IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

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

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

PDHS: Pattern-based Deep Hate Speech Detection with Improved Tweet Representation

P.Sharmila[1], Kalaiarasi Sonai Muthu Anbanathen[2], Deisy Chelliah[3], Sudhaman Parthasarathy [4], Subarmaniam Kannan[5]

[1,3,4]Thiagarajar College of Engineering, Madurai, Tamilnadu, India
[2,5]Faculty of Information Science and Technology, Multimedia University, Malaysia

Corresponding author: Kalaiarasi Sonai Muthu Anbananthen (e-mail: kalaiarasi@mmu.edu.my).

This work was partially supported by TMR&D grants (grant number MMUE/210038)

ABSTRACT Automatic hate speech identification in unstructured Twitter is significantly more difficult to analyze, posing a significant challenge. Existing models heavily depend on feature engineering, which increases the time complexity of detecting hate speech. This work aims to classify and detect hate speech using a linguistic pattern-based approach as pre-trained transformer language models. As a result, a novel Pattern-based Deep Hate Speech (PDHS) detection model was proposed to detect the presence of hate speech using a cross-attention encoder with a dual-level attention mechanism. Instead of concatenating the features, our model computes dot product attention for better representation by reducing the irrelevant features. The first level of Attention is extracting aspect terms using predefined parts-of-speech tagging. The second level of Attention is extracting the sentiment polarity to form a pattern. Our proposed model trains the extracted patterns with term frequency, parts-of-speech tag, and Sentiment Scores. The experimental results on Twitter Dataset can learn effective features to enhance the performance with minimum training time and attained 88%F1Score.

INDEX TERMS Attention Mechanism, Transformer, BERT, Sequence Modeling, Hate Speech, Natural Language Processing

I. INTRODUCTION

Twitter is the most popular and largest online social platform to express opinions. Tweets are short messages of 140 to 280 characters, and the content of tweets is mostly written in informal language. Also, the contents are noisy and unstructured [1], making them more difficult to analyze. The significant advantage of a brief text is that it conveys only one Sentiment about a subject.

The presence of hate speech, offensive, and cyberbullying language on tweets has increased, leading to severe problems [2]. Analyzing and detecting such informal tweets becomes a significant challenge. Hate speech detection is described as a classification task in [3]. Natural Language Processing (NLP) technique is employed to classify the tweet. Understanding the polysemous in tweet contents is the most difficult part of NLP. Furthermore, the monosemy nature of tweet contents makes hate speech classification and detection more complicated.

Any NLP task requires deep pre-training to reduce the complexity and computation of processing[4]. Neural language models and BERT[5] based approaches have been an increasingly powerful and successful approach to solving various NLP tasks [6]. For efficient computation, a model should be trained on a lower-dimensional dataset with less computational time.

Deep learning methods employ multiple layers of non-linear processing units to extract complex features that can be used to generate learning patterns and relationships beyond the sequence. The improvement of text representation methods, such as word and sentence encoders, has contributed significantly to the success of deep-learning approaches in NLP. Transformers with self-attention can be used to create rich representations.

A. TRANSFORMER - ATTENTION

An attention [5] approach is constructed to capture significant driving factors of encoder inputs in an adaptable manner to overcome the gradient vanishing problem of RNN. Attention is also used to focus on key features. The term attention is a weight used to compute the significance of an input element in order to boost performance.

The transformer model is entirely based on attention mechanisms without Recurrent Networks to address the long-term dependency issue. Hence Transformers are

preferable to RNNs because they enable parallel computing by taking the whole sequence as input. Also, Transformers does not require any labeled data. The most common practice of Transformer-based Attention is self-attention in the encoder part, where the input sequence pays Attention to itself, that is, the Query vector, Key vector and Value vector obtained from the same source, called self-attention. Recently Query vector, Key vector and Value vector obtained from different sources also in a different representation, called cross attention, is shown in Figure.1.

B.PATTERN FEATURES

Hate speech is detected using a variety of linguistic features. Parts-of-speech (POS) tags and aspect-based features are most commonly used, determined by nouns and noun phrases [7], to resolve the problem of polysemy and monosemy. Thus, aspect-based sentiment polarity and POStags are used as additional features as learning feature representations to the pattern.

Multi-head Cross Attention [5] is a variant of Attention used as an encoder in our proposed model for parallel processing of sequential modeling and data reduction with minimum computations.

As a result, this work proposes a novel hate speech detection as a ternary classification task to reduce the complexity and computation of NLP processing by reducing the dataset dimension. [8] inference from the stacked regression model, two-level attentions are implemented as a stacked model. The novelty of the proposed approach is to use dual-level cross attention on POS tagging and aspect-based sentiment polarity as the pattern-based on deep hate speech detection (PDHS). The proposed transformer-based cross-attention detection model would classify each tweet based on POS tagging features and sentiment polarity scores. The main contributions of this work are as follows

- Pre-processing and improved tweet representation
- Extract only adverbs, adjectives, verbs, and nouns using POS tagging linguistic features.
- Dual-level CrossAttention is applied with different feature embedding combinations as improved pattern representation.

This study is organized as follows: Section 2 discusses the related works, followed by the proposed PDHS methodology in section 3. Section 4 describes the dataset, results, and analysis of experiments and the conclusion is presented in section 5.

II. RELATED WORKS

Cyberbullying, hate speech, and inflammatory languages are examples of abuse, and various studies have been conducted to recognize and identify such languages. Detection of cyberbullying is subject to sentiment analysis. This is a predictive modeling task in which the model is trained to predict the polarity of textual data or expressions such as positive, neutral, and negative.

Three main sentiment categorization approaches are lexical-based, machine learning-based, and hybrid. The lexicon-based approach is powerful but requires an enriched linguistic dictionary. Furthermore, tweets are user-generated and may contain misspellings, typographical errors, and grammatical mistakes, making them unsuitable for expanding data. Also, this approach relies heavily on hand-crafted features, which are expensive to develop. The second approach is the machine learning approach. This approach depends on pre-labelled data. In general, this method is more accurate than the lexicon-based method. It does, however, require labelled data, which is not always readily available.

The hybrid approach integrates machine learning and the lexicons approach. The hybrid approach allows for a better comprehension of sentiment analysis, which leads to better categorization. Combining lexicon-based and machine learning methods showed significantly better results [18] but noisier on large data. Hence, the hybrid model is designed with deep neural networks for growing data. The different feature representations and algorithms are summarized in Table.1.

In this research, a deep neural network is proposed to extract features. CNN's provide automatic feature extraction, which is the primary advantage. Lee et al. [19] proposed a deep CNN model to extract aspects and transfer applied learning. However, it requires more number of layers for feature extraction. Instead of applying additional layers, it is preferable to use fewer engineering features to reduce computation risks. RNN is another popular deep learning algorithm used in sequence modeling. One of the main problems of RNN is the vanishing gradient problem.

