IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 October 2023, accepted 28 November 2023, date of publication 17 January 2024, date of current version 24 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3349958

RESEARCH ARTICLE

The Effect of Phrase Vector Embedding in Explainable Hierarchical Attention-Based Tamil Code-Mixed Hate Speech and Intent Detection

V. SHARMILA DEVI[1], S. KANNIMUTHU[1], (Senior Member, IEEE), AND ANAND KUMAR MADASAMY[2], (Member, IEEE)

[1]Department of Information Technology, Karpagam College of Engineering, Coimbatore 641032, India
[2]Department of Information Technology, National Institute of Technology Karnataka, Surathkal 575025, India

Corresponding author: Anand Kumar Madasamy (m_anandkumar@nitk.edu.in)

ABSTRACT The substantial growth in social media users has led to a significant increase in code-mixed content on social media platforms. Millions of users on these platforms upload pictures and videos and post comments regarding their recent or exciting activities. Responding to this uploaded content, a few users occasionally use offensive language to insult others or specific groups. Social media platforms encounter challenges identifying and removing hate speech and objectionable content in various languages. Hate speech, in its general sense, refers to harmful posts directed at individuals or groups based on factors such as their sexuality, religion, community affiliation, disability, and others. Typically, offensive language is directly or indirectly utilized in hate speech posts to insult someone, causing psychological distress to users. In light of this, we propose developing a system to automatically block, remove, or report posts written in code-mixed Tamil containing hate speech. We have gathered code-mixed Tamil comments from Twitter and the Helo App, categorizing them as hate speech and classifying their intent. We have identified three categories of hate speech intent, namely Targeted Individual (TI), Targeted Group (TG), and Others (O). The Targeted Individual (TI) class encompasses posts aimed at a specific individual target. At the same time, the Targeted Group (TG) category primarily focuses on identifying people based on their religion, community, gender, and other characteristics. The Others (O) category encompasses untargeted offensive posts and other posts containing offensive language. In this context, we propose using a phrase-based, Explainable Hierarchical Attention model for hate speech detection. The results demonstrate that the proposed method is more effective in identifying and explaining hate speech and offensive language in social media posts.

INDEX TERMS Social media, hate speech intent classification, offensive language, hierarchical attention network, phrase embedding.

I. INTRODUCTION

Social media platforms enable the public to express their ideas and opinions about specific individuals or topics and engage in discussions. In recent years, there has been a significant proliferation of offensive language on social media platforms like Twitter and Facebook. The substantial increase in social media users has resulted in a rise in hate speech posts within the public domain. Given the sheer volume of data, manually identifying, moderating, or removing offensive posts presents a considerable challenge. In this study, hate speech is defined as the act of insulting others using hurtful words or obscene text, typically exchanged between users. Some users post comments or tweets, often directed at specific individuals or groups, employing hate speech and profanity. Hate speech often

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

targets various characteristics, including gender, religion, race, and disability. These offensive posts cause significant harm, leading to psychological distress and mental trauma for social media users. Social media platforms such as Twitter, Facebook, and Instagram have implemented reporting options, allowing users to flag problematic posts. Specific research communities have redefined the issue of hate speech, encompassing aggression detection, cyberbullying, and the detection of abusive or offensive language.

Numerous efforts have been made to automate the detection of hate speech on social media platforms by applying Artificial Intelligence (AI) models. This pursuit arises from content moderators' need to possess dependable hate speech detection tools to facilitate the semi-automatic removal of offensive posts from user-generated content. Most of the studies in this field rely solely on labelled data, enabling them to perform the classification task aimed at identifying hate speech efficiently. Over the past few years, Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks have emerged as the preferred choices for text classification. It is worth noting that the analysis of code-mixed content in Tamil is gaining increasing prominence within the research community [7], [22].

Over the past few years, research on automated detection of hate speech content in various languages has significantly increased. Notably, datasets for hate speech detection are available for several Indian languages, including HASOC (Sandip Modha et al., 2020) [1], [14], [31], designed for identifying hate speech and offensive content in Indo-European languages. The HASOC dataset (Sandhip Modha et al., 2019) represents a pioneering multilingual dataset encompassing Hindi, German, and English. These datasets originated from content on Twitter and Facebook platforms and were employed in the shared task conducted during FIRE 2019.

In this paper, our objective is to explore the influence of a phrase-based model on the attention mechanism's effectiveness in identifying hate speech in code-mixed Tamil. The attention mechanism constitutes a vital component of modern transformer models. We have compiled code-mixed Tamil content from Twitter and the Helo app, annotating it for hate speech and categorizing its type. This dataset was utilized for hate speech detection in the FIRE 2020 - HASOC contest [13]. Determining the specific type of hate speech content is the most challenging among the various tasks involved. Our study involves evaluating the proposed model and interpreting its behaviour.

The proposed Hierarchical Attention Network (HAN) enhances the effectiveness of hate speech detection while shedding light on the pivotal words in classifying content as hate speech. This model not only assists social networks in removing offensive posts but also offers a method to obscure or filter out profanity in a specific language. Instead of outright blocking or removing a comment or post, this system allows users to receive feedback regarding the post, highlighting which specific words led to its classification as hate speech content. Subsequently, users can modify the comment and post it according to their preferences.

II. PROBLEM DESCRIPTION AND RESEARCH CONTRIBUTIONS

Hate speech detection entails identifying social media posts containing offensive content and discerning the specific nature of the hate speech. While many hate speech datasets are available for English, a corresponding dataset for code-mixed Tamil has been notably absent. Annotating hate speech and its specific nature is a challenging task, and we addressed this by collecting data from various social media platforms and having it annotated by three different annotators. Hate speech detection can be approached as a traditional text classification task. The effectiveness of traditional machine learning methods in this context heavily relies on the quality of feature extraction. However, achieving the optimal feature modelling necessitates expertise, which can be labour-intensive and time-consuming. The proposed hierarchical model mitigates the need for extensive human knowledge and reliance on intricate feature modelling.

Another significant challenge in hate speech classification lies in addressing the imbalanced nature of the data. To tackle this issue, we adopted an approach involving focal loss [11] in place of the more conventional categorical loss. Additionally, we employed context embedding-based text augmentation techniques to account for the minority class during the training process.

The primary goal of the proposed hate speech detection and intent classification system is to enable automatic learning from the annotated Tamil code-mixed hate speech dataset. Once the system has been trained on this dataset, it can be applied to classify posts on social media that it has not encountered before. The dataset has been structured and represented in the following manner,

$$T = \{(p_i, y_i), i = 1, \ldots.., n\}, \quad p_i \in p^k$$

$|T|$ or n represents the number of social media posts in the training set. p denotes the posts and y represent labels.

1) Hate Speech Detection: This module focuses on detecting Hate speech and offensive language in code-mixed Tamil. It is a coarse-grained binary classification in which social media comments are classified into two classes Hate and Offensive (HOF) and Non-Hate and Non-offensive (NOT).
   - Hate and Offensive (HOF) - The post has hate, offensive, harmful and profane words.
   - Non-Hate Offensive (NOT) - The post has no Hate speech offensive content.

