International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(22s) 2026 pp. 477-491
https://doi.org/10.70917/ijcisim-2026-5455
Article

IMPROVED MULTILINGUAL HATE SPEECH DETECTION THROUGH INDEXED COPHENETIC MULTILAYEREDEXTREME MACHINE RELYING ON N-GRAMS AND HIDDEN MARKOV MODEL

Ms. N Zahira Jahan[1], Dr. R Pushpalatha[2]

[1]Research Scholar, Kongu Arts and Science College, Erode, Tamilnadu, India
& Assistant Professor, Department of Computer Applications, Nandha Engineering College, Erode, Tamilnadu, India
Email: zahirajahann999@gmail.com
[2]Associate Professor, Department of Computer Science, Kongu Arts and Science College, Erode, Tamilnadu, India
Email: rpljour@gmail.com

Abstract: This study consider hate speech detection in multiple languages such as English, Tamil, and Tanglish by achieving higher accuracy, precision, recall, and minimal training time utilizing machine learning based Natural Language Processing (NLP) techniques. Proposed Indexed Cophenetic Multilayered Extreme Learning based on N-Grams and Hidden Markov (ICMEL-NGHM) technique introduced for achieving multilingual hate speech detection. Proposed technique and two existing methods are compared for effectiveness in hate speech identification with diverse metrics. Accuracy of proposed technique for detecting hate speech in English, Tamil, and Tanglish classes is 97.89%, 97.32%, and 96.34%,. Precision of proposed technique in detecting hate speech for English, Tamil, and Tanglish classes is 97.34%, 95.34%, and 94.32%. For English, Tamil, and Tanglish groups, Recall in identifying hate speech of proposed ICMEL-NGHM is 99.23%, 96.34%, and 98.23%. Proposed technique detects hate speech in 12.35 ms for English class, 10.55 ms for Tamil class, and 9.45 ms for Tanglish class. Results of proposed technique is better than state-of-the-art approaches.
Keywords: Multilingual Hate Speech Detection, Machine Learning, Natural Language Processing, Multilayered Extreme Learnt Machine, Character level N-grams, Baum-Welch Hidden Markov, andCophenetic Correlation

1. INTRODUCTION

In recent years, people have been using online social networks like Facebook, Instagram, Twitter, and others to express themselves and share their thoughts more and more. Hate speech against individuals causes societal unrest and violent crimes. This has led to development of numerous machine learning methods to identify hate speech in various languages. In [1], study focused on Fine-Grained Multilingual Hate Speech Detection for discovering hate speech utilizing Explainable Artificial Intelligence and Transformers. In [2], the study was developed with leveraging transfer learning for hate speech detection. However, the accuracy was not efficiently enhanced. In [3], the So-haTRed was implemented with accurately identify hate speech in Turkish social media text. In [4], lightweight and multilingual hate speech detection model named MLHS-CGCapNet was developed with multiple languages. Study in [5] concentrated for increasing multilingual hate speech detection. Phrase vector embeddings were introduced in [6]. In [7], study was presented with Gated Recurrent Units (GRU) and Bidirectional Encoder Representations from Transformers (BERT) named G-BERT. In [8], transfer learning technique was implemented with the aim of

performing hate speech detection in social media. In [9], the study was developed by leveraging syntactic information and sentiment knowledge. Language detection models were trained on a single German dataset can generalize to other, comparable datasets in [10]. Study in [11] concentrated on using bootstrapping and label modification. BERT and Hate Speech Word Embedding with Deep Model used [12]. Study's main goals in [13] were to design user interfaces for AI-based decision support systems. Study was focused [14] on operationalization of hate speech. In [15], Roman Urdu text developed. However, computational complexity remained unaddressed.

1.1 Contribution of the work

- Proposed ICMEL-NGHM technique is implemented for ensuring effective hate speech detection in multiple languages.
- Three pre-processing steps provide pre-processed speech in first hidden layer.
- Robust linear regression is applied with Rand Similarity Index in second hidden layer for extracting relevant features.
- Cophenetic Correlation Coefficient employed in third hidden layer to classify data into two different classes.
- The proposed technique is compared to two existing methods to demonstrate hate speech detection performance of proposed research work.

1.2 Organization of the paper

It given below, Section 2 introduces study region. Section 3 explains proposed technique. Results presented in Sections 4 and 5. Section 6 concludes with final thoughts.

2. RELATED WORKS

Study in [16] concentrated on Multi-class Hate Speech Detection in the Norwegian Language using Multilingual Fine-Tuned Transformers and Fast Recurrent Neural Network (FAST-RNN). In [17], Hybrid Deep Learning with FastText and Explainable AI was developed for performing fake news detection. However, precision was not enhanced to an efficient level. Study's main goals were to evaluate significance of contextual information in [18]. In [19], study focused to identify and mitigate hate speech targeting specific groups based on religion, ethnicity, nationality, gender, or sexual orientation. In [20], study was focused on the Roman Urdu Hate Speech Detection on Social Media Platform. In [21], the study was focused on SocialHaterBERT with enhancing accuracy. However, the model might either fail to detect hate speech false negative. By using linguistic features and transformers, the work in [22] concentrated on feature combination strategies in Spanish. In [23], study focused on the Probabilistic Clustering Model with the goal of performing the hate speech classification in Twitter. However, it failed to enhance accuracy to an efficient level. In [24], Linguistic Features and Word Embedding Approach were implemented. In order to increase the precision and resilience of hate speech detection models, in [25] integrated an adaptive ensemble learning model with a case study on the virus. Research was created in [26] to track and examine social media sites for hate speech. However, the recall was not enhanced to an efficient level. In order to identify the most suitable model for this task, the pre-trained language models were compared in [27] for performing the Spanish hate speech detection. However, it was not able to efficiently minimize the amount of time consumed for Spanish hate speech detection. The study used a hate speech recognition technique in [28] to find hate speech directed at marginalized minority groups on social media. However, it failed to enhance the accuracy efficiently. By using the Offensive Language Identification Dataset for Brazilian Portuguese (OLID-BR) dataset, the work in [29] attempts to the advancement of research and development of NLP techniques for identifying offensive language in Brazilian Portuguese. However, the time consumption was not minimized effectively. In [30], the study was focused on the hate and offensive speech detections depending on four kinds of hate such as religion, ethnicity, nationality, and gender in the Arabic social media with the

aim of discovering and mitigating harmful content that promotes discrimination, violence, and hatred. However, the precision and recall were not enhanced to an efficient level.