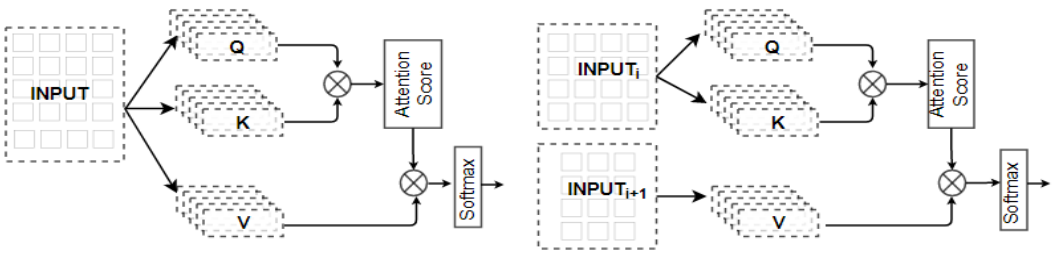


FIGURE 1. Overview of Self Attention and Cross Attention. In Self Attention, the input vector is projected into three vectors (Q, K, V). In Cross Attention, the vectors Q and K are from one input source and V from the next input source.

TABLE1. Different Feature Representation and Algorithms with F1 Score on Twitter Data

Author /Year	Classes	Feature Representation	Algorithm	F1Score	Inference
[5] (Davidson et al. , 2017)	Hate, Offensive, Normal	Transformer -BERT	CNN+Self Attention	0.89	Large Training Time
[9]Abozinadah et al. .(2015)	Abuse, Normal	Tweet features, BOW, Ngrams, TF-IDF	NaiveBayes	0.85	Complex Preprocessing
[10](Alakrotetal.,2020)	Offensive, Inoffensive	n-grams	SVM	0.88	Sparse One hot vector
[11](Han et al.,2020)	Adult, Regular User	Lexicon, N-grams, Bag of Means	SVM	0.78	Lack of sequence order
[12](Albadietal.,2018)	Hate, NotHate	Word Embeddings	GRU-RNN	0.77	Loss seems to be more
[13](Fernandez et al. l.,2018)	Online radicalisation	Semantic Context	SVM	0.85	High Computation cost
[14](MalmasiZampieri,2018)	Hate, Offensive, Normal	N-grams, Skipgrams, hierarchical word clusters	RBF kernel SVM	0.79	Slow Training
[15](Badjatiya et al. .,2019)	Sexist,Racist,Neither existnorracist	Random Embedding,	LSTM + GBDT	0.93	More Parameters , High Computation
[16](Duwairi et al. .,2021)	Hate, Offensive, Normal	Word-based frequency vectorization	RNN + LSTM	0.88	More Parameters , High Computation
[17](Zhang et al..,2018)	Racism,Sexism,Both, Non-hate	Word embeddings	CNN + GRU	0.94	Position, context and meaning not captured

To overcome this problem, memory-based RNNs such as LSTM GRU are preferred. These two networks can learn long-range dependencies between the input sequences. Here, the tweets serve as inputs are sequences of words.

Rudkowski et al. [20] showed the effectiveness of embedding words into mood analysis. Martini et al. [21] suggest a method for detecting ironic proposals on Twitter using an attention-based LSTM. Tadesse et al. [22] used a joint LSTM-CNN model to detect suicidal thoughts online. Basiotis [23] presents a collected model to track ironic tweets in sensitive RNNs at word and character levels.

Fortune et al. [24] suggested a summary of how the automatic detection of hate language in the text had made significant progress in NLP. Karamy et al. [25] suggested a systematic basis for text analysis of Twitter. Silva et al. [26] recommend a combination of word sets and linguistic features for filtering fairy news. One issue that arises while utilizing NLP is the polysomy issue. One of the methods to overcome this problem is by applying POS. Hence POS tagging and Aspect based feature is applied. Resignia et al. [27] proposed an improved model by combining POS, Lexicon, and word-to vector (W2V). But in our model, the dot product between POS tag and W2V is computed to extract task-related words.

Hate speech in social media networks is detectable but not recognizable based on user characteristics [31], such as gender. [28] proposed a deep CNN model to extract aspects and transfer applied learning. Sharma et al. [29] proposed W2V and CNN-based feature extraction methods on short text movie review corpus. García-Pablos et al. [30] presented W2VLDA to perform aspect and sentiment classification. Chen [31] proposes sentence level using BiLSTM-CRF and CNN-based sentiment analysis.

In our work, a good learning model to classify the emotions of the tweets will be developed. Hence, the proposed model works on deep neural lexicon-based approaches with different features as a pre-trained features compared with machine learning approaches and techniques for deep neural networks.

A.RESEARCH GAP

Some research gaps are identified from the above study on detecting hate speech on Twitter.

- Tweets are of word sequences; neither Feed-forward nor CNN are recommended. RNN is a chain-like structure used to represent tweets.
- No parallel processing and more computation are required for RNN, with the random initialization requiring more layers.
- Proper initialization and good representation as input improve the performance.

To improve performance, the bottlenecks addressed in our proposed work are enabling parallel processing for input sequences and proper initialization for good representation as input.

- BERT, a transformer-based context-dependent representation language model, is used as an encoder instead of Word2Vec, a context-independent embedding technique to enable parallel processing.
- Linguistic-based features such as POS and sentiment polarity are incorporated with the training model to improve the features representation

Hence, we propose a novel lexicon-based approach with different pre-trained features framed as the pattern by dual-level dot product cross attention for improved tweet representation and better computation performance of our model.

III PROPOSED PDHS DETECTION MODEL

The major aim of the proposed PDHS detection model is to classify each tweet into one of three classes based on POS tagging features and sentiment polarity scores using a cross-attention mechanism. The three classes are Hate-Speech: abusive or threatening speech against a particular group. Offensive: causing someone to feel resentful, upset or annoyed and Neutral.

The novelty of the proposed model employs a dual-level cross-attention pattern as features such as POS tagging, Sentiment Score, and Word embedding in order to improve the representation of tweets. Instead of concatenating the word Embedding with the POS feature, the dot product is computed to extract only adjectives, adverbs, verbs, and nouns. Extracting task-related words from tweet sentences reduces data sparsity.

The overall flow diagram of the proposed model is shown in Figure. 2. The proposed model consists of an input layer, a PDHS transformer block with eight encoders, and a classifier layer. Transformers are typically encoder-decoder structures, but the decoder layer is removed in this case, increasing the encoder to 8 layers. A single encoder block contains multi-attention heads. Cross attention replaces self-attention, which means that the input vectors Q(TWi), K(TWi), and V(TWi-1) to the encoder are not taken from a single word. Q(TWi) and K(TWi) represent word embedding and pos tagging, respectively, and V(TWi-1) represents the sentiment score of the previous word.

