Attention Mechanism-Based CNN-LSTM for Abusive Comments Detection and Classification in Social Media Text

BalaAnand Muthu First[1], Kogilavani Shanmugavadive[2], Veerappampalayam Easwaramoorthy Sathishkumar[3], Muthukumaran Maruthappa[4], Malliga Subramanian[5], Rajermani Thinakaran[6]
Dept. of Computer Science & Engineering, Tagore Institute of Engineering and Technology, Deviyakurichi, Attur, Salem, India[1]
Dept. of Artificial Intelligence, Kongu Engineering College, Perundurai, Erode, Tamil Nadu, India[2]
Dept. of Computing and Information Systems, Sunway University, Selangor, Malaysia[3, 4]
Dept. of Computer Science and Engineering, Kongu Engineering College, Perundurai, Erode, Tamil Nadu, India[5]
Faculty of Data Science and Information Technology, INTI International University, Negeri Sembilan, Malaysia[6]

***Abstract*—Human contact with one another through social networks, blogs, forums, and online news portals and communication has dramatically increased in recent years. People use these platforms to express their feelings, but sometimes hateful comments are also spread. When abusive language is used in online comments to attack individuals such as celebrities, politicians, and products, as well as groups of people associated with a given country, age, or religion, cyberbullying begins. Due to the ever-growing number of messages, it is challenging to manually recognize these abusive comments on social media platforms. This research work concentrates on a novel attention mechanism-based hybrid Convolutional Neural Network - Long Short Term Memory (CNN-LSTM) model to detect abusive comments by getting more contextual information from individual sentences. The proposed attention mechanism-based hybrid CNN-LSTM model is compared with various models on the dataset provided by the shared task on Abusive Comment Detection in Tamil – ACL 2022 which contains 9 class labels such as Misandry, Counter-speech, Xenophobia, Misogyny, Hope-speech, Homophobia, Transphobic, Not-Tamil and None-of-the-above. We obtained an accuracy of 67.14%, 68.92%, 65.35% and 68.75% on Naïve Bayes, Support Vector Machine, Logistic Regression and Random Forest respectively. Furthermore, we applied the same dataset to deep learning models like Convolutional Neural Networks (CNN), Long Short Term Memory (LSTM), Bidirectional-Long Short Term Memory (Bi-LSTM) and obtained an accuracy of 70.28%, 71.67% and 69.45%, respectively. To obtain more contextual information semantically a novel attention mechanism is applied to the hybrid CNN-LSTM model and obtained an accuracy of 75.98% which is an improvement over all the developed models as a process innovation.**

Keywords*—Attention mechanism; hybrid CNN-LSTM model; machine learning model; deep learning model; abusive comments detection

I. Introduction

The rise in web and social media interactions leads to a massive amount of information. Due to the freedom to convey everyone's opinion, sometimes the content posted on Facebook, Twitter, and YouTube may be offensive in nature [1]. Chakravarthi et al., [2] looked into the methods for recognizing several forms of abusive content, including aggressiveness, cyberbullying, hate speech, offensive language, abusive remarks, and abusive comments. Zampieri et al., [3] talked about automating the technique for detecting offensive language. The algorithms are trained using postings that have had the presence of any abusive or objectionable content noted. The entire problem was modeled as a supervised learning problem. In general, the Offensive Language Identification problem can be broadly categorized into Aggression Identification, Abusive Comments Detection, Hate Speech Identification, Offensive Language Identification and Toxic Comments Identification. The Aggression and Hate Speech Identification System classifies the given comments into Non-Aggressive, Covertly Aggressive and Overtly Aggressive. In the Abusive Comments Detection System, the class labels are Xenophobia, Misogyny, Hope-Speech, Homophobia, Transphobic, Not-Tamil, None-of-the-above categories. Toxic comments can be classified into Toxic, Severe Toxic, Obscene, Threat, Insult and Identity Hate categories.