Improved Ant Lion Optimizer with Deep Learning Dirven Offensive and Hate Speech Detection (IALODL-OHSD) was developed in [31] for increasing the classifier result. Hybrid model was introduced in [32] for Roman Urdu hates speech detection with higher precision. Hybrid approach was designed in [33] with learning speed based on performance. But, the training time was higher.

3. PROPOSED METHODOLOGY

Hate speech detection in multiple languages presents major challenges in Computer Science and Engineering. Social media has vital part of daily lives in the modern world. Social media and other online communication platforms have revolutionized how individuals interrelate and distribute ideas. Automatic hate speech detection on social media is significant issue. It is essential that hate speech and non-hate speech be accurately classified. This makes it simple for us to recognize actual individuals who pose a risk to our society. Development of efficient machine learning methods for real-time applications is required to identify hate speech in many languages. Because of this, proposed research work is implemented with advantage of tackling issues of multilingual hate speech detection by combining three essential sections such as pre-processing, feature extraction, and classification. To done this, novel framework called ICMEL-NGHM technique is implemented for automatically detecting hate speech in online social platforms. As the first of hate speech detection, proposed technique collects AI Tamil Hate Speech Detector dataset containing relevant features with speech samples and corresponding speech labels (hate speech and non-hate speech). Figure 1 illustrates architectural diagram of proposed technique.

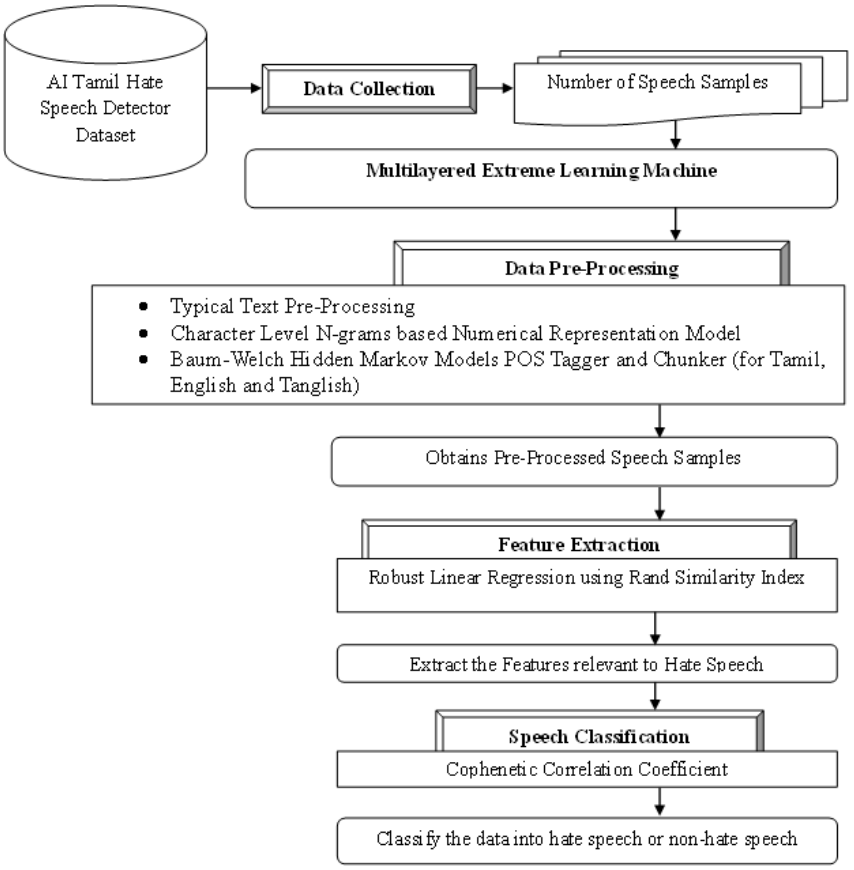

Fig. 1 Overview of the Proposed Technique for Multilingual Hate Speech Detection

As shown in Fig.1, proposed technique is implemented with Multilayered Extreme Learning Machinecontains different layers to perform three important sections such as data pre-processing, feature extraction, classification in the detection of hate speech in several languages. At first, the AI Tamil Hate Speech Detector dataset which contains texts in English, Tamil, and Tanglish, is the initial source of input data samples used by the proposed technique. Secondly, the proposed technique performs pre-processing in the first hidden layer by the implementation of three different steps including typical text pre-processing, Character Level N-grams based Numerical Representation model, Baum-Welch Hidden Maarkov Models POS Tagger and Chunker (for Tamil, English and Tanglish). After the pre-processed speech samples obtained, the proposed technique extracts relevant features in the second hidden layer by performing robust linear regression based on Rand Similarity Index. Finally, the proposed technique employs classification in the third hidden layer by determining Cophenetic Correlation Coefficient according to the relevant features selected. Therefore, the proposed technique effectively categorizes the input speech samples into two different classes such as hate speech and non-hate speech.

Data Collection

The most important and initial stage is to collect a varied dataset that includes different languages and dialects. It is imperative that the gathered information be appropriately classified as either 'hate speech' or 'non-hate speech'. The proposed technique utilizes the AI Tamil Hate Speech Detector dataset as a valuable resource for identifying the hate speech and non-hate speech in three languages. In this dataset, the four types of hate speech such as political hate speech, gender-based hate speech, religion-based hate speech, and community-based hate speech are discussed.

TABLE 1 which is provided below lists the speech samples from the dataset of AI Tamil Hate Speech Detector.

S. No	Language	Speech in social media
1	Tamil	அருமை படைப்புகள் நன்றி சகோ
2	Tamil	செமையா சகோ
3	Tanglish	Motivating Atma என் பொண்ணு பார்க்க போறியா
4	Tanglish	Good technical மளினி கேள்விக்கு பதில் சொல்லாமல் திரும்ப கேள்வி கேட்டுமடக்கிறா
5	English	FREEDOM OF LIFE
6	English	Good speech and bold speech ..still looking same school kid

As shown in TABLE 1, AI Tamil Hate Speech Detector dataset contains the three different linguistic and social media speech data.