The proposed PDHS-Predefined Pattern Transformer model has a number of encoders. Instead of concatenating both features of each tweet word, it computes dot product attention between its corresponding word Embedding and its POS embedding. Then extracted tags of nouns, verbs, adverbs, and adjectives are fed to the next level of dot attention with sentiment scores. To check the negation, consider the predecessor of that particular tweet word. Each encoder contains a multi-head cross attention layer, normalization layer, feed-forward layer and finally, the output classifier layer.

A. INPUT LAYER

The input tweet sequence is preprocessed and tokenized. Then compute word embeddings and position embeddings for each token to maintain the sequential input order.

1) TOKENIZATION AND WORD EMBEDDING

Word embedding is a fundamental task that involves expressing each word as a set of numeric feature vectors. Figure. 3 illustrates tokenization as well as word embedding [32] or word IDs.The initial step is to vectorize all of the training data's tweets as a numeric vector and then vectorize each word to its corresponding POS vector [27].

2) POSITIONAL EMBEDDING

Positional embeddings are there to give a transformer knowledge about the position of the input vector shown in Figure.4. They are added to corresponding input vectors. Position embeddings (PE as depicted in Figure.4) are calculated using Eqn.1 and 2.

$$P(k,2i) = sin\left(\frac{k}{n^{2i/d}}\right) \quad (1)$$

$$P(k,2i+1) = cos\left(\frac{k}{n^{2i/d}}\right) \quad (2)$$

where k denotes the position of a term in the sequence, $0 \leq k < L/2$. d is the dimension or size of the output embedding. The scalar value n is usually set to 10000 as default. The position function P(k, i) is used to map a position k in the input sequence, and i is the mapping column indices $0 \leq i < d/2$. The value sin and cos are used to normalize the vector between -1 and 1.

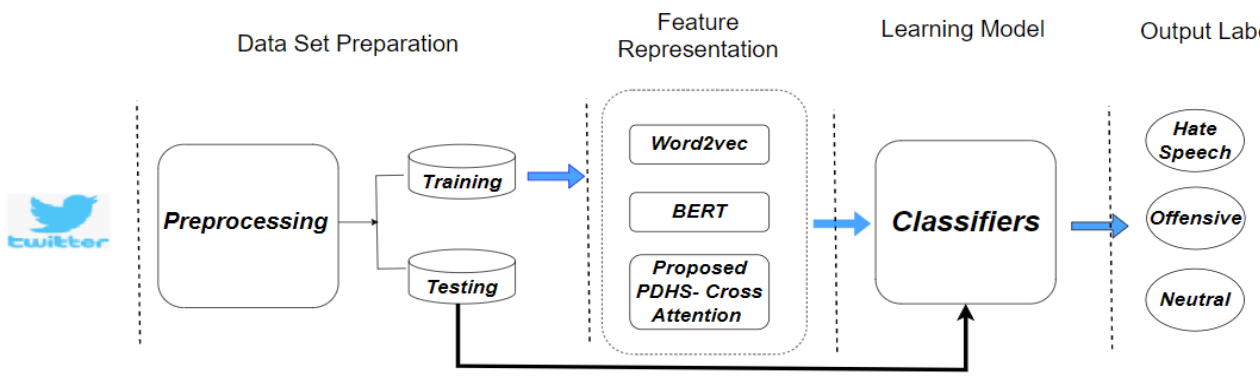


FIGURE 2. Overall Flow Diagram of the Proposed Model

3) PDHS CROSS ATTENTION LAYER

As a first-level attention mechanism, words with the POS tag as a verb, adjective, or adverb are extracted as sentiment words and dot products with the corresponding word Embedding. As a result, the sparsity of word features is reduced by using dot product attention representation.

$$Mulithead\ (Q_i\ , K_i\ , V_{i-1}) = concat(\mathrm{hd}_1, \mathrm{hd}_2 \ldots \mathrm{hd}_n) \quad (3)$$

$$hd_i = Attention\ (Q_i\ , K_i\ , V_{i-1}) \quad (4)$$

$$Attention\ (Q_i\ , K_i\ , V_{i-1}) \ = FA\ .SA \quad (5)$$

$$FA = \ softmax(Q_i * \ , K_i\ ) \quad (6)$$

$$SA \ = \ sigmoid[\ FA * V_{i-1}] \quad (7)$$

where Q_i *and* K_i are the query vector (word embedding) and key vector (POS embedding) of i[th] word. V_{i-1} is the value vector (sentiment score) of previous word of i. FA and SA are first and second-level Attention, respectively. hd_i is the attention head score of the i[th] word

Now cross dot product is computed between extracted words and the predecessor of that corresponding word, this time with a sentiment score of the corresponding words and predecessor of that word. A sentiment score is assigned to each extracted word [33]. If the word has a positive sentiment score, then look at the predecessor word's sentiment score for negation. If so, the negative sentiment score count increases by one; else, the positive score count increases by one. Similarly, look at the predecessor word sentiment score for negation for negative words. If so, the positive sentiment score count increases by one; else, the negative sentiment score count increases by one. The Final Score is calculated by adding the tweet's summation scores for each extracted word. Based on this Score, each tweet is categorized as either negative (hateful), positive (Offensive), or neutral (neither).

The individual attention scores are fed into the normalization layer and then the feed-forward layer, where the inputs are again fed to the next encoder. The final encoder is the output embedding.

4) CLASSIFIER LAYER

The output embedding is then fed into the softmax classifier layer, determining whether the target label is hate speech, offensive speech, or neither.

Our proposed model incorporates POS tagging and Sentiment Score as Feature Engineering. In contrast to the existing model, which concatenates POS features with word embedding, the novelty of our proposed model computes dot product attention as first-level Attention to extract specific POS tags. Then, as illustrated in Figure 5, these extracted POS features are then dotted product with sentiment scores of that word and previous word to create second-level Attention.