   During the annotation part, we labelled the posts as HOF if they had any form of offensive languages like profanity, aggression and hurtful words. Otherwise, we labelled them as NOT. Hate speech detection is a binary classification task classifying whether or not the

given post is hate speech.

$$y_{hate} = \{HOF, NOT\}$$

2) Hate Speech Intent/Target Classification: This task focuses on classifying the intent of offensive posts as targeted individual (TI) or targeted group (TG), or others (O).
   - Targeted Individual (TI): A targeted individual (TI) means the post focuses on one person by using offensive language, and profane, like unacceptable language in the absence of insults and abuse. The usage of nasty words and cursing also.
   - Targeted Group (TG): Targeted group (TG) means the post is offensive language and is targeted to a group of people like women or any gender, religion, or racism etc.
   - Others (O): The post has offensive language, but it's not targeted at individuals or groups, and it mainly focuses on events, policies, organisations etc.

Hate speech intent classification is a multi-class classification task that categorises the intent of hate speech post.

$$y_{intent} = \{TI, TG, O\}$$

A. RESEARCH CONTRIBUTIONS

This scenario presents a phrase-based, explainable Hierarchical Attention Network designed to detect and classify code-mixed social media posts.

The proposed research comprises two modules: the Hate speech detection module, which is designed to determine whether a post contains offensive content, and the Hate speech target categorization module, which classifies the specific target of the hate speech.

The key contributions of this work are as follows:

- We have collected the code-mixed Tamil dataset and annotated them as hate speech and its intent. We have studied the existing benchmark dataset for Hate Speech detection.
- We propose a phrase-based Hierarchical attention network framework to detect offensive content on social media posts and classify its target intent. We have created phrase embedding for Code-Mixed Tamil corpora.
- We have experimented with focal loss and text augmentation using contextualised embedding to handle the imbalanced nature of the Hate Speech intent classification.
- We conducted various experiments to evaluate the proposed model's classification performance of different parameter settings and compared it with existing state-of-the-art models.
- We have used attention weights and visualised the explainable behaviour of the proposed model.

The rest of the paper is discussed as follows. Section III delves into state-of-the-art datasets and conducts a comprehensive literature survey to identify hate speech and offensive language. Section IV provides an in-depth explanation of the process we followed to create the dataset. Moving on to Section V, we provide a detailed illustration of the proposed phrase-based Hierarchical Attention model. Section VI discusses our experimental setup, the parameters employed, the results obtained, and an analysis of explainable attention weights. Finally, in Section VII, we conclude this paper by summarizing our findings and discussing potential avenues for future research.

III. RELATED WORKS

In recent years, there has been extensive research on hate speech detection and identification and other related areas. The term 'hate speech' is often combined with other terms such as 'offensive,' 'profane words,' 'abusive language,' and 'cyberbullying.' To elaborate on them, we must identify the hate speech: (1) Hate speech mainly targets individuals or groups depending on specific characteristics. This form of hate speech aims to harm or demean others due to factors such as their ethnicity, religion, gender, or other defining attributes. (2) To explain a clear idea, hate speech is used to cause harm or promote hatred. (3) Sometimes, they use offensive or profane words.

Pratiwi et al. [21] proposed a strategy to mitigate the issue of hate speech on social media platforms by introducing a system in which users are required to possess unique codes to interact with one another. They analyzed Indonesian tweets as their dataset, focusing on identifying hate speech within this context.

Saroj and Pal [25] compiled a dataset sourced from Facebook and Twitter, encompassing Hindi and English content pertaining to India's parliamentary election in 2019 (PEI data - 2019). They conducted sentiment analysis using various classifiers and categorized the data into three distinct classes: hate speech, offensive but not hateful, and neither hate or abuse. Alsafari et al. [2] curated an Arabic text corpus dataset derived from Twitter, employing four different extraction methods. Within this dataset, four distinct types of hate were identified: religion-based, ethnicity-based, national-based, and gender-based. Alsafari developed classification models for two-, three-, and six-class scenarios, incorporating various feature extraction techniques.

Wang and Ding [28] elaborated on their participation in the SemEval 2019 task, where they focused on the Multilingual Detection of Hate Speech on Twitter. Their study specifically addressed two targets: immigrants and women. They developed an attention-LSTM model incorporating Hierarchical Attention Networks (HAN) and Bidirectional Gated Recurrent Units (BiGRU) alongside capsule models. Baruah et al. [6] introduced a novel deep-learning technique involving the Multi Dimension Capsule Network for sentence representation in classification tasks. Their approach allows them to handle comments written in both Hindi and English, as showcased in the TRAC dataset. Mandal et al. [13], [14] contributed to the field by providing the HASOC dataset, which serves as a valuable resource for identifying offensive

TABLE 1. Summary of hate speech datasets.

Author	Language	Source	Dataset Name	Classes
Burnap and Williams (2016)	English	Twitter	BURNAP dataset	Sexual orientation, race, disability and religion
Waseem and Hovy (2016)	English(16K)	Twitter	WASEEM dataset	Racism, Sexism and others
Sandhip Modha and Thomas Mandip (2019)	Hindi, German and English	Twitter and Facebook	HASOC ((FIRE 2019)	Hate speech / NOT Hate
Sandip Modha, Thomas Mandip et.al., (2020)	Tamil, Hindi, Malayalam, English, German	Twitter and Youtube	HASOC (FIRE 2020)	Hate speech / NOT Hate
Fernquist et.al,, (2019)	3056 English comments	Swedish web	FLKA dataset	Aggression, insult, dislike and neutral
Goa and Huang (2017)	1528 English comments	Fox News Website	GH dataset	Hate speech / Not Hate speech
Gao et.al (2017)	62 million tweets	Twitter	GKH dataset	Hate speech / NOT Hate
De Gibert et al. (2018)	10,568 English	Twitter	GPGC Dataset	Hate speech / Not Hate speech
Hammer (2017)	24,840 English	Youtube	H dataset	Threatening or violent/clean
Haddad et al. (2019)	Tunisian Arabic (6039 comments)	Facebook	HUO dataset	Hateful /abusive /normal
Ishman and Sharmin (2019)	Bengali (5126 comments)	Facebook	IS dataset	HS/Inciteful/Religious /Hatred/Communal hatred /religious comment/ Political
Kumar Sharma et.al (2018)	English (2235) comments	Youtube	KKS dataset	Insulting/ not insulting
Kumar et al. (2018)	Hindi - English (39,000 texts)	Facebook	KRBM dataset	Overtly aggressive/covertly aggressive/not aggressive
Kolhatkar et al.(2019)	English (1043 comments)	Canadian News Website	KWCFST dataset	Constructiveness, toxicity, negation and appraisal
Mubarak et al. (2017)	Arabic	Twitter	MDM dataset	Obscene/offensive but not obscene/clean
Martins et al. (2018)	English (975 comments)	Twitter	MGANH dataset	Hate speech/offensive but not Hate speech/none
Mathur et al. (2018)	Hindi-English (3679 tweets)	Twitter	MSSM dataset	Hate speech /abusive/not offensive
Mossie and Wang (2020)	English (5876) posts	Facebook	MW dataset	Hate speech/ no Hate speech/ethnic/religious/ political/economic
Nascimento et al. (2019)	Brasilian Portuguese (7672 posts)	Twitter	NCCVG	Offensive / non-offensive

language and hate speech. The dataset is divided into two parts: the first part involves Twitter posts in Hindi, German, and English, while the second part includes Tamil and Malayalam content, both in native and native scripts. The posts were primarily sourced from platforms such as YouTube and Twitter.