TABLE 2Data (speech samples) distribution by class within the AI Tamil Hate Speech Detector dataset.

Class	English	Tamil	Tanglish
Hate-speech (HS) - 'Cl_1'	1535	1240	350
Non-hate-speech (NHS) -'Cl_2'	600	600	675
Total	2135	1840	1025

As illustrated in TABLE 2, AI Tamil Hate Speech Detector dataset contains social media comments in Tamil, English, and Tanglish which are divided into two different classes. Let's take into account that the proposed technique incorporated with training set as '$\{SS, Y\}$'. Here, the incoming strings such as speech samples'$SS = \{SS_1, SS_2, \ldots., SS_n\}$' and features'$F = \{f_i, f_j\}$' taken as input from social media that might be included in the classes

'$Y = \{Cl_1, Cl_2, \ldots, Cl_N\}$'. The proposed technique classifies incoming strings into two classes such as hate speech and non-hate speech. The hate speech 'HS' is classified as 'Cl_1' and the non- hate speech 'NHS' is classified as 'Cl_2'.The Multilayered Extreme Learning Machine is a neural network architecture that leverages the efficiency and ease of Extreme Learning Machines to create deeper and more potent models. The Multilayered Extreme Learning Machine is constructed with different layers such as one input layer, multiple hidden layers and one output layer. The inputs are passed through several hidden layers. The input features are represented by numerous neurons in each layer. The neuronal connections are initiated with weights and bias at randomly. A random weight and bias are assigned to each neuron. In the first, second, and third hidden layer of multilayered extreme learnt machine, the proposed technique carries out pre-processing, feature extraction, and classification respectively. Initially, the input layer takes in the provided speech samples'$SS = \{SS_1, SS_2, \ldots., SS_n\}$' as input and constructs the random weights matrix 'w_{ij}' as follows.

$$w_{ij} = [w_{11}\, w_{12}\, \ldots\, w_{1n}\, w_{21}\, w_{22}\, \ldots\, w_{2n}\, \vdots\vdots\, \ldots\, \vdots\, w_{m1}\, w_{m2}\, \ldots\, w_{mn}\,] \quad (1)$$

The bias is added in the hidden layer, where the value is kept at '1'and the weight '$w_1, w_2, \ldots, w_m$' related to the input layer is randomly assigned. The generated input weights did not change during the operation. The input layer determines the activity of the neuron '$X(t)$' as given below.

$$X(t) = \sum_{i=1}^{n} [SS_i * w_{ij}] + b \quad (2)$$

Here, the weight between thej^{th} input layer neuron and thei^{th} hidden layer neuron is indicated as 'w_{ij}', and the bias is denoted as 'b'. Second, inputs are forwarded to the hidden layers from the input layer. The subsequent hidden layer uses the outputs from the previous one as inputs. The output layer generates the final class prediction probabilities for hate speech and non-hate speech.

3.1 Pre-Processing (at the first hidden layer)

The fundamental module in the hate speech detection pipeline is pre-processing, particularly when working with multilingual datasets**.** For the input data to be of higher quality, pre-processing is essential. The process of converting the raw data into an appropriate format for further analysis is known as pre-processing. Besides, the Part-of-Speech (POS) tagging and chunking are fundamental techniques in the NLP tasks used to extract syntactic structure from the text data. The POS Tagger and Chunker are utilized for assigning POS tag to the sentence and parsing tagged sentence to identify groups of words that form syntactic units respectively. From that, the POS Tagger and Chunker help to identify key entities and relationships within text making it easier to extract relevant information. Besides, the machines are better able to interpret its meaning and intent of the text by understanding the grammatical structure of text. The POS tagger assigns a grammatical category (i.e., POS tag) to each word in a sentence (i.e., text) based on its context and grammatical function. These tags are nouns, verbs, adjectives, adverbs, prepositions, and conjunctions, etc. For the Tamil language, the Bureau of Indian Standards has registered 47 POS tags for standardization. There is 17 tags in the English POS tags. The Chuker discovers and segments phrases within a sentence. These phrases are noun phrases (NPs), verb phrases (VPs), or prepositional phrases (PPs). In our work, the proposed technique carries out pre-processing as initial process in the first hidden layer which includes three essential steps such as typical text pre-processing, Character Level N-grams based Numerical Representation model, and Baum-Welch Hidden Maarkov Models based POS Tagger and Chunker (for Tamil, English and Tanglish) while taking the inputs from the input layer.

i) Typical text pre-processing

The text pre-processing is a crucial step in NLP tasks including hate speech detection. In order to prepare the raw text data for machine learning models, it must be cleaned, transformed, and normalized.

- **Text cleaning:** The text cleaning is the procedure of removing the noise or irrelevant information (i.e., unnecessary characters) such as HTML tags, URLs, punctuation, whitespace, and special characters because it might not contribute to the meaning of the text.
    - **HTML tags-**HTML tags are not relevant for text analysis. They are used to organize web pages.
    - **URLs-**URL links to websites are eliminated because they don't add to the text's semantic meaning.
    - **Punctuation-** Although crucial for human discourse, punctuation occasionally make machine learning algorithms less effective.
    - **Whitespace-**The extra spaces, tabs, and newlines are normalized to improve readability and processing efficiency.
    - **Special characters-**The emojis, symbols, and accents are examples of special characters that are eliminated or substituted with more conventional representations.
- **Text normalization:** The text normalization is the method of converting text to a standard format by removing stop words. It involves removing common words that do not hold significant meaning such as 'the', 'and', and 'of'.
    - **Stemming-** It reduces words to their root form to improve feature representation. (e.g., 'running' to 'run').
    - **Lemmatization-**It reduces words to their base form (e.g., 'better' to 'good').
- **Tokenization:** The tokenization is the process of breaking the text into individual words or tokens.

ii) Character Level N-grams based Numerical Representation model

In NLP tasks, the Character-Level N-grams based Numerical Representation is a method that divides text into discrete characters.It handles each character as a distinct token, in contrast to word-level tokenization, which divides text into words.It indeed significantly reduces the vocabulary size compared to word-level tokenization especially for languages with large vocabularies. It is capable of handling words that are absent from the training vocabulary, particularly in rare or richly morphological languages. It is more resilient to faults in formatting, typos, and misspellings. From that, the Character-Level N-grams based Numerical Representation is the process of converting the characters in the text into numerical representations by obtaining unigrams, bigrams, and trigrams for given text. For instance, there are about 170,000 words in the English language's lexicon, but we only utilize 256 distinct characters (letters, numerals, and special characters).