To improve the performance, we must identify the optimal combination of features as patterns. The proposed

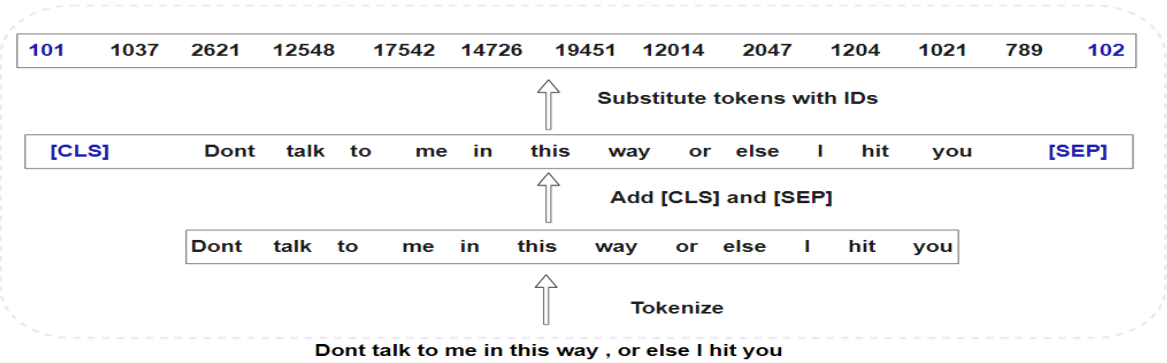


FIGURE 3. Tokenization and Word Embedding

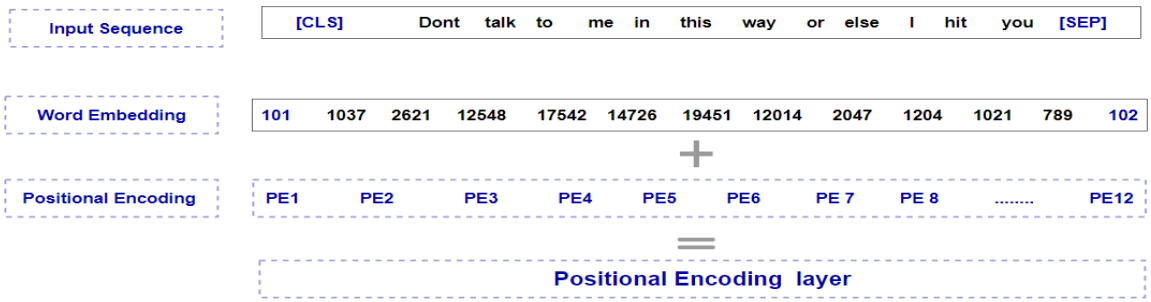


FIGURE 4. Positional Encoding

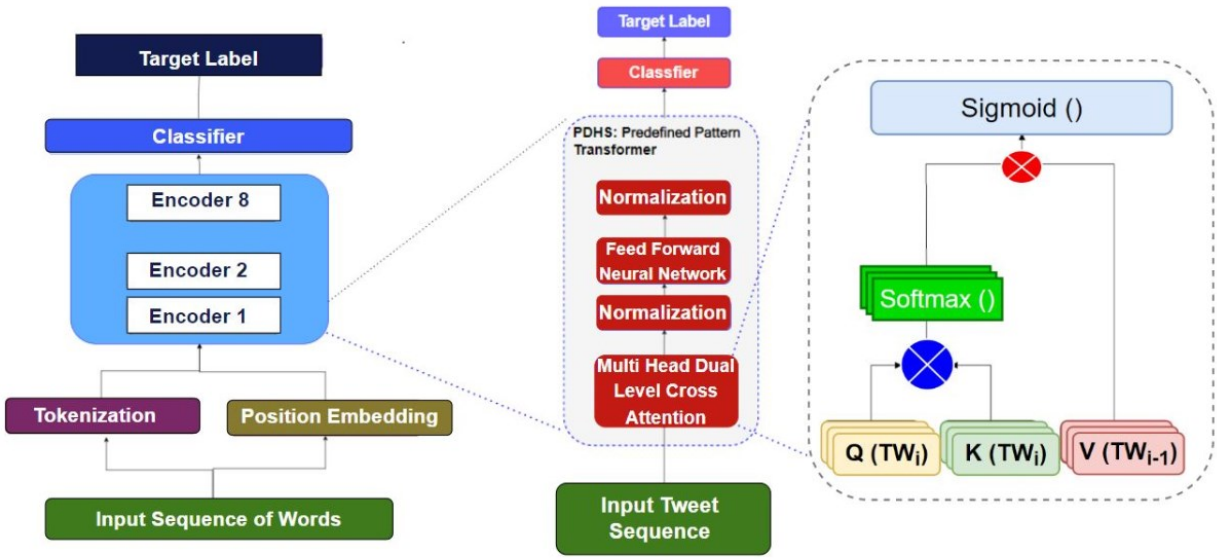


FIGURE 5. The Architecture of the Proposed Dual-level Cross Attention for predefined Patterns.

algorithm generates patterns from sentiment words and predecessors of sentiment words, as described in Algorithm 1. Adjectives, adverbs and verbs are extracted using POS tags, and the extracted T_{word} check for negation, and the count increase or decreases by one depending on the conditions:

if the T_{word} sentiment Score is negative, then look at the predecessor word
if it is also negative, then the T_{word} increase by one
else T_{word} decrease by one.
Repeat the above step as if the
T_{word}sentimentscoreispositive, then look at the predecessor word
If it is also positive, the Tword increases by one;
else
T_{word} decreases by one.
Finally, the Tweet Score is computed by
$T_{tScore} = T_{tScore} + T_{word}$.

Every word in the sequence gets an integer sequence as a one-hot vector and a corresponding POS tag, dot products to form the word Embedding. It is then passed to the next level of Attention with a Sentiment score.

POS tagging is important to identify nouns, verbs, adjectives, adverbs, etc. This will be used to avoid ambiguity or polysomy problems. Sentiment score captures the sentimentword from the tweets as 'positive', 'negative', and 'neither. Semantic features concentrate on the meanings of words ina sentence. Hate is mostly an emotional word, expressing a negative emotion. As a result, we feel that relying on the tweet's emotional polarity is a significant signal of whether or not the tweet is potentially hateful.

So with the support of these POS and Sentiment scores in our proposed PDHS model, extract the limited tags: nouns as aspect terms and verbs, adjectives, and adverbs as sentiment words.

C. PROPOSED MODEL TRAINING

Word-embedding with positional encoding is used as an embedding layer in our model, which helps to improve deep learning performance. The machine-learning and deep-learning model generates a multiclass output that differentiates between different types of hate speech, offensive speech, and neither. To start, our model requires 12 to 16 GB of GPU memory and trained with 6 to10 epochs, the learning rate of 0.01 to 0.5, Adam optimizer.

Hyperparameters used in our proposed model to determine the data dimension depend on:
L - Number of Transformer encoder blocks.
H - Hidden size - the size of *Q, K* and *V* vectors.
A - Number of multi-attention heads.

Our proposed model employs eight encoder blocks, a hidden size of 215, and twelve multi-attention heads. Tweets are about 140 to 280; hence the hidden size of Embedding is varied from 150 to 200.