In their work, Paschalides et al. [18] developed a big-data processing system to detect and identify online hate-related speech, utilizing a big-data approach. They introduced a novel ensemble-based classification algorithm to identify hate speech and enhance the overall performance of the MANDOLA system in detecting hate speech. Alonso et al. [1] addressed the challenge posed by the daily generation of vast amounts of content on social media platforms like Facebook and Twitter, making moderating and identifying hate speech increasingly tricky. They developed a straightforward ensemble of transformer models for automatic hate speech detection to tackle this issue. Alshalan and Al-Khalifa [3] explored various neural network models, with a focus on Convolutional Neural Networks (CNN) and

TABLE 2. Example posts.

Comments	Hate or Not	Target
Aaiiii Jolly Yellam onnah polam onnah polam oannaaa polam	NOT	-
ippo marriage aga pothu la panitu muditu iru intha kelavi vayasula en asingama thitti vanvura	OFF	TI
yeiiiii lusuuuu kundathi ne kavin kuda pesalena avan enna sapdama erukka porana ne yen ippo bigg boss v2 kulla vantha saniyan enakkkaaaaa po muthevi	OFF	TI
dae gumbal birth thailis ... Gundan Stunt panlanu soldranunga Athiya proove panna vakku punda ila... Punagaimavanungaluku vaai maira paaru..	OFF	TG
yethukku ipdila pani reentry kudrukra nama la fool aakranga intha vj tv	OFF	OTHERS

TABLE 3. Dataset statistics.

Module	Training	Balanced Training	Testing
Hate Speech Detection	HS: 1980 NHS:2020	-	HS: 475 NHS:465
Total Posts	4000	-	940
Hate Speech Intent Detection	I:1759 TG:127 O:94	I:1759 TG:1759 O:1759	TI:441 TG:30 O:4
Total Posts	1980	5277	475

Recurrent Neural Networks (RNN), to identify hate speech in Arabic tweets. They also incorporated the bidirectional encoder representations from transformers (BERT) in their Arabic hate speech detection approach. Kapil and Ekbal [9] contributed by creating benchmark datasets for hate speech detection, deviating from standard annotation schemas. They proposed a deep multi-task learning (MTL) framework to extract sufficient information from the classification task, thereby enhancing the performance metrics of macro-F1 and weighted-F1.

Pereira-Kohatsu et al. [20] introduced HaterNet, a system employed for Hate Crime identification in Spanish, to identify and monitor hate speech on Twitter. They utilized a public dataset consisting of 6000 expert-labeled tweets to facilitate the detection of hate speech in Spanish. The study involved the application of various classification approaches, including text classification models such as the combination of LTSM and MLP neural networks. Mathew et al. [15] contributed to the field by creating HateXplain, a benchmark hate speech dataset that addresses many issues. Their dataset encompasses a 3-class classification scheme, categorizing content as hate, offensive, or regular, with specific target communities indicated. Madukwe et al. [12] elaborated on various pre-processing steps and suitable data formats to facilitate data availability in the public domain. They also discussed the manipulation of hate speech detection by comparing previous datasets and applying diverse approaches. Table 1 in their work summarises hate speech datasets available in the literature.

IV. DATASET DEVELOPMENT AND DESCRIPTION

As part of this work, we have curated a novel code-mixed Tamil dataset for hate speech detection and intent classification. This dataset was compiled by crawling 4000 code-mixed Tamil posts from Twitter and the Helo app. Among these posts, 1980 have been identified as hate speech, while the remaining 2020 posts fall under the non-hate category. To ensure accurate annotations, we engaged three annotators in the process. The first two annotators independently labelled the posts, and the third annotator resolved any discrepancies or conflicts in their annotations. To our knowledge, this represents the first-ever hate speech dataset tailored for the Tamil language. In addition, we organized a shared task known as HASOC during FIRE 2020 [13], utilizing the hate speech detection dataset, though not including the intent detection dataset.

Furthermore, we categorized the hate speech posts into subcategories, distinguishing between targeted and untargeted posts. We differentiate between targeted individuals and groups within the targeted category. Due to the relatively lower frequency of comments in the untargeted and other categories, we have consolidated our analysis into three primary classes: Targeted Individuals (TI), Targeted Groups (TG), and Others (O). An illustrative example of the dataset structure can be found in Table 2, while detailed dataset statistics are provided in Table 3. It is important to note that the hate speech intent detection data exhibits significant class imbalance, with a notable skew towards the category of targeted individuals. Table 3 delves into the number of

FIGURE 1. Word cloud for words.

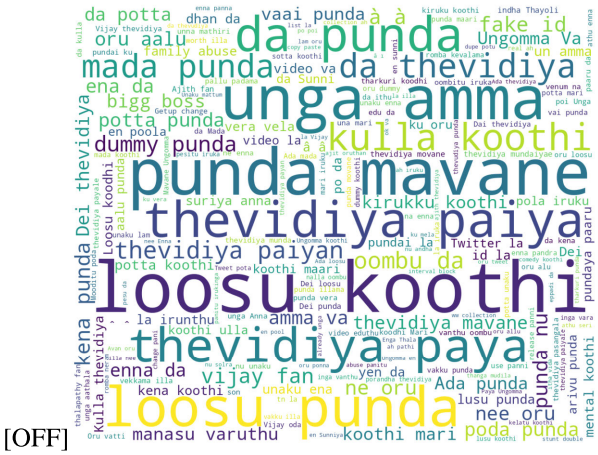

FIGURE 2. Word cloud for phrases.

posts utilized in our balanced training approach, which we elaborate on in Section V-D, where we discuss the balanced training procedure leveraging contextual embedding techniques.

Figure 1 provides an insightful visualization of word clouds for hate and non-hate content, utilizing unigrams and bigrams. It is evident from the figure that certain bigrams occur with notable frequency in hate speech posts. This bigram word cloud motivated us to explore the creation of phrase embeddings to enhance the effectiveness of learning and understanding comments more comprehensively and accurately.

V. METHODOLOGY

In recent years, hate speech has emerged as a pressing and widely recognized social issue, particularly prevalent in online media platforms. Hate speech instances within social media have the potential for more significant harm and danger. Consequently, there is an urgent demand for the practical identification of hate speech. In response to this need, we introduce a novel approach—a phrase vector-influenced hierarchical attention network. In this model, the attention layer plays a crucial role in discerning the contribution of each part of a tweet or post to its overall hateful content. This paper specifically investigates the impact of phrase and word representations in the context of a Hierarchical Attention Network for detecting hate speech in code-mixed Tamil. Our proposed attention-based model draws inspiration from how humans comprehend posts. Human readers tend to focus on specific words while considering context information from the entire post to determine whether it contains hate speech content. This approach aims to replicate and enhance this cognitive process within the framework of our computational model.