Abuse is the act of making remarks that are hurtful to a particular person or group of people. A phrase that uses vulgar or harsh language in a discourse is referred to as abusive language either in oral form or in text form. The lack of eye contact among users of various social media platforms allows them to speak on the topic fearlessly. Therefore, it is necessary to automatically ban, discourage, or restrict users whose actions are hostile. Online abuse has contributed to issues including low self-esteem, despair, harassment, and in very extreme situations, death. Because abusive content can be communicated in a variety of ways, identifying and handling such comments is crucial and difficult. It is nearly impossible to manually find and remove abusive remarks from a large internet comment stream. Additionally, very few research has been done to identify abusive language in Dravidian languages like Tamil. The goal of this research project is to find instances of abusive language in Tamil-language YouTube comments. At the comment level, each post is tagged with nine different class labels. The data set was taken from the ACL 2022 shared task as in [4] which contains YouTube comments and Twitter posts written in Tamil language. The dataset description along with class labels are shown in Table I.

TABLE I. DATASET DESCRIPTION

Dataset	Comments	Labels
Training	2000	Misandry Counter-speech Xenophobia, Misogyny Hope-speech Homophobia Transphobic Not-Tamil None-of-the-above
Validation	240	
Testing	561	

The term "misandry" refers to established bias towards men. The term "counter-speech" refers to a strategy that presents an alternate story in place of offensive speech to combat hate speech or disinformation. The term "misogyny" refers to inherent bias against women. YouTube comments and posts with the class name "hope-speech" provide encouragement, assurance, advice, inspiration, and insight. The term "xenophobia" refers to a strong dislike of foreigners. The term "homophobia" refers to an antipathy, prejudice, or fear of homosexuals or homosexuality. The term "transphobia" is used to describe a broad spectrum of unfavorable attitudes, sentiments, or behaviors toward transgender persons. Sample comments in Tamil and their respective classes are represented in Table II.

TABLE II. COMMENTS AND CLASSES

Comments	Classes
பேச்சை எதிர்பார்த்தேன். நல்ல விளக்கம் அருமை. நியாயமான முறையில் பதிவு செய்துள்ளீர்கள். நீண்ட காலம் வாழ்வதற்கு வாழ்த்துக்கள். I was looking forward to your speech. Good description Awesome. You have registered reasonably. Congratulationson living a long life.	Hope-speech
இது பெண்மையை பயம் முறுத்தும் புரிதலற்ற பேச்சு.... This is an incomprehensible talk that scares women	Counter Speech
ராஜா நீ நாயி Raja you dog	Homophobia
கரு.நீ பெரியார் கருவா ? Are u egg of Periyar?	Misandry
சொத்து என்பதே இப்போது வந்ததுதான் அதற்கு முன்னமே ஆண்களுக்கு மட்டும் தான் சொத்து Property is what it is now and before that it was only for men	Misogyny
தமிழ் நாடு உனக்கு சோறு போடுது Tamil Nadu will give you rice	None of the above
பிச்சை பையன் எச்ச ராஜா The remnant king of the begging boy	Xenophobia
Seaman is a scumbag. Even a counselor is jealous. Waste talking fellow	Not-Tamil
அலி...நீ உழைக்கிற சாதியா Transgender …you are a working caste	Transphobic

Initial pre-processing steps like tokenization, emoji removal and label encoding are carried out. Significant features are extracted by utilizing Term Frequency-Inverse Document Frequency (TF-IDF), Count Vectorizer methods and fed into machine learning models such as Naïve Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LR) and Random Forest (RF). Word embedding is a technique used by state-of-the-art deep learning models like as Convolutional Neural Networks (CNN), Long Short Term Memory (LSTM) and Bidirectional-Long Short Term Memory (Bi-LSTM) to incorporate the context or high-level meaning of each word present in the comments. Although CNN has been shown to be effective at categorizing text, its classification performance is typically hampered by the frequent omission of essential long-distance sequential data, particularly in sentences with negation and semantic transition. Similar to this, the LSTM model is capable of capturing contextual data, particularly the meaning of long text data. However, this approach is unable to concentrate on the text's most crucial passages. In this work, a hybrid deep learning framework based on an attention mechanism and a combination of CNN and LSTM is presented to identify and categorize abusive remarks in social media text. To precisely identify the abusive comments, the ACL2022 shared task dataset which contains comments in Tamil language sentences was selected as the input of the neural network. CNN layers are used in the proposed hybrid CNN-LSTM model to capture features that specify the word's local information in its context. The text is then modeled with attention signals and sequence learning is performed using LSTM layers. Finally, the input from earlier neural networks is combined and offensive remarks are detected using fully linked networks. The impact of applying feature extraction and word embedding techniques to various learned models is also examined in this research work.