D. EVALUATION METRICS

The final step of the text classification and detection task is evaluation, which is used to estimate the performance of our proposed model. Standard metrics derived from a confusion matrix, such as precision, recall, and F measures, are used to evaluate our proposed tweet classification model. The measures of precision, recall, and F1Score is shown in equation (8) to (11). True positives (TPs) are predictions that an actual positive will be positive. True negatives (TNs) are predictions that an actual negative will be negative. False

Algorithm 1: PDHS Algorithm to detect hate speech

INPUT: Tweet T_t // sequence of words

OUTPUT: classified as Hate, Neutral, Offensive

```
T_t Score = 0            // Initial score as 0
For each T_t do          // for all words in each tweet
Tword = 0
For each word in the Tweet do
// to extract limited tag words
Extract tags: Adjective, Adverb, and Verb as Ext-tag
        If Ext-tag is Negative
{
If the predecessor of Ext-tag is positive
        {Tword→ +1}
Else
{Tword→ -1}
        Else
If the predecessor of Ext-tag is negative
{Tword→ -1}
Else
{Tword→ +1}
T_t Score = T_t Score + Tword
End For
If T_tScore < 0
Twitter is Hate/Non-Offensive
Else if T_tScore = 0
Twitter is Neutral / Neither
Else T_tScore>0
Twitter is Offensive
```

positives (FPs) are misinterpreting a true positive as a negative. False negatives (FNs) are the misinterpretations of a true negative as a positive.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

$$Precision = \frac{TP}{TP+FP} \quad (9)$$

$$Recall = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

$$F1\ Score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (11)$$

IV. EXPERIMENTANDIMPLEMENTATION

A.DATASET

We collect the publicly available data of 2484 tweets manually classified as 'Hateful,' 'Offensive', and 'Neither'. Also, split 80% as training and 20% as testing sets. Link-https://www.kaggle.com/datasets?search=labeled_data.csv

B. PREPROCESSING

TABLE2. Preprocessing Steps

Preprocessing steps	Description
Lowercase Conversion	Convert all characters into lowercase
Stop-word removal	Words from the stop-word list (NLTK) are removed,
Punctuation Marks removal	Punctuations (! () [] {} ; : ' • " \ , <> . ? $ % & ^ * £ _ ~ +) are removed from
URL and User Mentions replacement	Replace URL links and @USER with 'URL' and '@username.'
Date-Time Replacement	Replace or change the date and time format ('ddmmyyyy' and 'hhmmss')
Number replacement	Replace numbers as 'num'
Hashtag replacement	Replace hashtags (#) should replace with 'hashtag'
Emoticon removal	Emojis from tweets are removed.
Stripping whitespaces	Additional whitespaces are removed.

Tweets are highly unstructured. So some preprocesses are applied to form well-structured data for further analysis. [33] and [34] proposed the preprocessing techniques for Twitter data before sentiment analysis. Table.2 details the procedures involved in preprocessing. The preprocessing steps in this hate speech detection are case conversion, removal of stopwords, URL, punctuation symbols, special characters, and replacement of date, time and hashtag. Finally, stripping of white space and tokenization. Tweets and preprocessed tweets are given in Figure.6.

```
                                                tweet  \
0  !!! RT @mayasolovely: As a woman you shouldn't...
1  !!!!! RT @mleew17: boy dats cold...tyga dwn ba...
2  !!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...
3  !!!!!!!!! RT @C_G_Anderson: @viva_based she lo...
4  !!!!!!!!!!!!! RT @ShenikaRoberts: The shit you...
5  !!!!!!!!!!!!!!!!!!"@T_Madison_x: The shit just...
6  !!!!!!"@__BrighterDays: I can not just sit up ...
7  !!!!&#8220;@selfiequeenbri: cause I'm tired of...
8  " &amp; you might not get ya bitch back &amp; ...
9  " @rhythmixx_ :hobbies include: fighting Maria...

                                        processed_tweets
0  woman complain clean hous amp man alway take t...
1  boy dat cold tyga dwn bad cuffin dat hoe st place
2          dawg ever fuck bitch start cri confus shit
3                                  look like tranni
4      shit hear might true might faker bitch told ya
5       shit blow claim faith somebodi still fuck hoe
6                sit hate anoth bitch got much shit go
7             caus tire big bitch come us skinni girl
8                amp might get ya bitch back amp that
9                     hobbi includ fight mariam bitch
```

FIGURE 6.Tweet Preprocessing

TABLE3. Comparison of Classification Machine Learning Algorithms

Algorithm	Testing					Training				
	Accuracy	Precision	Recall	F1 Score	Prediction Time	Accuracy:	Precision:	Recall	F1 Score	Training Time
Bagging Classifier	0.929585	0.966584	0.92483	0.945246	0.219115	0.989009	0.996851	0.986403	0.9916	29.851784
SGD Classifier	0.928021	0.96124	0.927891	0.944271	0.00301	0.983383	0.992049	0.982607	0.987306	0.122096
Logistic Regression	0.926344	0.964089	0.922279	0.942721	0	0.978726	0.990241	0.977282	0.983719	1.125309
DecisionTree Classifier	0.923662	0.953728	0.928912	0.941156	0.044081	0.998845	0.999943	0.9983	0.999121	3.842501
Linear SVC	0.916732	0.946599	0.92551	0.935936	0.002543	0.997019	0.998298	0.997167	0.997733	0.410218
RandomForest Classifier	0.91662	0.948619	0.923129	0.935701	2.75743	0.998845	0.99949	0.998754	0.999122	40.718343
AdaBoost Classifier	0.907567	0.972508	0.884354	0.926338	0.254679	0.90965	0.971744	0.888448	0.928231	1.271948
Multinomial NB	0.893372	0.901663	0.940306	0.920579	0.057981	0.94456	0.956763	0.959039	0.9579	0.036372
KNeighbors Classifier	0.857606	0.895161	0.887245	0.891186	34.074224	0.897727	0.927596	0.915982	0.921753	0.004012

C.FEATUREEXTRACTION RESULTS

First, generate the POS tag for all words in tweets; Tword with POS tag nouns, verbs, adjectives, and adverbs are extracted from Tweet. Then, compute dot attention with sentiment scores. Figure.7 shows the proposed model's POS tag extraction and sentiment score. The result of the first Attention again dots product with sentiment scores of previous words to classify the tweets.

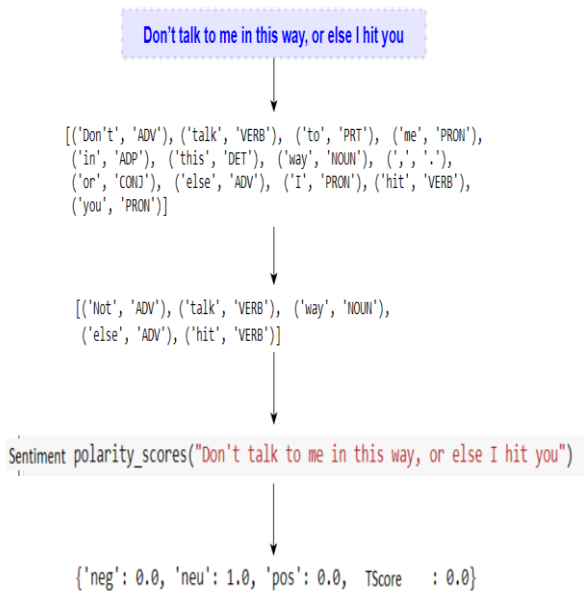


FIGURE 7.Feature Extraction using our proposed PDHS model

V. RESULT AND OBSERVATION

The proposed model is evaluated against hand-crafted features using TF-IDF and W2V neural word Embedding. The proposed pattern-based PDHS features are more suitable for keyword identification and perform better. The result of different machine learning algorithms on test and train data is shown in Table.3.