A. PHRASE EMBEDDING

Phrase embedding refers to learning phrase vectors from unsupervised text. In social media or short text, phrases play a vital role in understanding or classifying the post compared to words alone. The importance of phrases in hate speech

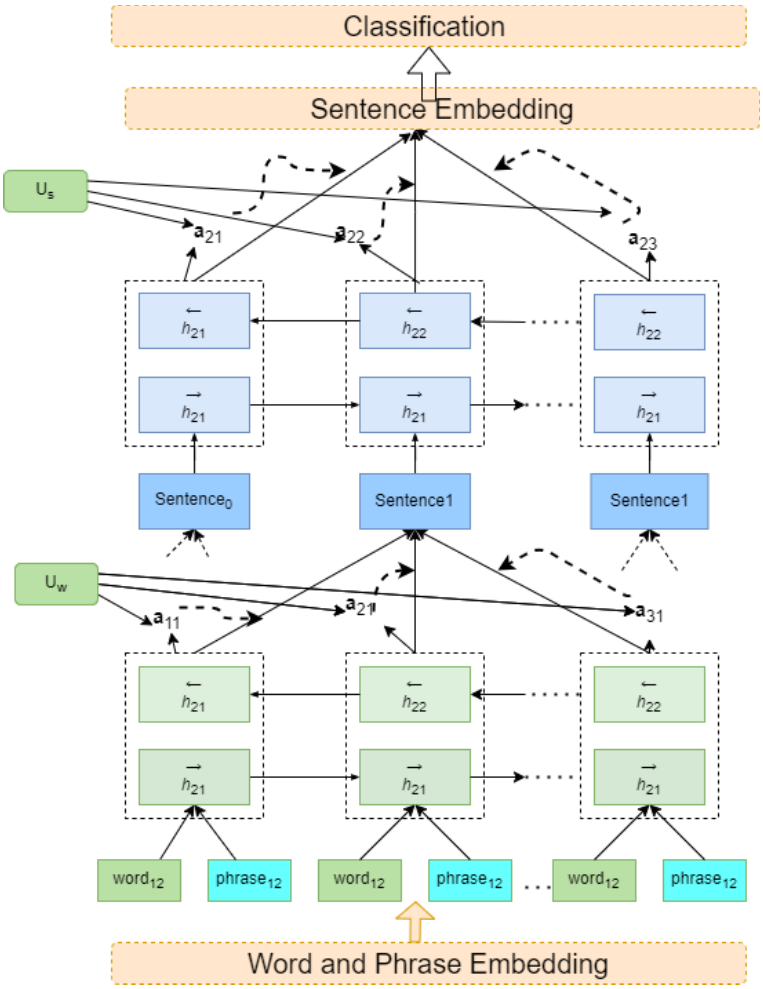

FIGURE 3. Frame work.

and non-hate speech posts is shown in Fig.1. The traditional way of converting the phrase embedding is to identify the correlated words (randomly connect them) and treat them as single words during the word2vec training.

Indian Government to Investigate Whatsapp alleged breach of Privacy

The bigrams can be identified converted as a unigram.

Indian_Government to Investigate_WhatsApp alleged_ breach of_Privacy.

The model discussed above faces several challenges when randomly connecting words to create phrase vectors, mainly when the words in a phrase are unrelated. This approach can produce phrase vectors that lack meaningful semantic content, thereby negatively impacting the overall quality of word vectors, especially when dealing with a vast vocabulary. We have employed a technique based on normalized Point-wise Mutual Information (PMI) to address this issue and ensure that only meaningful terms are associated. This method is designed to extract significant bi-grams from the unsupervised corpus. For instance, in the earlier example, ''Indian government'' is identified as a meaningful phrase and is appropriately transferred into a uni-gram representation, preserving its semantic relevance. This approach improves the overall quality of phrase and word vectors by focusing on meaningful combinations of words.

In the model described above, the unsupervised corpus transforms a phrase-annotated corpus, which serves as input for the Word2Vec model [16]. During training, phrases like ''Indian Government'' are treated as single words, and the model learns distributed vectors close to those associated with terms like ''Indian Government.'' To accomplish this, we utilized the popular Gensim library [23] to train phrase vectors for the code-mixed Tamil corpora, which were meticulously collected and cleaned. Words play a pivotal role in identifying offensive posts on social media platforms. Similarly, we aim to incorporate words and phrases into our hierarchical attention model and explore their combined impact on its performance. As part of our efforts, we have transformed a dataset comprising 90,000 code-mixed Tamil comments and posts into a corpus marked with phrases. This marked corpus was then utilized to train the Word2Vec-skip-gram model.

Algorithm 1 Phrase-Based Hierarchical Attention Network

Algorithm parameters: Posts post, Word Embedding W_e, $PhraseEmbeddingP_e$

foreach *post* **do**
  sentence = SentTokenize(post);
  foreach *sentence* **do**
    word = WordTokenize(sentence);
    foreach *word* **do**
      $w_{it} = W_e x_{it}$
      $p_{it} = P_e x_{it}$
      $z_{it} = [w_{it}, p_{it}]$
      $\overrightarrow{h_{it}} = \overrightarrow{GRU}(z_{it})$
      $\overleftarrow{h_{it}} = \overleftarrow{GRU}(z_{it})$
      $h_{it} = [\overrightarrow{h_{it}}, \overleftarrow{h_{it}}]$
      $u_{it} = tanh(W_w.h_{it} + b_w)$
      $\alpha_{it} = \frac{\exp({u_{it}}^T.u_w)}{\sum_{t=1}^{T}\exp({u_{it}}^T.u_w)}$
      $s_i = \sum_{t=1}^{T} d_{it}.h_{it}$
    end foreach
    $\overrightarrow{h_i} = \overrightarrow{GRU}(s_i)$
    $\overleftarrow{h_i} = \overleftarrow{GRU}(s_i)$
    $h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}]$
    $u_i = tanh(W_s.h_i + b_s)$
    $\alpha_i = \frac{\exp({u_i}^T.u_s)}{\sum_{l=1}^{L}\exp({u_{it}}^T.u_w)}$
  end foreach
end foreach

B. HIERARCHICAL ATTENTION MECHANISM

The attention network is a robust and well-established mechanism commonly employed in various architectural designs. In our approach, we have harnessed a unified model that combines hierarchical attention architecture [29] from natural language processing with phrase vector representations of textual data. The attention mechanism is pivotal in aiding the Bidirectional Long-Term Memory (Bi-LSTM) in determining which parts of the input text to ''attend'' to. It assists the model in learning which segments of the input text should be focused on and which have already been processed. This framework is visually depicted in Figure 2, clearly illustrating the overall architecture.