All of these cutting-edge models are compared to the suggested innovative attention mechanism-based hybrid CNN-LSTM model's performance. The models must categorize the comments into one of the nine classes when the test data is given. The objectives of this research work are 1) Abusive comments detection from YouTube comments in Tamil using machine learning and deep learning models; and 2) Embedding an attention mechanism in a hybrid CNN-LSTM model.

The research work that is currently available and the tools created for abusive comment detection are described in Section II. The proposed system models are mentioned in Section III. The performance evaluation with models' comparison is represented in Section IV and Section V ends with contributions and prospects for future direction.

II. LITERATURE REVIEW

Devlin et al., [5] described an approach of detecting emotions in social media comments in various languages. The

authors employed Neural Network (NN) layers and adoptive parameters to produce different models. Sanh et al. [6] show that using information distillation, a Bidirectional Encoder Representation from Transformers (BERT) model can be made 40% smaller while preserving 97% of its language knowledge.

Vinay et al., [7] classified the comments as personal attacks and not personal attacks. They employed LSTM with word embedding, CNN with word embedding, and CNN with character embedding. Out of all the models, CNN with character-level embedding achieves the best results. A hybrid method that applies sentiment analysis to machine learning approaches was developed by Hasan et al., [8]. In their study, supervised techniques like NB and SVM were in the investigation of strategic positions.

Huang et al., [9] discovered that hierarchical LSTMs leave extensive context modeling, by allowing them to enhance sentiment categorization significantly. They used LSTMs in particular because they tackle the aforementioned vanishing gradient problem. Zhao et al. [10] used CNNs to test various feature embeddings, ranging from character to sentence level. The researchers expected that the word embedding layer would be critical for sentiment analysis since short texts have a limited quantity of contextual information. They discovered that character-level embedding worked better than other embeddings on one dataset and behaved similarly well on the other using the two datasets they utilized.

Pinkesh et al., [11] classified a tweet as racist, sexist or neither. They conduct extensive research with several deep learning architectures to learn semantic embedding. They tested on a 16K annotated tweets benchmark dataset and revealed that deep learning approaches beat machine learning algorithms. Rotaru et al., [12] discovered how essential phonetic symbols affect people's emotions. Their model's phonetic transcription feature maps show that their model performs fantastically in texts that are overly dense and filled with incorrect words. Additionally, they think that including this grammatical precedence in the sub-word design will improve the model's performance.

Sharif et al., [13] recognized aggressive texts in Bengali using a weighted ensemble strategy and developed methods such as m-BERT, Distill-BERT, Bangla-BERT, and XLM-R. These mentioned models work better than regular methods. Shreelakshmi et al., [14] worked with messages that include code-mixed data from Hindi and English that originate from multilingual consumers. Many previous methodologies omitted data from these low-resource languages.

Sentiment polarity of blended text was utilized by Sevda et al., [15] to classify sentences into those with three emotional meanings. According to this paper, shared parameters are used to translate words. They also showed a simple pre-processing technique based on clustering for gathering variations in code-mixed text. Raut et al. [16] investigated methods for extracting data from Twitter tweets. They also studied supervised learning techniques that could be utilized to identify textual tweet polarity, including SVM for Document Classification. Their conclusion suggests that SVM can recognize text features like a big feature set, or a sparse instance vector.

Cambria et al., [17] discussed new avenues for performing sentiment analysis. During the initial stage itself, a set of emotion class labels were derived from 5,553 tweets using inductive coding. Malliga et al., [18] presented an offensive language detection system using an adapter and transformer-based model. The sentimental analysis of Twitter data was carried out by Barbosa et al., [19] utilizing machine learning and deep learning techniques. They gathered test data from Twitter and examined tweets using syntactic components such as symbols, re-tweets, emoticons, tags, links, punctuation, and exclamation points. They have employed the polarity classifier and the subjectivity classifier as their two primary classifier types. A high-quality collection of Hindi-English code-switch data with 15,744 YouTube comments was produced by Ray et al., [20]. They looked at the collection's growth, the findings of trend analysis using the sample as a guide, and the consistency of the annotators.