Let's assume that the input speech sample 'IP_{ss}' is the text below.

$$IP_{ss} = ['FREEDOM\ OF\ LIFE'] \quad (3)$$

This input text's formulation after it has been broken down into individual characters by character level tokenization is as follows.

$$CLT = ['F','R','E','E','D','O','M','O','F','L','I','F','E'] \quad (4)$$

As shown above, the input text is break into their constituent individual characters. Followed by, the single characters, pairs of consecutive characters, triplets of consecutive characters (i.e., unigrams, bigrams, and trigrams) of the given input text are obtained by the performance of N-gram creation.

Unigrams-Single characters:

$$Uni_{g=}['F','R','E','E','D','O','M','O','F','L','I','F','E'] \quad (5)$$

Bigrams-Pairs of consecutive characters:

$$Bi_g = ['FR','RE','EE','ED','DO','OM','MF','OF','FL','LI','IF','FE'] \quad (6)$$

Trigrams-Triplets of consecutive characters:

$$Tri_g = ['FRE','REE','EED','EDO','DOM','OMF','MFO','OFF','FLI','LIF','IFE'] \quad (7)$$

After that, all unique characters in the dataset are identified as given below.

$$A -> 1, B -> 2, \ldots Z -> 26 \quad (8)$$

Followed by, the unique integer index is assigned for each n-gram character in the text with the creation of character vocabulary. Finally, each character in the text(i.e., tokenized text) is replaced with its corresponding index(i.e., sequence of integers)by the application of numerical encoding. Thus, the numerical representation for each text is obtained in a significant manner.

iii)Baum-Welch Hidden Maarkov Models based POS Tagger and Chunkermodel (for Tamil, English and Tanglish)

In the proposed technique, Baum-Welch Hidden Maarkov Models are employed for POS Tagging and Chunking (Tamil, English and Tanglish)by learning the probabilities of transitioning between different POS tags and Chuck labels and the probabilities of emitting different words given a specific POS tag and Chuck labels. The Hidden Maarkov Models are affective for sequential data in order to predict POS tags and chunk boundaries especially for languages like Tamil, English and Tanglish. The Baum-Welch algorithm is an iterative expectation-maximization (EM) algorithm utilized for determining the parameters of a Hidden Maarkov Models.From that, the proposed technique utilizes Baum-Welch Hidden Maarkov Models to significantly discover the most likely sequence of hidden states (POS tags or chunk labels) for a given sequence of words by learning the probabilistic relationships between words and their underlying tags. i.e., This involves calculating the probability of each possible tag sequence and selecting the highest probability sequence. Here, the sequence of words is treated as a sequence of observations, and the hidden states are treated as the underlying linguistic structure such as POS tags or Chunk labels.

At first, the transition probability matrix '$A\ (i,j)$' is formed by initializing the probability of transition matrix 'A' from one POS tag (state 'i') to another (state 'j'). The likelihood of changing from one hidden state (POS tag or chunk label) to another is represented by this matrix. This matrix describes the hidden Markov process's behaviour. Secondly, the emission probability matrix '$B\ (k,j)$' is constructed by initializing the probability of emission observation 'k' given state 'j'. This matrix represents the probability of observing a particular word given a specific POS tag (hidden state). The connection between the observed outputs and the hidden states is specified by this matrix. Followed by, the initial state probability vector '$\pi\ (i)$' is formed by initializing the probability of starting in a particular POS tag (state 'i')which indicates the probability of starting in a particular hidden state. In order to increase the probability of the observed data, the Baum-Welch algorithm iteratively re-estimates these parameters. Given the entire observation sequence, the probability of being in state 'i' at time 't' and changing to state 'j'at time '$t+1$' ($\gamma_{t(i.j)}$) is ascertained as follows.

$$\gamma_{t(i.j)} = \alpha_{t(i)} * x(i,j) * y\left(j, o_{(t+1)}\right) * \beta_{(t+1)(j)} \,/\, p(O) \quad (9)$$

Here, '$\alpha_{t(i)}$' is the probability of being in state 'i' at time 't' given the observation sequence up to time 't' which estimates the probability of a partial observation sequence. '$\beta_{t\,(i)}$' is the probability of observing the sequence of observations from time '$t+1$' to the end, given state 'i' at time 't'which determines the probability of a future observation sequence. '$x(i,j)$' represents the transition probability from state 'i' to state 'j'. '$y(i,k)$' indicates the emission probability from observing observation 'k' to state 'i' Probability of being in state 'i' at time 't' given the observation sequence($\gamma_{t(i)}$) is expressed as.

$$\gamma_{t(i)} = \sum_j \quad \gamma_{t(i,j)} \text{ (10)}$$

Below mathematical expression for re-estimation of the transition probabilities '$x'(i,j)$'.

$$x'(i,j) = \frac{\sum_t \quad \gamma_{t(i,j)}}{\sum_t \quad \gamma_{t(i)}} \text{ (11)}$$

The likelihood of emissions is recalculated '$y'(i,k)$' as follows.

$$y'(i,k) = \frac{\sum_t \quad [\gamma_{t(i)} * I(\, o_{(t)} = v_k)]}{\sum_t \quad \gamma_{t(i)}} \text{(12)}$$

Here, $p(O)$ is probability of the observation sequence and $I(\, o_{(t)} = v_k\,)$ is the indicator function which is equal to 1 if $o_{(t)} = v_k$, 0 otherwise. It is utilized for counting the number of times a particular observation is emitted from a state.Then, the model parameters are iteratively re-estimated until convergence by the Baum-Welch.