Not all words, phrases, or sentences within a social media post carry equal significance in semantically grasping the post's intention. To address this, we have introduced a Hierarchical attention mechanism that identifies the pivotal words and phrases crucial for capturing the offensive information conveyed in the post. These informative words and phrases are selectively aggregated to form a coherent and meaningful vector representation for each post. This approach allows us to distil the most salient content from the post, enabling more effective identification of offensive content.

Our approach initially inputs the embedded word and phrase vectors into a one-layer Multilayer Perceptron (MLP). This MLP generates representations in the hidden layer, which we then utilize to identify the informative words and phrases. This identification process is based on the similarity between context vectors. Subsequently, we calculate the vector representation for the given post, thereby creating a comprehensive and semantically meaningful representation. The phrase-based Hierarchical attention algorithm is concisely summarized in Algorithm 1, providing a step-by-step outline of the methodology employed.

C. FOCAL LOSS

We have employed the focal loss function in all our experiments based on the Hierarchical Attention Network (HAN). Focal loss functions were initially introduced by Facebook Artificial Intelligence research [11]. They are predominantly utilized in training processes for object recognition tasks designed to address the inherent class imbalance nature of the dataset. The focal loss function enhances the model's focus on misclassified instances belonging to the minority class by introducing a modulating term to the cross-entropy loss. In our proposed work, we conducted experiments using categorized and focal loss models to

Algorithm 2 Text Augmentation using Contextualized Embedding

Input: $Posts_{train}, N, LM, Labels : L = l_1, l_2..l_{|L|}$
Output: : $Posts_{aug\_train}$
$Posts_{aug} = []$, $Surr\_words = []$, $Aug\_word = []$

for $l = 1, 2, ..., |L|$ **do**
  $Posts_l \leftarrow D_{train} - \overline{Posts_l}$
  if $Posts_l << Posts_{l+1}$ **then**
    $Posts_{min} \leftarrow Posts_l$
    $N = size(Posts_{lmax}) - size(Posts_l)$
    for $n = 1, 2, .., N$ **do**
      for $n = 1, 2, \ldots,$N **do**
        for $w = w_1, w_2....w_n, w_0 \subset w \subset Rand(Posts_{min})$ **do**
          $Surr\_words \leftarrow (w_{-1}, w_{+1})$
          $Aug\_word \leftarrow LM(Surr\_words, Posts_{min})$
        end for
        $posts_n \leftarrow Insert(Aug\_word, Posts_{min})$
        $D_{aug} \leftarrow D_{aug} + posts_n$
      end for
      $Posts_{aug\_train} \leftarrow Posts_{train} + Posts_{aug}$
    end for

FIGURE 4. Contextualized embedding for text augmentation.

TABLE 4. Accuracy and F1 scores of first level hate speech identification.

Models	P	R	F1-Score	Accuracy
Word2Vec-CBoW_100+BiRNN_50 +ATtt_50	0.72	0.72	0.72	71.9
Word2Vec-CBoW_200+BiRNN_50 +ATtt_50	0.84	0.84	0.84	84.0
Word2Vec-CBoW_300+BiRNN_50 +ATtt_50	0.85	0.85	0.85	84.9
Word2Vec-SG_100+BiRNN_50 +ATtt_50	0.88	0.88	0.87	87.4
Word2Vec-SG_200+BiRNN_50 +ATtt_50	0.92	0.91	0.91	91.3
Word2Vec-SG_300+BiRNN_50 +ATtt_50	0.92	0.92	0.92	91.8

evaluate their respective performance and effectiveness. Mathematically, focal loss adds $(1-p_t)^r$ to the Cross-Entropy function.

$$FocalLoss(p_t) = -(1-p_t)^r log(p_t)$$

If the parameter (r) equals zero, it is the same as cross-entropy loss. Setting r greater than zero reduces the relative loss for accessible well-classified instances and more focus on misclassified cases.

D. CONTEXTUALISED EMBEDDING FOR TEXT AUGMENTATION

The proposed hate speech intent classification model encountered a highly imbalanced dataset, a common challenge in machine learning. We adopted a context embedding-based text augmentation technique to mitigate this imbalance and enhance the model's performance, as detailed in [7]. The algorithmic process is visually depicted in Figure 3. Here, the minority hate speech classes have been identified, and the data has been oversampled and balanced. The number of

TABLE 5. F1 scores and accuracy of hate speech detection model.

Models	Loss	Macro-F1	Weight-F1	Accuracy
Word+HAN+BiLSTM	CL	0.88	0.88	88.4
	FL	0.91	0.91	91.2
Word+HAN+BiGRU	CL	0.91	0.91	91.4
	FL	0.92	0.92	91.6
Word+HAN+BiLSTM_w+BiGRU_s	CL	0.90	0.90	90.3
	FL	0.91	0.91	91.1
Phrase+HAN+BiLSTM	CL	0.89	0.89	88.8
	FL	0.92	0.92	91.5
Phrase+HAN+BiGRU	CL	0.92	0.92	91.9
	FL	0.93	0.93	92.6
Phrase+HAN+BiLSTM_w+BiGRU_s	CL	0.89	0.89	88.7
	FL	0.92	0.92	92.0

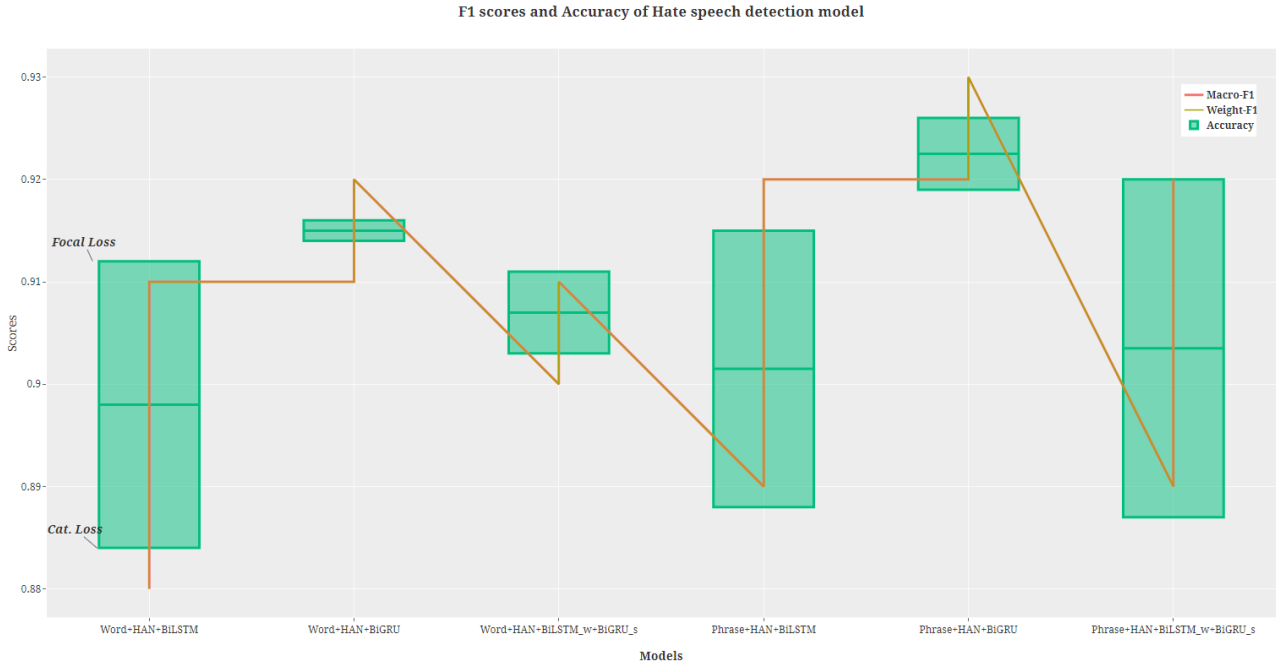

FIGURE 5. F1 scores and accuracy of hate speech detection.

