Department of Computer Science and Engineering, Indian Institute of Technology Patna, India. She has more than 400 publications with 7000 citations. Her current research interests include text mining pattern recognition, natural language processing, multiobjective optimization, and biomedical information extraction.

MANJEEVAN SEERA received the Ph.D. degree in computational intelligence from Universiti Sains Malaysia. He has over 17 years of experience in both academia and industry. He is currently an Associate Professor of business analytics with the School of Business, Monash University Malaysia. His research specialization is on machine learning principles and applications in finance and engineering. His recent research interest includes the design and development of advanced machine learning models for Fintech applications, particularly the detection and prediction of fraudulent financial transactions.