3.2 Feature Extraction (at the second hidden layer)

The features play a crucial role in managing the input data. The important features must be chosen for efficient data classification. The primary goal of feature selection is to identify the most informative feature in order to reduce the dataset's dimensionality. One of the most important steps in the hate speech detection procedure is feature extraction. After the pre-processed text data i.e., speech samples obtained, the proposed technique performs feature extraction with the implementation of robust linear regression using Rand Similarity Index. Here, the robust linear regression is performed by extracting the more relevant features for hate speech detection depending on the Rand Similarity Index. TheRand Similarity Index is a statistical method utilized formore relevant featuresfor hate speech detection by estimating the relationship between the variables (i.e. features) as given below.

$$RSI = \left[\frac{[f_i \cap f_j]}{N}\right] \qquad \text{(13)}$$

As shown in above equation (13), the Rand Similarity Index'RSI' is determined. Here, the mutual dependence between the features is denoted as '$f_i \cap f_j$'. The output that it returns falls between 0 and 1. The following is the selection process for hate speech identification using the well-matched major aspects from this index value 'Z'.

$$Z = \{1; \qquad relevant\ features\ \ 0; \qquad irrelevant\ \ features \qquad \text{(14)}$$

As shown in above equation (14), two features that are highly matched are indicated by an index value of 1, while those that are not matched are indicated by an index value of 0.The features that match the most are chosen as important and relevant and the others are eliminated. These pertinent features are utilized in the identification of hate speech in order to reduce the complexity of time for classifying the data into two different classes.

3.3 Speech Sample Classification (at the third hidden layer)

The process of giving a specific data point a class label is known as classification. In the context of detecting hate speech, this entails classifying texts i.e., speech sample as either 'hate speech' or 'non-hate speech'. In our work, the proposed technique utilizes Cophenetic Correlation Coefficient in the third hidden layer for the purpose classifying the speech samples with features extracted into two different classes such as hate speech and non-hate speech depending on the significant features extracted. The Cophenetic Correlation Coefficient is taken into account for speech classification by determining the distance between the training input speech samples and testing speech samples.The degree to which a dendrogram maintains the distances between the original data points is shown statistically by the Cophenetic Correlation. It is attained as given below.

$$CCC = \frac{\sum_{i=1}^{n} \quad (ss_i{}^{tr} - \underline{ss^{tr}})(ss_i{}^{ts} - \underline{ss^{ts}})}{\sqrt{\sum_{i=1}^{n} \quad (ss_i{}^{tr} - \underline{ss^{tr}})^2 \sum_{i=1}^{n} \quad (ss_i{}^{ts} - \underline{ss^{ts}})^2}} \text{ (15)}$$

As shown in above equation (15), the Cophenetic Correlation Coefficient 'CCC' is estimatedto examine the testing and training data with the selected relevant features. Here, the training pattern of the speech sample is indicated as '$SS_i{}^{tr}$', the testing pattern of thespeech sample is denoted as '$SS_i{}^{ts}$', the average of the training pattern of the speech sample is symbolized as '$\underline{SS^{tr}}$', and the average of the testing pattern of the speech sample is denoted as '$\underline{SS^{ts}}$'.The outcomes of this coefficient are -1 or +1. If '$Y = +1$', the speech sample is classified as hate speech, Otherwise, the speech sample is classified as non-hate speech if '$Y = -1$'. Following speech classification, the gradient descentfunction '$\frac{\partial loglog\ p(Y)}{\partial\ \alpha_t}$' is used to update the layer weights in the output layer as given below.

$$w'_{ij} = w_{ij} + \left[Lr * \frac{\partial loglog\ p(Y)}{\partial\ \alpha_t}\right] (16)$$

As shown above equation (16), the weights between the layers are updated in the output layer are updated in order to minimize error. Here, the updated weight is denoted as 'w'_{ij}', the current weight is indicated as 'w_{ij}', the learning rate is denoted as 'Lr', and the probability of outcomes at the output layer is signified as '$p\ (Y)$'. It finds the local minimum of a function (i.e., error rate) by changing the current weight. This process is repeated until the smallest fault is found. The square of the discrepancy between the expected and actual outcomes is then used to calculate the error rate. The following is how this is expressed mathematically.

$$E = \frac{1}{2}(Y_A - Y_P)^2 (17)$$

As shown in above equation (17), the error rates 'E' is determined while taking into account the actual outcomes 'Y_A', and the predicted classification results 'Y_P'. Then, the final projected class probabilities for every class are generated by the output layer. Hence, the proposed technique successfully achieves better performance in the hate speech identification by precisely and timely discovering the hate speech and non-hate speech in the multiple languages. The algorithmic representation of the proposed technique for effective multilingual hate detection is obtained as given below algorithm 1.

Algorithm 1: Indexed Cophenetic Multilayered Extreme Learning based on N-Grams and Hidden Markov technique
Input: AI Tamil Hate Speech Detector dataset,Speech Samples '$SS = \{SS_1, SS_2, \ldots., SS_n\}$', Features '$F = \{f_i, f_j\}$' **Output:** Robust hate speech classification for multiple languages
Begin **1: Take** number of Speech Samples '$SS = \{s_1, s_2, \ldots., s_n\}$', and Features '$F = \{f_i, f_j\}$' as input **[**At the input layer] **2:Define** the neurons' activity in the input layer as shown in equation (2) **3: Forwards** the input to the hidden layers **4: For** each data i.e., input speech samples**[**At the first hidden layer**]** **5: Perform** three different pre-processing steps **6: Obtain** pre-processed speech samples **7: For** each feature from pre-processed speech sample as input at the first hidden layer[At the second hidden layer] **8: Estimate** Rand Similarity Index as shown in equation (13) **9: if** $(Z = 1)$ **then**

10: Features are highly matched

11: else ($Z = 0$)

12: Features are not matched

13: Extract features that are relevant to hate speech

14:Discard other features

15: End if

16: End For

17:For each extracted significant feature [At the third hidden layer]

18: Determine Cophenetic Correlation Coefficient as shown in equation (15)