TABLE 6. F1 scores and accuracy of data balanced hate speech type classification.

Models	Loss	Macro-F1	Weight-F1	Accuracy
Word+HAN+BiLSTM	CL	0.52	0.88	88.1
	FL	0.54	0.88	89.3
Word+HAN+BiGRU	CL	0.53	0.89	89.9
	FL	0.53	0.89	90.3
Word+HAN+BiLSTM_w+BiGRU_s	CL	0.55	0.89	89.3
	FL	0.53	0.89	90.1
Phrase+HAN+BiLSTM	CL	0.50	0.89	91.8
	FL	0.51	0.90	93.0
Phrase+HAN+BiGRU	CL	0.56	0.92	92.4
	FL	0.56	0.90	92.0
Phrase+HAN+BiLSTM_w+BiGRU_s	CL	0.50	0.89	90.9
	FL	0.51	0.90	92.4

class instances in the majority class instances is computed and considered as "M." To balance the minority class of each Hate speech intent (Targeted Group and Others), we used the parameter "M" as a reference variable. A sample post has been taken at random from the minority class and used to determine the words that surround each word. The current

word is created using the words nearby and instances of the minority class. Finally, we have produced equivalent samples for each randomly chosen instance. For minority classes, we generate an 'N' instance to balance the training dataset.

VI. EXPERIMENTAL RESULTS

This section describes the proposed hate speech detection system's experimental settings, evaluation metrics, parameters, and baseline systems.

A. EXPERIMENTATION SETUP

We have used the Tamil code-mixed Hate speech dataset we developed in this work. The dataset contains the post and its corresponding hate speech labels. Initially, each post is labelled as offensive or not. Then, if it is offensive, we categorize them again as targeted or untargeted. The reason for categorization is that the targeted abusive posts are more vulnerable than the untargeted ones, and the same should be reported or removed immediately. In the case of targeted posts, again, we are categorizing them to target individuals or groups and others.

B. PERFORMANCE MEASURES

To evaluate the performance of the proposed model for the Tamil author profiling task, we have used accuracy and F1 scores. The evaluation metric accuracy is defined as the proportion of documents that are classified correctly on the test set. The Macro-F1 measure has been calculated using the following Precision and Recall values.

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

$$Precision = \frac{TP}{TP + FP}$$

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

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

$$Macro - F1 = \frac{1}{|C|} \sum_{j=1}^{|C|} \frac{2P_j R_j}{P_j + R_j}$$

$$Weighted - F1 = \frac{\sum_{j=1}^{|C|} \frac{2|C_j| P_j R_j}{P_j + R_j}}{\sum_{j=1}^{|C|} |C_j|}$$

C. RESULTS AND ANALYSIS

This subsection describes the results of the hate speech detection module. As the dataset is balanced, we have not used contextualized embedding for oversampling. We have conducted several experiments by varying the word vector types and their vector size.

We have varied the word attention size, sentence attention size, and different sequential units to conduct the experiments and select the best parameters. A select few results from the initial level experiments are shown in Table 4 and Fig.4. Here, SG models perform better when compared to CBoW models. The vector sizes varied from 100, 200, and 300, and we observed that word vectors of size 300 gave better results. In the initial experiments, we used the RNN layer for the sequential modelling of words and sentences. Later, we found that when compared to LSTM and GRU, the RNN's performance is considerably lower.

Once we finalized the parameters for the Hierarchical Attention Networks, our experimentation extended to various sequential models, including both word and phrase embedding approaches. Additionally, we explored two different loss functions: cross-entropy loss and focal loss. The comprehensive results of these experiments, involving word embedding-based models and phrase embedding-based models for Hate speech detection, are presented in Table 5. In this table, we have provided metrics such as Macro-F1 score, Weighted F1, and accuracy, summarizing the outcomes of our conducted experiments. Notably, even though the Hate Speech detection dataset was balanced, focal loss models consistently outperformed those using categorical loss. The focal loss models exhibited improvements in both accuracy and Macro-F1 scores, ranging from 1% to 3%. Moreover, when comparing the word-based Hierarchical Attention Networks with our proposed Phrase-based Hierarchical Attention Networks, the proposed model showcased superior performance in accuracy and Macro-F1 scores. We explored various models throughout our experimentation, including Bidirectional LSTM (BiLSTM) and Bidirectional Gated Recurrent Unit (BiGRU). Among these models, Bidirectional GRU models demonstrated the best performance compared to BiLSTM. We also experimented with BiLSTM for generating sentence embeddings from words and phrases (BiLSTM_w), as well as GRU for creating overall embeddings from sentences (GRU_s) within social media posts. However, this combined model exhibited performance similar to the BiLSTM models. The most promising results were achieved using the Phrase-based BiGRU model trained with focal loss, which yielded a remarkable Macro-F1 score of 0.93 and an accuracy rate of 92.6%. These outcomes underscore the efficacy of our proposed model in the domain of Hate speech detection. Additionally, we compared our developed models with existing ones, and the comparative results are presented in Table 7 and Fig. 5.

We encountered a highly imbalanced dataset for Hate speech intent classification, which posed challenges in achieving the expected model performance even when using the Focal loss function. The model exhibited a strong bias towards the majority class, and the maximum Macro-F1 score achieved was only 0.50. To address this issue, we implemented contextualized embedding techniques to balance the training data. Despite including the Focal loss function, the initial imbalance in the data impacted performance.

TABLE 7. Comparison with existing systems.

Methods	Precision	Recall	Macro-F1
Proposed-Phrase+HAN	0.93	0.93	0.93
Proposed-Word+HAN	0.92	0.92	0.92
Transformers [24]	0.90	0.90	0.90
TFIDF+Char n-grams [5]	0.88	0.88	0.88
ULMFiT [4]	0.88	0.88	0.88
Ensemble+Char+Word n-gram [19]	0.87	0.87	0.87
XLM-RoBERTa [6]	0.87	0.87	0.87
Ensemble+BiLSTM [30]	0.88	0.87	0.87
TFIDF+Char n-gram [27]	0.86	0.86	0.86
Transliteration+mBERT [26]	0.86	0.86	0.86
SubWord Embedd+ BiLSTM [8]	0.85	0.85	0.85
Attention+BiLSTM+CNN [10]	0.84	0.84	0.84
CNN+BiLSTM [17]	0.84	0.83	0.83

FIGURE 6. Attention weights for the word "mental."