Logistic regression [28] is mainly used for binomial classification, and the resultant confusion matrix is well-explained; it limits the size of the data set.CNN with only word2vec showed a moderate result. Support Vector Machine (SVM) is inefficient with large data. The scale of datasets has a big impact [35], and it is particularly relevant in the case of Twitter, which is constantly growing in data. Deep learning methods show their true potential.

TABLE 4. Accuracy and Training Loss for DNN models

Models	Accuracy	Training Loss
RNN	0.9905	0.0310
LSTM	0.9916	0.0256
GRU	0.9913	0.0261
BERT	0.9911	0.0259
PDHSmodel	**0.9915**	**0.0232**

Hence, deep learning methods are highly suggested. CNN only with word2vec (without any additional features) have comparably fewer true positive values. RNN with word2vec with POS tag and sentiment scores showed better results than CNN. The proposed PDHS model, with a predefined pattern feature, has a better accuracy value, as shown in Table.4.

A. COMPARATIVELY ANALYSIS OF PATTERN

The proposed [35] pattern-related features based on POS tagging have an F1 score of 70%. However, in our model, the sentiment score is included in addition to POS tagging and has an F1 score of 88%. The Contrast-based classifier pattern [36] is created by combining relational statements

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3210177

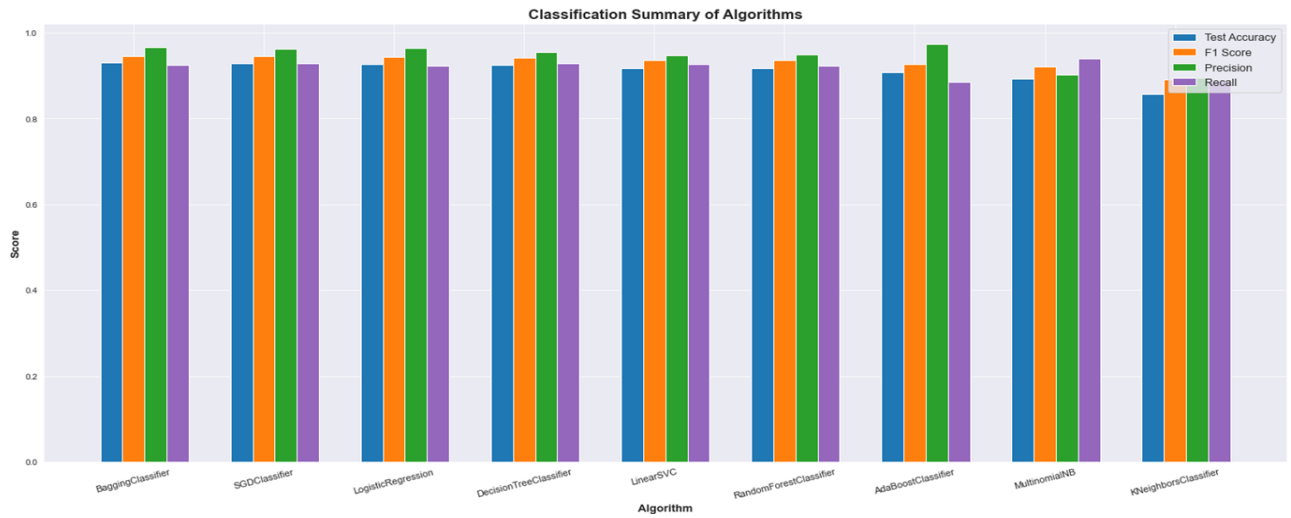


FIGURE 8. Comparison of Classification Machine Algorithms

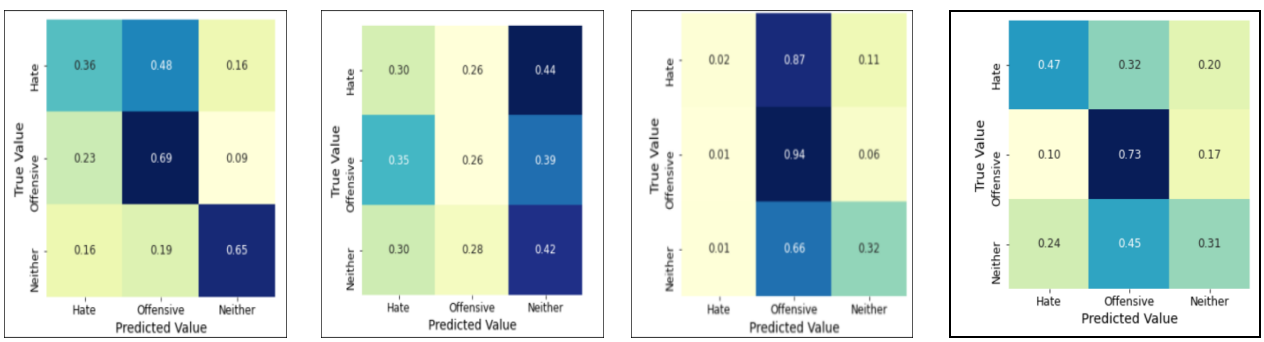
with Random Forest (RF) classifier. But RF and ensemble BFS-RF [38] classifier requires high training time. Hence in our model, we employ a softmax classifier to categorize the tweets as hate speech, offensive or not. Figure.8 depicts a graph comparing classification on several machine learning algorithms.

B. COMPARATIVELY ANALYSIS OF BERT CLASSIFIER

Self-attention is used in BERT [5], but our model incorporates cross attention with additional features as a pattern. The first dot product attention reduces data size by extracting only Noun, Verb, Adjective, and Adverb POS tagging words. Cross dot product with extracted POS tag terms with its corresponding previous term is the second level of Attention. This allows it to identify the keywords easily.

For parallel processing, multi-head Attention is used. Positional encoding is used to solve the Sequence order problem. As a result, our proposed PDHS dual-level cross-attention model was used to process the entire tweet embedding along with position encoding. Softmax classifiers are also used for multiclass categorization. Figure.9 shows normalized confusion matrices for ternary classifier models based on the different feature TF-IDF scores, sentiment scores, word2vec and our proposed PDHS model.

As shown in Table.4, there is no significant difference in the accuracy when using Lstm, GRU, or BiLstm, the most popular RNNs. So Transformer is preferred for better accuracy with minimum training loss. Also, the proposed PDHS model, with a Predefined pattern feature, has a better accuracy value. Table.5 shows the comparison of results for different combinations of features. Hate speech detection with our proposed model has improved the F1 Score while maintaining comparable Precision and Recall values.