Mathur et al., [21] developed a Hindi-English code flipped dataset into Abusive hate speech and non-offensive. A multi-label hostility detection dataset was provided by Bhardwaj et al. [22] and was divided into five categories using the BERT algorithm: fake, hate, offensive, defamatory, and non-hostile. Mulki et al., [23] classified the dataset of Twitter into abusive, hate and normal classes using the Naïve Bayes algorithm. Hassan et al., [24] used SVM and CNN+Bi-LSTM to classify offensive Arabic texts and achieved better results for the CNN+Bi-LSTM approach.

Kogilavani et al., [25] carried out sentiment analysis on Tamil code-mixed data. From the dataset, significant features are identified through a hybrid model. Leite et al. [26] reported a toxic language dataset made up of 21000 tweets that were categorized using the BERT algorithm into GBTQ+phobia, racism, insult, xenophobia, obscenity, misogyny, and non-toxic language. The brief literature review helped to realize the need to identify abusive comments that are present in the Tamil language.

Ashraf et al., [27] presented a method to detect the contextual information present in YouTube comments based on topics such as politics, religion and others. The context is identified with the help of linguistic information present in the comments. Jahan et al., [28] employed a transliterated code-mixed dataset for abusive comments detection in Bangla-English data. The result shows that the transliterated dataset does not improve the accuracy of the system.

III. Materials and Methods

Fig. 1 represents the workflow of the methodologies used in this proposed work. As the first step, pre-processing such as tokenization, emoji removal and label encoding are carried out for the dataset.

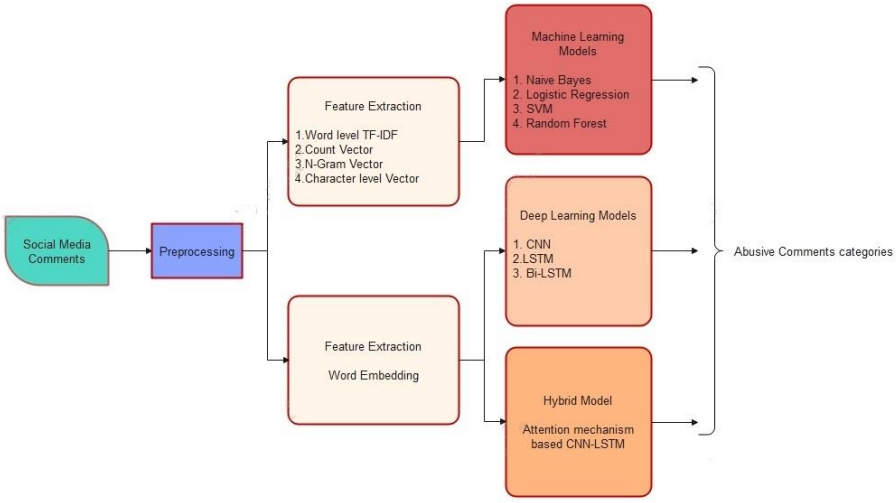


Fig. 1. Proposed system work flow.

Significant features are extracted from Tamil sentences by utilizing TF-IDF and Count Vectorizer methods. The resultant word vectors are fed into machine learning models such as NB, SVM, LR and RF. Word embedding is a technique used by deep learning models like as CNN, LSTM, and Bi-LSTM to incorporate the context or high-level meaning of each word present in the comments. Traditional approaches represent text in terms of TF, IDF, TF-IDF with sparse features. Recently deep learning models have been adopted with word embedding to extract contextual information from the text data. However, these deep learning models do not consider sequential information among the sentences. LSTM is used to capture the semantics of lengthy texts. However, this methodology is unable to focus on important textual passages. When used with text data, this restriction lowers the accuracy of deep learning models. It is necessary to concentrate on particular passages of the text, which are very similar to how people read. The goal of this research work is to suggest a hybrid CNN-LSTM model that is attention-based and can identify offensive comments. The suggested method uses a convolution layer as its first step in capturing attention signals, which reflect the local information of each word in its context. The text is then modeled using LSTM using attention signals. A word with a larger attention weight typically refers to more valuable information.