19: if ($Y = +1$) **then**

21: **Classify** the data (i.e., speech sample) as hate speech

22: **else** ($Y = -1$) t**hen**

23: Classify the data (i.e., speech sample) as non-hate speech

24:End if

25:End For

// [At the output layer]

26:For each obtained outcome

27:Update the weight as shown in equation (16)

28: Estimate error rate by using equation (17)

29: Attain thefinal projected class probabilities for every classat the output layer

30: End For

31: End For

Algorithm 1 Indexed Cophenetic Multilayered Extreme Learning based on N-Grams and Hidden Markov technique for multilingual hate speech detection

In algorithm above, proposed technique makes use of a multilayered extreme learning machine. Data collection is to gather varies quantities of speech samples from AI Tamil Hate Speech Detector dataset. Input layer takes provided speech samples as input. Pre-processing is employed in first hidden layer to provide pre-processed speech samples. Then, robust linear regression using Rand Similarity Index is carried out in second hidden layer to extract pertinent features. Speech classification is carried out in the third hidden layer for categorizing input data as hate speech and non-hate speech utilizing Cophenetic Correlation Coefficient.Then, the output layer obtains the final predicted class probabilities for each class.

4. EXPERIMENTAL SETUP

Proposed technique and two existing method uses Python high-level general-purpose programming language. In order to detect hate speech on social media platforms in Tamil, English, or Tanglish, the proposed technique takes into account the AI Tamil Hate Speech Detector dataset that was taken from https://github.com/dreamspace-academy/ai-tamil-hate-speech-project. Number of data samples is used as input.

Investigation of the Accuracy

Number of accurate forecasts (i.e., true positive and true negative) divided by the total number of predictions is known as accuracy. It is determined in percentage (%.)

$$Acc = \left[\frac{Tpos+Tneg}{Tpos+Tneg+Fneg+Fpos}\right] (18)$$

In equation (18), True Positive '$Tpos$'is accurately identified as hate speech, True Negative '$Tneg$' is accurately identify non-hate speech. False Negative '$Fneg$' is misidentification of hate speech as non-hate speech and False Positive '$Fpos$'is incorrectly categorizes non-hate speech as hate speech.

TABLE 3 Accuracy of three different methods for hate speech class detection (English, Tamil, Tanglish) while using the speech data samples from the AI Tamil Hate Speech Detector dataset. For the English, Tamil, and Tanglish hate speech classes, the accuracy of the proposed technique is 97.89%, 97.32%, and 96.34% respectively. The accuracy of the existing [1] for the hate speech classes in English, Tamil, and Tanglish is 88.23%, 89.23%, and 84.54% respectively. For the hate speech class (English, Tamil, and Tanglish), the existing [2] has accuracy rates of 78.34%, 85.34%, and 80.23% respectively.

AI Tamil Hate Speech Detector dataset with different classes	Accuracy (%)		
	Existing [1]	Existing [2]	Proposed technique
Hate speech class (English)	88.23	78.34	97.89
Hate speech class (Tamil)	89.23	85.34	97.32
Hate speech class (Tanglish)	84.54	80.23	96.34

TABLE 3 shows accuracy. It enhanced by 11% and 20% than [1] [2] uses proposed technique.

Results of Precision

True positive rate (TP), or hate speech identified as hate speech, and the false positive rate (FP), or hate speech identified as non-hate speech, are used to calculate precision 'Pre'.

$$Pre = \frac{TP}{TP+FP} (19)$$

TABLE 4 Results of comparing the F1-score for the hate speech class (English, Tamil, and Tanglish) with the AI Tamil Hate Speech Detector dataset. The precision of the proposed technique is97.34%, 95.32%, and 94.32%, for hate speech class (English, Tamil, and Tanglish). 87.23%, 88.23%, and 87.23% are the precision for English, Tamil, and Tanglish hate speech classes of the existing [1]. The precision of the existing [2] is 82.32%, 84.34%, and 82.34% for hate speech class (English, Tamil, and Tanglish).

AI Tamil Hate Speech Detector dataset with different classes	Precision (%)		
	Existing [1]	Existing [2]	Proposed technique
Hate speech class (English)	87.23	82.32	97.34
Hate speech class (Tamil)	88.23	84.34	95.34
Hate speech class (Tanglish)	87.23	82.34	94.32

TABLE 4 provides precision. The precision of proposed technique is enhanced up to 9% and 15% when compared to existing [1] and [2].

Case scenario of Recall

True positive rates, or TP (hate speech identified as hate speech) and FN (non-hate speech identified as hate speech), respectively, are used to calculate recall 'Rec'.

$$Rec = \frac{TP}{TP+FN} \quad (20)$$

TABLE 5 Recall of three different methods for hate speech class detection (English, Tamil, Tanglish). The recall of the proposed technique is97.34%, 95.32%, and 94.32%, for hate speech class (English, Tamil, and Tanglish). 87.23%, 88.23%, and 87.23% are the recall for English, Tamil, and Tanglish hate speech classes of the existing [1]. The recall of the existing [2] is 82.32%, 84.34%, and 82.34% for hate speech class (English, Tamil, and Tanglish).

AI Tamil Hate Speech Detector dataset with different classes	Recall (%)		
	Existing [1]	Existing [2]	Proposed technique
Hate speech class (English)	88.23	80.23	99.23
Hate speech class (Tamil)	84.34	82.34	96.34
Hate speech class (Tanglish)	88.34	83.43	98.23

TABLE 5 measures recall. Recall of proposed technique increased by 13% and 19% than [1] and [2].

Impact of Training time

Length of time needed to distinguish hate speech and non-hate speech samples to execute hate speech detection is known as the training time. It determined in milliseconds (ms).

$$TT = \sum_{i=1}^{N} \quad S_i * Time\ (identify\ single\ speech\ data\ sample\ as\ hate\ speech\ or\ not) \quad (21)$$

TABLE 6 Performance outcomes three different methods in terms of training time using AI Tamil Hate Speech Detector dataset for hate speech class (English, Tamil, Tanglish). The proposed technique detects hate speech in 12.35ms, 10.55ms, and 9.45ms for class English, Tamil, and Tanglish, respectively. For class English, Tamil, and Tanglish, the existing [1] detects hate speech in22.35ms, 23.95ms, and 26.85ms respectively. The hate speech is detected by the existing [2] in the English, Tamil, and Tanglish classes in 15.25ms, 19.25ms and 24.45ms, respectively.