SVM – TF-IDF with POS tag and Sentiment Score

CNN - Word2vec without any additional features

RNN - Word2vec with POS tag and Sentiment Score

Proposed PDHS model with predefined pattern features

FIGURE 9. Concatenation of TF-IDF scores, Sentiment Scores, Word2Vec, and Doc2Vec

TABLE 5. Result comparison of proposed PDHS model (Precision, Recall, F1-Score)

Category	Precision	Recall	F1-Score
Only Tf- IDF			
Hate -Speech	0.56	0.59	0.58
Offensive-Speech	0.52	0.42	0.42
Neither	0.59	0.53	0.54
Overall	0.56	0.52	0.51
Tf-IDF with POS tag + Sentiment-Score			
Hate-Speech	0.63	0.073	0.33
Offensive-Speech	0.65	0.05	0.40
Neither	0.92	0.27	0.42
Overall	0.73	0.13	0.38
RNN Word2vec with POS tag with Sentiment-Score			
Hate-Speech	0.328	0.114	0.590
Offensive-Speech	0.721	0.386	0.872
Neither	0.845	0184	0523
Overall	0.631	0.184	0.661
Proposed PDHS Predefined Pattern feature model			
Hate	**0.799**	**0.101**	**0.87**
Offensive	**0.863**	**0.046**	**0.88**
Neither	**0.891**	**0.179**	**0.89**
Overall	**0.874**	**0.105**	**0.88**

TABLE 6. Test Accuracy-5 epochs and10 epochs withTrainingTimefor DNN models

Models	TestAccuracy (5Epochs)	TestAccuracy (10Epochs)	TimetoTrainEpoch (seconds)
CNN	72.88	75.560	18.040
Lstm	81.40	80.08	20.237
GRU	75.960	78.960	18.935
BiLstm	78.840	81.120	30.113
BERT	80.121	81.005	29.501
Proposed PDHS	**81.880**	**81.920**	**28.475**

Table.6 compares the proposed PDHS model's training time and accuracy with all other models. The GRU and CNN models took less time to train per epoch, but the model requires more epochs to obtain maximal performance. So, our proposed model outperforms the other models.

V. CONCLUSION and FUTURE WORK

The most challenging task of automated hate speech detection on Twitter was addressed in this paper by a novel Pattern-based Hate Speech detection model with improved representation. In our proposed model, First level dot attention extracts the words related to corresponding POS tags by avoiding the preprocessing step such as stop words removal and stemming. Second level attention with sentiment score as additional input to improve the performance training. Thus, the experimental results on the Twitter data set outperformed supervised machine learning algorithms in terms F1-Scores by 88% and with the minimum training time.

In the future, we will improve the proposed model for unstructured social media datasets by incorporating multi-modality features like images and emojis.

ACKNOWLEDGMENTS

This work was partially supported by TMRD grants (grant number: MMUE/210038).

REFERENCES

[1] U.Naseem, I.Razzak, and P.W. Eklund, "A survey of preprocessing techniques to improve short-text quality: a case study on hate speech detection on Twitter", Multimedia Tools and Applications, vol.80, pp.35239–35266, 2021. https://doi.org/10.1007/s11042-020-10082-6

[2] S.Sadiq, A.Mehmood, S.Ullah, M.Ahmad, G.S.Choi, and B.-W. On, "Aggression detection through deep neural model on Twitter," FutureGenerationComputerSystems,vol.114,pp.120–129,2021.

[3] A.Tripathy, A.Anand, and S.K.Rath, "Document-level sentiment classification using hybrid machine learning approach," Knowledge and Information Systems, vol.53, no. 3,pp.805–831,2017.

[4] S.Minaee, N.Kalchbrenner, E.Cambria, N.Nikzad, M.Chenaghlu, and J.Gao, "Deep learning--based text classification: a comprehensive review". ACM Computing Surveys (CSUR), vol.*54*(3), pp.1-40, 2021.

[5] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In NAACL-HLT, 2019

[6] Q. Xu, L. Zhu, T. Dai, and C. Yan, "Aspect-based sentiment classification with multi-attention network," Neurocomputing, vol. 388, pp. 135–143,2020.

[7] J.Zeng, X.Ma, and Zhou, "Enhancing attention-based LSTM with position context for aspect-level sentiment classification," IEEE Access, vol. 7, pp. 20462–20471,2019.

[8] Anbananthen KS.M, Subbiah S, Chelliah D *et al.* An intelligent decision support system for crop yield prediction using hybrid machine learning algorithms.*F1000Research* 2021, **10**:1143 (https://doi.org/10.12688/f1000research.73009.1)

[9] E. A. Abozinadah, A. V. Mbaziira, and J. Jones, Detection of abusive accounts with Arabic tweets, "Int.J.Knowl.Eng.-IACSIT,vol.1,no.2, pp. 113–119, 2015.

[10] A. Alakrot, L. Murray, and N. S. Nikolov, "Towards accurate detection of offensive language in online communication in Arabic, "Procediacomputerscience, vol. 142, pp.315–320, 2018.

[11] K.X. Han, W. Chien, C.C. Chiu, and Y.T. Cheng, "Application of support vector machine (SVM) in the sentiment analysis of twitter dataset", *Applied Sciences*, Vol.10, no.3, pp.1125, 2020.

[12] N. Albadi, M. Kurdi, and S. Mishra, "Are they our brothers? analysis and detection of religious hate speech in the Arabic Twittersphere," in 2018IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining(ASONAM).IEEE, pp.69–76, 2018.

[13] M. Fernandez, M. Asif, and H. Alani, "Understanding the roots of radicalization on Twitter," in Proceedings of the 10th ACM conference on web science, pp. 1–10, 2018.

[14] S.Malmasi and M.Zampieri, "Challenges in discriminating profanity from hate speech," Journal of Experimental & Theoretical Artificial Intelligence, vol. 30, no. 2, pp. 187–202, 2018.

[15] P. Badjatiya, M. Gupta, and V. Varma, "Stereotypical bias removal for hate speech detection task using knowledge-based generalizations," in The World Wide Web Conference, pp. 49–59, 2019.

[16] R.Duwairi, A.Hayajneh, and M.Quwaider, "A deep learning framework for automatic detection of hate speech embedded in Arabic tweets," Arabian Journal for Science and Engineering, no. 46(4), pp.4001-4014, 2021.