FIGURE 7. Attention weights for the word "o***i."

FIGURE 8. Attention weights for the word "entha."

FIGURE 9. Attention weights for the word "avar."

The results of our experiments with word embedding and phrase embedding models, incorporating contextualized embedding for Hate speech intent classification, are presented in Table 6. Notably, the phrase-based Bidirectional LSTM (BiLSTM) model achieved an accuracy rate of 93% when trained with focal loss. In contrast, the phrase-based Bidirectional Gated Recurrent Unit (BiGRU) model demonstrated the highest F1-Macro score of 0.56, considering both focal and categorical loss. The introduction of contextualized embedding led to significant improvements in F1 scores for the minority class, increasing them from 0.02 to 0.22. Similar to our Hate speech detection module findings, the proposed phrase-based model consistently outperformed the word embedding-based Hierarchical Attention Network (HAN) models. These results underscore the effectiveness of contextualized embedding in mitigating class imbalance and enhancing the performance of the models for Hate speech intent classification.

D. EXPLAINABLE ATTENTION WEIGHTS

This section delves into the interpretability of the model results through explainable attention weights. The figures presented provide insights into how specific words influence the model's decisions.

Fig.6 illustrates the significance of the word "mental" (an offensive term) in the model's determination of a post as hate speech. In Fig.7, the model assigns a confidence score of 93.23% to the offensive word, emphasizing its role in

identifying hate speech. Interestingly, Fig.8 demonstrates that the word ''entha'' (which translates to ''this'' in English) plays a crucial role in recognizing non-hate posts, despite not being a positive word.

Fig.9 highlights ''avar'' and ''dhana'' as significant indicators of non-hate speech comments. It is important to note that this interpretive section did not include machine-generated posts used in the hate speech intent classification. These figures show how the model's attention is directed towards specific words when making predictions, contributing to a better understanding of its decision-making process.

VII. CONCLUSION AND FUTURE SCOPE

The proliferation of social media has transformed the dynamics of message sharing. Social media users can now instantly disseminate posts to a vast global audience. Unfortunately, this ease of communication has also led to the widespread sharing of Hate speech content through social media platforms. Hate speech is typically characterized by the expression of intense hostility or aversion towards a specific individual or group based on factors such as religion, ethnicity, or gender orientation. The prevalence of hate speech and offensive content messages presents a significant research challenge, particularly within the natural language processing community. While several approaches have proven effective in identifying hate speech, they often need to explore the impact of phrase representations. In the context of this work, the code-mixed Tamil dataset used is characterized by a significant class imbalance. Consequently, we employed various metrics, such as the variance of the F1 measure, Macro F1 score, and Weighted F1 score, to compare and understand the distinct behaviours exhibited by the proposed method. The findings of this work suggest that the utilization of phrase vector representation significantly enhances the performance of the attention network. This model can potentially aid social network organizations in developing automatic Hatelexicons for different categories of hate speech. As part of future work, we plan to delve deeper into identifying various types of hate speech, including insults, harmful language, and profanity, with a particular focus on targeting groups such as women, religious communities, organizations, and more.

COMPETING INTERESTS

The authors declare that they have no competing interests.

AUTHORS CONTRIBUTION STATEMENT

V. Sharmila Devi: conceptualization, methodology, data creation and validation, implementation, experimentation, and writing—original draft.

S. Kannimuthu: conceptualization, methodology, writing, review editing, and supervision.

Anand Kumar Madasamy: conceptualization, data validation, writing, and review editing.

ETHICAL AND INFORMED CONSENT FOR DATA USED

Not Applicable.

DATA AVAILABILITY

The part of datasets used in this study are publicly available (https://competitions.codalab.org/competitions/25295), and the other dataset is available with the third author, which can be shared based on the request.

DECLARATIONS

The authors declare that they have no competing interests.

REFERENCES

[1] P. Alonso, R. Saini, and G. Kovács, ''Hate speech detection using transformer ensembles on the hasoc dataset,'' in *Proc. Int. Conf. Speech Comput.*, St. Petersburg, Russia. Cham, Switzerland: Springer, Oct. 2020, pp. 13–21.

[2] S. Alsafari, S. Sadaoui, and M. Mouhoub, ''Hate and offensive speech detection on Arabic social media,'' *Online Social Netw. Media*, vol. 19, Sep. 2020, Art. no. 100096.

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

[4] G. Arora, ''Gauravarora@hasoc-dravidian-codemix-fire2020: Pretraining ulmfit on synthetically generated code-mixed data for hate speech detection,'' in *Proc. FIRE, CEUR*, vol. 2826, 2020, pp. 362–369.

[5] N. N. Balaji and B. Bharathi, ''Ssncse-nlp@hasoc-dravidiancodemix-fire2020: Offensive language identification on multilingual code mixing text,'' in *Proc. FIRE, CEUR*, vol. 2826, 2020, pp. 370–376.

[6] A. Baruah, K. A. Das, F. A. Barbhuiya, and K. Dey, ''Iiitg-adbu@hasoc-dravidian-codemix-fire2020: Offensive content detection in code-mixed dravidian text,'' in *Proc. FIRE, CEUR*, vol. 2826, 2020, pp. 427–433.

[7] V. S. Devi and S. Kannimuthu, ''Author profiling in code-mixed WhatsApp messages using stacked convolution networks and contextualized embedding based text augmentation,'' *Neural Process. Lett.*, vol. 55, no. 1, pp. 589–614, Feb. 2023.

[8] K. Dong, ''Yun@hasoc-dravidian-codemix-fire2020: A multicomponent sentiment analysis model for offensive language identification,'' in *Proc. FIRE, CEUR*, vol. 2826, 2020, pp. 391–396.

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

[10] S. Kumar, A. Saumya, and J. P. Singh, ''Nitpainlp@hasoc-dravidian-codemix-fire2020: A machine learning approach to identify offensive languages from dravidian code-mixed text,'' in *Proc. FIRE, CEUR*, vol. 2826, 2020, pp. 384–390.

[11] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, ''Focal loss for dense object detection,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[12] K. J. Madukwe, X. Gao, and B. Xue, ''Dependency-based embedding for distinguishing between hate speech and offensive language,'' in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. Intell. Agent Technol. (WI-IAT)*, Dec. 2020, pp. 860–868.

[13] T. Mandl, S. Modha, M. A. Kumar, and B. R. Chakravarthi, ''Overview of the HASOC track at FIRE 2020: Hate speech and offensive language identification in Tamil, Malayalam, Hindi, English and German,'' in *Proc. Forum Inf. Retr. Eval.*, Dec. 2020, pp. 29–32.

[14] T. Mandl, S. Modha, P. Majumder, D. Patel, M. Dave, C. Mandlia, and A. Patel, ''Overview of the HASOC track at FIRE 2019: Hate speech and offensive content identification in indo-European languages,'' in *Proc. 11th Forum Inf. Retr. Eval.*, Dec. 2019, pp. 14–17.

[15] 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.