AI Tamil Hate Speech Detector dataset with different classes	Training time (ms)		
	Existing [1]	Existing [2]	Proposed technique
Hate speech class (English)	22.35	15.25	12.35
Hate speech class (Tamil)	23.95	19.25	10.55
Hate speech class (Tanglish)	26.85	24.45	9.45

TABLE 6 displays training time. It minimized by 55%, 42% than [1], [2] with proposed technique.

5. CONCLUSION and FUTURE WORK

Proposed ICMEL-NGHM uses multilayered extreme learning machine with numerous layers whereas three important procedures including pre-processing, feature extraction and classification are employed. Proposed technique

quickly and accurately guarantees detection of hate speech in multiple languages. Experiment is conducted by proposed technique and two existing methods in various metrics. Observed findings of proposed technique outperform earlier ones in achieving effective multi language hate speech detection. In future work, proposed method is further enhanced to apply novel machine learning algorithm for hate speech detection. Also, the various datasets will be considered for enhancing the hate speech detection performance.

DECLARATIONS

Conflict of Interest

The authors declare no potential conflict of interest.

Consent for Publication

Not applicable.

Ethics approval and consent to participate

Not applicable.

Data Availability Statement

Data are available within the article. The data have been gathered from (https://github.com/dreamspace-academy/ai-tamil-hate-speech-project).

6. ACKNOWLEDGEMENTS

None.

REFERENCES

1. Siddiqui, J.A., Yuhaniz, S.S., Shaikh, G.M., Soomro, A., Mahar, Z. A.: Fine-Grained Multilingual Hate Speech Detection Using Explainable AI and Transformers. IEEE Access, 12, 143177-143192 (2024). **DOI:** 10.1109/ACCESS.2024.3470901
2. Ramos, G., Batista, F., Ribeiro, R., Fialho, P., Moro, S., Fonseca, A., Guerra, R., Carvalho, P., Marques, C., Silva, C. :Leveraging Transfer Learning for Hate Speech Detection in Portuguese Social Media Posts. IEEE Access. 12, 101374 – 101389 (2024). **DOI:** 10.1109/ACCESS.2024.3430848
3. Altinel, A. B., Baydogmus , G.K., Sahin, S., Gurbuz, M. Z.: So-haTRed: A Novel Hybrid System for Turkish Hate Speech Detection in Social Media With Ensemble Deep Learning Improved by BERT and Clustered-Graph Networks. IEEE Access. 12, 86252-86270 (2024). **DOI:** 10.1109/ACCESS.2024.3415350
4. Kousar, A., Ahmad, J., Ijaz, K., Yousef, A., Shaikh, Z.A., Khosa, I., Chavali, D., Anjum, Mohd.: MLHS-CGCapNet: A Lightweight Model for Multilingual Hate Speech Detection. IEEE Access. 12, 106631- 106644 (2024). **DOI:** 10.1109/ACCESS.2024.3434664
5. Hashmi, E., Yayilgan, S. Y., Hameed, I. A., Yamin, M. M., Ullah, M., Abomhara, M.: Enhancing Multilingual Hate Speech Detection: From Language-Specific Insights to Cross-Linguistic Integration. IEEE Access. 12, 121507-121537 (2024). **DOI:** 10.1109/ACCESS.2024.3452987
6. Devi, V. S., Kannimuthu, S., Madasamy, A.K.: The Effect of Phrase Vector Embedding in Explainable Hierarchical Attention-Based Tamil Code-Mixed Hate Speech and Intent Detection. IEEE Access. 12, 11316-11329 (2023). **DOI:** 10.1109/ACCESS.2024.3349958
7. Keya, A.J., Kabir, M. Md., Shammey, N.J., Mridha, M. F., Islam, R. Md., Watanobe, Y. : G-BERT: An Efficient Method for Identifying Hate Speech in Bengali Texts on Social Media. IEEE Access.11,79697-79709 (2023). **DOI:** 10.1109/ACCESS.2023.3299021
8. Yuan, L., Wang, T., Ferraro, G., Suominen, H., Rizoiu, M-A.: "Transfer learning for hate speech detection in social media", Springer, Journal of Computational Social Science, Volume 6, 2023, Pages 1081-1101. https://doi.org/10.1007/s42001-023-00224-9
9. Fan, X., Liu, J., Liu, J., Tuerxun, P., Deng, W., Li, W. : Identifying Hate Speech Through Syntax Dependency Graph Convolution and Sentiment Knowledge Transfer. IEEE Access.12, 2730-2741 (2023). **DOI:** 10.1109/ACCESS.2023.3347591
10. Seemann, N., Lee, Y. S., Hollig, J., Geierhos, M.: Generalizability of Abusive Language Detection Models on Homogeneous German Datasets. Springer, Datenback-Spektrum. 23, 1-25, (2023). DOI:10.1007/s13222-023-00438-1