A. Feature Extraction

The process of extracting features from text input and generating numerical representations of them so that learning models can use them to produce predictions is known as feature extraction. The words or concepts that appear most frequently and have the fewest relevant data start to take precedence over the ones that appear less frequently when a vector is constructed from a text using word frequency. To avoid rating terms in papers purely on their frequency in a single text, as suggested by Sajeetha et al., [29] word frequency must be rescaled.

1) TF-IDF: With TF-IDF, words are given varying weights based on how important they are to the document. It calls attention to a particular problem that, while uncommon in our corpus, is quite crucial. Textual data has therefore been converted into real-valued vectors using TF-IDF. How frequently a word appears in a publication is referred to as term frequency. It might be compared to the likelihood of a term appearing in a document. It establishes the proportion between the frequency of a word (wi) and the overall number of words in the comment (cj). The inverse document frequency statistic can be used to assess how frequently or infrequently a phrase appears across all the documents in a corpus. By dividing the entire number of comments in the dataset by the number of comments that contain the term t, it is possible to get the logarithm of the overall term.

2) Count vector: The Count Vectorizer is used to turn a text into a vector based on how frequently each word appears in the text as a whole. This is helpful when organizing each word in a vector and working with numerous texts of this type. Each distinct word is represented by a column in a matrix created by Count Vectorizer, and each sample of text from a remark is represented by a row. The value of each field indicates how many phrases were contained in the text sample.

3) N-Gram vector: The continuous or neighboring sequence of words in a sentence is represented using the N-gram vector. After performing tokenization and stop words removal process, N-Grams from the text can be represented in vector format.

4) Character level vector: To generate a character level vector, the input sentences are decomposed into a sequence of characters including special characters. To extract morphological information from the text, this kind of character level vector is used.

5) Word embedding: Word embedding, according to Sajeetha et al., [30] is a kind of word vector that makes it possible to represent words with similar meanings. Individual words are often represented in word embedding as real-valued vectors in a predetermined vector space. Since each word is translated into a single vector and its values are learned like that of a neural network, the method is typically categorized under deep learning models. The main concept of the approach is to represent each word with dense sparse representations. A real-

valued vector with a sizable number of dimensions—often tens or hundreds—represents each word.

B. Machine Learning Models

1) Naive Bayes (NB): A group of classification methods known as NB classifiers are based on Bayes' Theorem, which states that all pairings of feature pairs are conditionally independent given the value of the class variable. Mathematically, Bayes theorem can be stated as in (1).

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)} \quad (1)$$

Where y is the class variable and X is the input feature vector. NB learners and classifiers may be extremely fast in comparison to more complex algorithms. Each distribution can be estimated as a one-dimensional distribution individually since the class conditional feature distributions are separated. As a result, problems brought on by the curse of dimensionality are lessened.

2) Support Vector Machine (SVM): Both classification and regression problems can be resolved using the supervised machine learning method known as SVM. Even if categorization more closely reflects the data, regression has downsides. The SVM technique categorizes n-dimensional space by trying to find the best judgment boundary or line so that the following data points can be quickly assigned to the right category. The best choices are known as hyperplanes. When there are only two input features, the hyperplane is essentially a line. When there are three input features, the hyperplane transforms into a two-dimensional plane. Once a hyperplane is generated, it is used to make predictions or classifications by using hypothesis function *h* as in (2) where *x* is input feature, *w* is weight and *b* is bias.

$$h(x_i) = \begin{cases} +1 \; if \; w.x + b \geq 0 \\ -1 \; if \; w.x + b < 0 \end{cases} \quad (2)$$