[17] Z. Zhang, D. Robinson, and J. Tepper, "Detecting hate speech on Twitter using a convolution-gru based deep neural network, "in European

semantic web conference. Springer, 2018, pp. 745–760.
[18] S. M. Vohra and J. B. Teraiya, "A Comparative Study of Sentiment Analysis Techniques," J. Inf. Knowl. Res. Comput. Eng., vol., 2, no. 2, pp. 313–317, 2012.
[19] Y. Lee, M. Chung, S. Cho, and J. Choi, "Extraction of product evaluation factors with a convolutional neural network and transfer learning," Neural Processing Letters, vol. 50, no. 1, pp. 149–164, 2019.
[20] E.Rudkowsky, M.Haselmayer, M.Wastian, M.Jenny, Š.Emrich, and M. Sedlmair, "More than bags of words: Sentiment analysis with word embeddings, "Communication Methods and Measures, vol.12, no.2-3, pp. 140–157, 2018.
[21] A. T. Martini, M. Farrukh, and H. Ge, "Recognition of ironic sentences in twitter using attention-based LSTM," Int. J. Adv. Comput. Sci. Appl, vol. 9, no. 8, pp. 7–11, 2018.
[22] M.M.Tadesse, H.Lin, B.Xu, and L.Yang, "Detection of suicide ideation in social media forums using deep learning, "Algorithms, vol.13, no.1, pp. 7, 2020.
[23] J.A. Gonzalez, L.F. Hurtado, and F.Pla, "Transformer based contextualization of pre-trained word embeddings for irony detection in Twitter", Information Processing and Management, vol.57, no.4, pp.102262, 2020.
[24] P. Fortuna and S. Nunes, "A survey on automatic detection of hate speech in text, "ACM Computing Surveys (CSUR), vol.51, no.4, pp.1–30,2018.
[25] A.Karami, M.Lundy, F.Webb,andY.K.Dwivedi, "Twitterandresearch: a systematic literature review through text mining," IEEE Access, vol. 8, pp. 67698–67717, 2020.
[26] R. M. Silva, R. L. Santos, T. A. Almeida, and T. A. Pardo, "Towards automatically filtering fake news in Portuguese," Expert Systems with Applications, vol. 146,p. 113199, 2020.
[27] S.M.Rezaeinia, R.Rahmani, A.Ghodsi, and H.Veisi, "Sentiment analysis based on improved pre-trained word embeddings," Expert Systems with Applications, vol. 117, pp. 139–147, 2019.
[28] E.F.Unsvåg and B.Gambäck, "The effects of user features on twitter hate speech detection," in Proceedings of the 2nd workshop on abusive language online (ALW2), 2018, pp. 75–85.
[29] A. K. Sharma, S. Chaurasia, and D. K. Srivastava, "Sentimental short sentences classification by using CNN deep learning model with finely tuned word2vec, "Procedia Computer Science, vol.167, pp.1139–1147,2020.
[30] A. García-Pablos, M. Cuadros, and G. Rigau, "W2vlda: almost unsupervised system for aspect-based sentiment analysis," Expert Systems with Applications, vol. 91, pp. 127–137, 2018.
[31] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using bilstm-CRF and CNN, "Expert Systems with Applications, vol. 72, pp. 221–230, 2017.
[32] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality, "Advances in neural information processing systems, vol.26, 2013.
[33] S.Symeonidis, D.Effrosynidis, and A.Arampatzis, "A comparative evaluation of preprocessing techniques and their interactions for Twitter sentiment analysis," Expert Systems with Applications, vol. 110, pp. 298–310, 2018.
[34] U. Naseem, I. Razzak, and P. W. Eklund, "A survey of preprocessing techniques to improve short-text quality: a case study on hate speech detection on Twitter," Multimedia Tools and Applications, vol. 80, no. 28, pp. 35239–35266, 2021.
[35] K.Korovkinas,P.Danėnas,andG.Garšva, "Svmaccuracyandtraining speed tradeoff in sentiment analysis tasks, "International Conference on Information and Software Technologies, Springer,2018, pp.227–239.
[36] Bouazizi, M. and Ohtsuki, T., 2017. A pattern-based approach for multiclass sentiment analysis in Twitter. *IEEE Access*, *5*, pp.20617-20639.
[37] Loyola-González, O., Monroy, R., Rodríguez, J., López-Cuevas, A. and Mata-Sánchez, J.I., 2019. Contrast pattern-based classification for bot detection on twitter. *IEEE Access*, *7*, pp.45800-45817.
[38] Subbiah, S., Anbananthen, K.S.M., Thangaraj, S., Kannan, S. and Chelliah, D., 2022. Intrusion detection technique in wireless sensor network using grid search random forest with Boruta feature selection algorithm. *Journal of Communications and Networks*, *24*(2), pp.264-273.

P. Sharmila, M.E. (Computer Science and Engineering). She is currently pursuing her PhD (Information and Communication Engineering) at Anna University, Chennai, Tamil Nadu, India. Currently, she is working as an Assistant Professor in the Department of Computer Applications, Thiagarajar College of Engineering, Madurai, Tamil Nadu, India. Her current research interest centres on natural language processing, machine learning and deep learning

Dr Kalaiarasi Sonai Muthu Anbananthen is an Associate Professor in the Faculty of Information Science and Technology at Multimedia University (MMU), Malaysia. She was a programme coordinator for the Masters of Information Technology (Information System). She acts as a reviewer in various Scopus, and SCI indexed technical journals. She has published more than 80 articles in journals, conferences and book chapters. Her current research interests focus on Data Mining, Sentiment Analysis, Artificial Intelligence, Machine Learning, Deep Learning and Text Analytics.

Dr Deisy Chelliah is working as a Professor and Head of the Information Technology Department, Thiagarajar College of Engineering, Madurai, Tamilnadu, India. She published more than 70 articles in journals and conferences. She acts as a reviewer in various SCI and Scopus indexed technical journals. She completed two projects sponsored by Microsoft and AICTE. She is a member of ISTE and CSI. Her research area of interest includes Image Analysis and Text Analytics.

Sudhaman Parthasarathy, PhD, is working as a Professor of Data Science in the Thiagarajar College of Engineering, Madurai, India. As a habitual rank holder, he has been teaching at the post-graduate level for the past 20 years. He has published research papers in peer-reviewed conferences and International Journals such as *Computers in Industry, Software Quality Journal, International Journal of Project Management* and *Business Process Management Journal.* He has authored several chapters in the refereed edited books of IGI, USA and Springer, London. His current research

interests include enterprise information systems, ERP and software engineering.

Dr Subarmaniam Kannan has been a lecturer in the Faculty of Information Science and Technology, Multimedia University, since 2000. He has a PhD in Semantic Learning (Knowledge Engineering) from Multimedia University. He is also a Certified Information Systems Auditor (CISA) and Certified Cisco Networking Associate (CCNA) registrar and instructor for MMU-Melaka Local Networking Academy. He was programme coordinator for Data Communications and Networking Programme from 2013 to 2021. His research area includes semantic web technology, ontology and knowledge management, automatic speech recognition for Bahasa Malaysia and edge computing analytics.