11. Bigoulaeva, I., Hangya, V., Gurevych, I., Fraser, A.: Label modification and bootstrapping for zero-shot cross-lingual hate speech detection. Springer. Language Resources and Evaluation. 57, 1515-1546 (2023). DOI:10.1007/s10579-023-09637-4
12. Saleh, H., Alhothali, A., Moria, K. : Detection of Hate Speech using BERT and Hate Speech Word Embedding with Deep Model. Taylor and Francis. Applied artificial intelligence. 37 (1), e2166719-405 (2023). https://doi.org/10.1080/08839514.2023.2166719
2. 13. Meske, C. & Bunde, E.: Design Principles for User Interfaces in AI-Based Decision Support Systems: The Case of Explainable Hate Speech Detection. Springer. Information Systems Frontiers. 25, 743-773 (2023). https://doi.org/10.1007/s10796-021-10234-5
13. Papcunova, J., Martoncik, M., Fedakova, D., Kentos, M., Bozoganova, M., Srba, I., Moro, R., Pikuliak, M., Simko, M., Adamkovis, M. :Hate speech operationalization: a preliminary examination of hate speech indicators and their structure. Springer. Complex and Intelligent Systems. 9, 2827-2842 (2023). DOI:10.1007/s40747-021-00561-0
14. Mehmood, F., Ghafoor, H., Asim, M. N., Ghani, M. U., Mahmood, W., Denge, A.: Passion-Net: a robust precise and explainable predictor for hate speech detection in Roman Urdu text. Springer. Neural Computing and Applications. 36, 3077-3100 (2024). DOI:10.1007/s00521-023-09169-6
15. Hashmi, E. & Yayilgan, S. Y.: Multi-class hate speech detection in the Norwegian language using FAST-RNN and multilingual fine-tuned transformers. Springer. Complex and Intelligent Systems. 10, 4535-4556 (2024). DOI:10.1007/s40747-024-01392-5
16. Hashmi, E., Yayilgan, S. Y., Yamin, M. M., Ali, S., Abomhara, M. :Advancing Fake News Detection: Hybrid Deep Learning with FastText and Explainable AI. IEEE Access. 11, 4462-44480 (2024). **DOI:** 10.1109/ACCESS.2024.3381038
17. Perez, J. M., Luque, F. M., Zayat, D., Kondratzky, M., Moro, A., Serrati, P.S., Zajac, J., Miguel, P., Debandi, N., Gravano, A., Cotik, V. :Assessing the Impact of Contextual Information in Hate Speech Detection. IEEE Access. 11, 30575-30590 (2023). **DOI:** 10.1109/ACCESS.2023.3258973
18. Khan, M. M., Shahzad, K., Malik, M. K. :Hate Speech Detection in Roman Urdu. ACM Transactions on Asian and Low-Resource Language Information Processing. 20 (1), 1-19 (2020). https://doi.org/10.48550/arXiv.2108.02830
19. Bilal, M., Khan, A., Jan, S., Musa, S.: Context-Aware Deep Learning Model for Detection of Roman Urdu Hate Speech on Social Media Platform. IEEE Access. 10, 121133-121151 (2022). **DOI:** 10.1109/ACCESS.2022.3216375
20. Valle-Cano, G. D., Quijano-Sánchez, L., Liberatore, F., Gomez, J., : SocialHaterBERT: A dichotomous approach for automatically detecting hate speech on Twitter through textual analysis and user profiles. Elsevier, Expert Systems with Applications. 216, 1-17 (2023). https://doi.org/10.1016/j.eswa.2022.119446
21. García-Díaz, J.A., Jiménez-Zafra, S.M., García-Cumbreras, M. A., Valencia-García, R. : Evaluating feature combination strategies for hate-speech detection in Spanish using linguistic features and transformers. Springer. Complex and Intelligent Systems. 9, 2893-2914 (2023). DOI:10.1007/s40747-022-00693-x
22. Ayo, F. E., Folorunso, O., Ibharalu, F. T., Osinuga, I. A., Abayomi-Alli, A.: A probabilistic clustering model for hate speech classification in twitter. Elsevier. Expert Systems with Applications. 173, 1-21 (2021). https://doi.org/10.1016/j.eswa.2021.114762
23. García-Díaz, J.A., Canovas-García, M., Colomo-Palacios, R., Valencia-García, R.: Detecting misogyny in Spanish tweets. An approach based on linguistics features and word embeddings. Elsevier. Future Generation Computer Systems. 114, 506-518 (2021). https://doi.org/10.1016/j.future.2020.08.032
24. Agarwal, S. & Chowdary, C.R.: Combating hate speech using an adaptive ensemble learning model with a case study on COVID-19. Elsevier. Expert Systems with Applications. 185, 1-9 (2021). https://doi.org/10.1016/j.eswa.2021.115632
25. Modha, S., Majumder, P., Mandl, T., Mandalia, C. :Detecting and visualizing hate speech in social media: A cyber-Watchdog for surveillance. Elsevier. Expert Systems with Applications. 161, 1-11 (2020). DOI:10.1016/j.eswa.2020.113725
26. Plaza-del-Arco, F. M., Molina-Gonzalez, M. D., L. Urena-Lopez, A., Martín-Valdivia, M.T.:Comparing pre-trained language models for Spanish hate speech detection. Elsevier, Expert Systems with Applications, Volume 166. 1-10 (2021). https://doi.org/10.1016/j.eswa.2020.114120
27. Mossie, Z. & Wang, J-H. : Vulnerable community identification using hate speech detection on social media. Elsevier. Information Processing and Management. 57 (3), 1-16 (2020). https://doi.org/10.1016/j.ipm.2019.102087
28. Trajano, D., Bordini, R., Vieira, R.: OLID-BR: Offensive Language Identification Dataset for Brazilian Portuguese. Springer, 1-25 (2021). https://doi.org/10.1007/s10579-023-09657-0
29. Alsafari, S., Sadaoui, S., Mouhoub, M.:Hate and offensive speech detection on Arabic social media. Elsevier. Online Social Networks and Media. 19, 1-15 (2020). https://doi.org/10.1016/j.osnem.2020.100096
30. Motwakel, A., Al-onazi, B.B., Alzahrani, J.S., Alazwari, S., Othman, M., Zamani, A. S., Yaseen, I., Abdelmageed, A. A.: Improved Ant Lion Optimizer with Deep Learning Driven Arabic Hate Speech Detection. Computer Systems Science and Engineering. 46(3), 3321-3338 (2023). https://doi.org/10.32604/csse.2023.033901
31. Ashiq, W., Kanwal, S., Rafique, A., Waqas, M., Khurshaid, T., Montero, E.C., Alonso, A.B., Ashraf, I. Roman urdu hate speech detection using hybrid machine learning models and hyperparameter optimization. Scientific Reports, 14(1):1-22 (2024). doi: 10.1038/s41598-024-79106-7.

32. Hashmi, E., Yayilgan, S.Y., Abomhara, M.: Metalinguist: enhancing hate speech detection with cross-lingual meta-learning. Complex & Intelligent Systems. 11, 1-17 (2025). https://doi.org/10.1007/s40747-025-01808-w