3) Logistic Regression (LR): LR is a technique for estimating the probability of a discrete result given an input variable. LR models often yield binary outputs, such as true or false, yes or no, and so forth. It is a simple classification model that produces top-notch results and has linearly separable classes. It is a typical categorization technique used in the industrial sector. When performing classification tasks, the analytical technique of logistic regression is useful for assessing if a new sample fits into a particular category. The following equation as in (3) represents logistic regression where *x* is the input feature, *y* is the output variable, b_0 is bias and b_1 is the coefficient for input.

$$\mathrm{y} = \frac{e^{(b0+b1x)}}{1+e^{(b0+b1x)}}) \quad (3)$$

4) Random Forest (RF): The supervised machine learning algorithm RF is frequently employed to address classification and regression problems. As its name suggests, RF is made up of a sizable number of connected decision trees. The class with the highest score is chosen as our model's projection from among the predictions made by each tree in the random forest. It produces decision trees from a variety of data using regression and majority voting, respectively. One of the key characteristics of the RF algorithm was its capacity to handle data sets with both continuous and categorical variables, as in classification and regression issues. Categorization issues, yield improved outcomes.

C. Deep Learning Models

1) Convolutional Neural Network (CNN): CNN is a deep learning technique built on the Multi-Layer Perceptron (MLP). Features are retrieved and used for classification tasks in conventional algorithms. In CNN, higher concepts are created by a series of convolutional layers once these features are automatically extracted. The three main characteristics of CNN are convolution, pooling, and fully connected layers. The most fundamental part of a CNN is the convolution layer, which consists of a number of separate filters commonly referred to as masks or kernels. The input and filters are convolved to produce either an activation map or a feature map. The pooling layer comes next, and its function is to minimize the input's spatial dimensions. Its objective is to simplify the representation of intricate layers. The pooling layer's output is flattened into a single vector and sent as the fully connected layer's input. Fully linked layers, which are utilized to carry out a particular task, such as classification, are the vectors produced by a multilayer perceptron's several convolutions and pooling processes.

2) Long Short Term Memory (LSTM): LSTMs are one type of Recurrent Neural Network (RNN) that stores intermediate outcomes in LSTM networks. When linking old knowledge to fresh information in a traditional RNN, the problem of vanishing gradients usually arises. There is only one layer in the repeating module of a typical RNN. LSTM holds information in a gated cell. A cell can accumulate information and can read and write into its memory. By default, the LSTM has a propensity to retain information for a long time. The Sequential 3 model has a 20000 feature maximum, 128 embedding dimensions, a 40 sequential length, and a 196 LSTM out. The Spatial dropout1d layer receives the output of the Embedding layer, the LSTM layer receives the output of the LSTM layer, and the dense layer receives the output of the dense layer. Nine nodes in the dense layer function as sigmoid activation nodes. The categorical cross entropy loss function and Adam optimizer are used to train the LSTM.

3) Bidirectional-Long Short Term Memory (Bi-LSTM): The process of building a NN that can store sequence information both future to past and past to future is known as Bi-LSTM. Unlike a standard LSTM, a Bi-LSTM has input those flows in both directions. With a standard LSTM, we may make input flow in one way, either backward or forward. However, there is a way to maintain both the present and the future while allowing information to move both ways. In the suggested system, the values for maximum features, embedding size, sequential length, and lstm out are all set to 20,000. The spatial dropout1d layer receives the output of the embedding

layer, the dense layer receives the output of the bidirectional layer as input, and the bidirectional layer receives the output of the dense layer. Nine nodes in the dense layer employ an activation function known as a softmax. The Nadam optimizer and the categorical cross-entropy loss function are used to train the Bi-LSTM.

D. Attention Mechanism-Based Hybrid Model

By fusing CNN with LSTM, the suggested approach creates a novel hybrid model based on an attention mechanism. Through an attention mechanism, neural networks facilitate the dynamic selection of pertinent features from text data. Either the raw input or its higher level representation can receive it directly. This attention mechanism's calculations involve weighing the order of text pieces and giving more weight to pertinent text information. Mainly attention mechanism is used to capture the context between sentences so that if the sentence does not have direct abusive statements also, will be detected by this mechanism. The proposed attention mechanism-based CNN-LSTM hybrid model structure is represented in Fig. 2.