[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean, ''Efficient estimation of word representations in vector space,'' 2013, *arXiv:1301.3781*.

[17] A. P. Ajees, ''Ajees@hasoc-dravidian-codemix-fire2020,'' FIRE, CEUR, Working Notes, 2020.

[18] D. Paschalides, D. Stephanidis, A. Andreou, K. Orphanou, G. Pallis, M. D. Dikaiakos, and E. Markatos, "MANDOLA: A big-data processing and visualization platform for monitoring and detecting online hate speech," *ACM Trans. Internet Technol.*, vol. 20, no. 2, pp. 1–21, May 2020.
[19] V. Pathak, M. Joshi, P. Joshi, M. Mundada, and T. Joshi, "Kbcnmujal@hasoc-dravidian-codemix-fire2020: Using machine learning for detection of hate speech and offensive codemix social media text," in *Proc. FIRE, CEUR*, vol. 2826, 2020, pp. 351–361.
[20] J. C. Pereira-Kohatsu, L. Quijano-Sánchez, F. Liberatore, and M. Camacho-Collados, "Detecting and monitoring hate speech in Twitter," *Sensors*, vol. 19, no. 21, p. 4654, Oct. 2019.
[21] N. I. Pratiwi, I. Budi, and M. A. Jiwanggi, "Hate speech identification using the hate codes for Indonesian tweets," in *Proc. 2nd Int. Conf. Data Sci. Inf. Technol.*, Jul. 2019, pp. 128–133.
[22] S. Rajendran, M. A. Kumar, R. Rajalakshmi, V. Dhanalakshmi, P. Balasubramanian, and K. P. Soman, "Tamil NLP technologies: Challenges, state of the art, trends and future scope," in *Proc. Int. Conf. Speech Lang. Technol. Low-Resource Lang.* Cham, Switzerland: Springer, Nov. 2022, pp. 73–98.
[23] R. Řehůřek and P. Sojka, "Gensim-statistical semantics in Python," genism.Org., Nov. 2023.
[24] S. Sai and Y. Sharma, "Siva@hasoc-dravidian-codemixfire-2020: Multilingual offensive speech detection in code-mixed and romanized text," in *Proc. FIRE, CEUR*, vol. 2826, 2020, pp. 336–343.
[25] A. Saroj and S. Pal, "An Indian language social media collection for hate and offensive speech," in *Proc. Workshop Resour. Techn. User Author Profiling Abusive Lang.*, May 2020, pp. 2–8.
[26] P. Singh and P. Bhattacharyya, "Cfilt iit bombay@hasoc-dravidian codemix fire 2020: Assisting ensemble of transformers with random transliteration," in *Proc. FIRE, CEUR*, vol. 2826, 2020, pp. 411–416.
[27] P. V. Veena, P. Ramanan, and G. R. Devi, "Cenmates@hasoc-dravidian-codemix-fire2020: Offensive language identification on code-mixed social media comments," in *Proc. FIRE, CEUR*, vol. 2826, 2020, pp. 377–383.
[28] B. Wang and H. Ding, "YNU NLP at SemEval-2019 task 5: Attention and capsule ensemble for identifying hate speech," in *Proc. 13th Int. Workshop Semantic Eval.*, Jun. 2019, pp. 529–534.
[29] 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.*, Jun. 2016, pp. 1480–1489.
[30] Y. Zhu and X. Zhou, "Zyy1510@hasoc-dravidian-codemixfire2020: An ensemble model for offensive language identification," in *Proc. FIRE, CEUR*, vol. 2826, 2020, pp. 397–403.
[31] S. Modha, T. Mandl, P. Majumder, and D. Patel, "Tracking hate in social media: Evaluation, challenges and approaches," *SN Comput. Sci.*, vol. 1, pp. 1–16, 2020.

V. SHARMILA DEVI received the M.Tech. degree from Karunya University, Coimbatore, in 2013. She is currently doing her Ph.D. research with the Karpagam College of Engineering, Coimbatore. Her research interest includes analyzing social media text to identify hate speech in Indian languages. She has participated in various international NLP shared tasks. She received the First Place in the MAPonSMS Shared Task at FIRE2018 and Second Place in the APDA Shared Task at FIRE2019.

S. KANNIMUTHU (Senior Member, IEEE) received the B.Tech. degree in IT, the M.E. degree in CSE, and the Ph.D. degree in computer science and engineering from Anna University, Chennai. He is currently a Professor with the Department of Information Technology, Karpagam College of Engineering, Coimbatore, Tamil Nadu, India. He is also an In-Charge of the Center of Excellence in Algorithms. He is also an IBM Certified Cybersecurity Analyst. He has more than 16 years of teaching and industrial experience. He is also a recognized Supervisor with Anna University. Three Ph.D. candidates are completed their research under his guidance. He is also guiding nine Ph.D. research scholars. He has published 60 (SCI: 25 and Scopus: 39) research articles in various international journals. He has published two books *Artificial Intelligence* and *LinkedList Demystified: A Placement Perspective* and three book chapters (Scopus indexed). He is also acting as a Mentor/Consultant of DeepLearning.AI, MaxByte Technologies Dhanvi Info Tech, and Hubino. He is also an Expert Member of the AICTE Student Learning Assessment Project (ASLAP). He has visited more than 100 engineering colleges and delivered more than 160 guest lectures on various topics. He is a reviewer of 60 journals and three books. He has successfully completed the consultancy project through Industry-Institute Interaction for ZF Wind Power Antwerpen Ltd., Belgium. He has received funds from CSIR, DRDO, and ISRO to conduct workshops and seminars. His research interests include artificial intelligence, data structures and algorithms, machine learning, computer vision, big data analytics, blockchain, and virtual reality.

ANAND KUMAR MADASAMY (Member, IEEE) is currently an Associate Professor with the Department of Information Technology, National Institute of Technology Karnataka. He has over 185 research articles (Scopus indexed) to his credit, published in reputed international journals and conference proceedings. He has received around 2200 Google scholar citations for his research papers. His research interests include natural language processing, machine translation systems for Indian languages, machine learning, deep learning for natural language processing, and text analytics. He has completed the "Computing Tools for Tamil Language Learning and Teaching" Project funded by the Government of Tamil Nadu, the "Tamil-Malayalam Subtitle Translation System" Consultancy Project for Sharp Software Development Ltd., and "Identification and Extraction of Question and Answers, from UnStructured Documents" Consultancy Project with EduMinster (U.S). He is also a SERB-CRG Awardee (2023–2026) of the 42 lakhs cost project titled "A Deep Explainable Framework for Semantically Similar Document Retrieval and Summarization of Legal Text." He has organized more than nine international shared tasks in Indian languages named DPIL2016, MTIL2017, INLI2017, INLI2018, and HASOC22. He has developed NLP tools and resources (POS tagging, morphological analyzer and generator, and machine translation) for Tamil and other Dravidian languages. He received the P. K. Das Memorial South India's "Best Faculty Award" (CSE-Junior), in 2017, the Research Excellence Award from Amrita Vishwa Vidyapeetham, in January 2018, and the Young Scientist Award, in November 2016, in NLP.