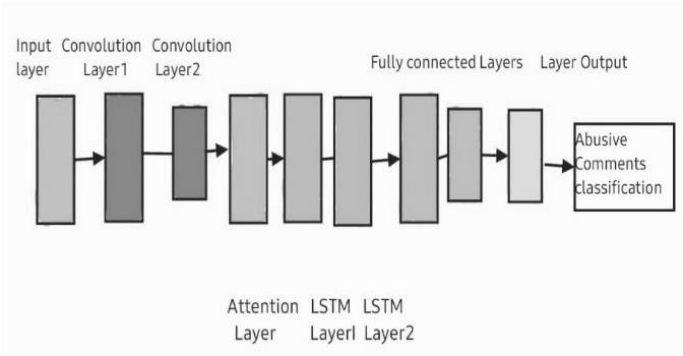


Fig. 2. Attention mechanism-based CNN-LSTM hybrid model.

To develop a CNN+LSTM hybrid model, first CNN layers are added, then attention mechanisms, LSTM layers, and Deep Layers on the output. The first LSTM layer of the LSTM+CNN model receives word vectors for each token in the sequences. This architecture can be seen as defining both the CNN Model for feature extraction and the LSTM Model for feature interpretation over time steps. The Sequential 7 model's maximum features input is set to 20,000, the embedding dimension to 128, the sequential length to 40, and the LSTM out to 196. With a pooling size of 2, a kernel size of 3, and activation functions of ReLU and sigmoid, Conv1d has 128 output filters. Text sequences are created with Conv1d. The output of the embedding layer is provided as input to the conv1d layer, the output of the conv1d layer is input to the MaxPooling1d layer, the output of the MaxPooling1d layer is input to the LSTM layer, and the output of the LSTM layer is input to the dense layer. In the dense layer, nine nodes have sigmoid activation functions. CNN+LSTM is trained using the Adam optimizer and the categorical cross entropy loss function.

IV. Performance Evaluation

Initially, the dataset is applied to word level TF-IDF, count vector, N-Gram vector and character level vector. The proposed system chooses the models based on accuracy metrics instead of loss because the model with minimum loss may not be the model with the best metric. The accuracy obtained by all these feature extraction techniques that are applied to various machine learning models is represented in Table III. The results show that, for all the models, the best accuracy is produced by the character level vector which extracts morphological features from the given text input. Among the entire machine learning models SVM model produces the highest accuracy of 0.6892.

A. Experiments

1) Precision: Precision is the proportion of positive class predictions that fall into that category. Precision values are calculated using as in (4) where TP is True Positive and FP is False Positive.

$$\text{Precision} = TP / (TP + FP) \quad (4)$$

2) Recall: Recall is a metric that indicates how well the model properly detects True Positives. The recall is determined as in (5) where FN is False Negative.

$$\text{Recall} = TP / (TP + FN) \quad (5)$$

3) F1-Score: The weighted mean of Precision and Recall is the F1 Score. This score is generated using an equation as in (6) and takes into consideration both false positives and false negatives.

$$\text{F1 Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision}) \quad (6)$$

The precision, recall, F1-Classification reports of machine learning models such as NB, SVM, LR and RF are presented in Table IV. The result shows that the highest precision of 0.74 is obtained for counter-speech class abusive comments by the NB model. LR produces a precision value of 0.80 for the Not-Tamil type of abusive comments. A precision value of 1.00 is produced by SVM for the Misandry type and by RF for the Not-Tamil type.

TABLE III. Accuracy Comparison of various Machine Learning Models based on Feature Extraction

Models	Word level TF-IDF	Count Vector	N-Gram Vector	Character Level Vector
Naïve Bayes	0.6312	0.6417	0.6315	0.6714
Support Vector Machine	0.6291	0.6010	0.6718	0.6892
Logistic Regression	0.6342	0.6214	0.6171	0.6535
Random Forest	0.6593	0.6432	0.6154	0.6875

TABLE IV. CLASSIFICATION REPORT OF MACHINE LEARNING MODELS

Class Label	Naïve Bayes			Support Vector Machine			Logistic Regression			Random Forest			Support
	P	R	F1-Score	P	R	F1-Score	P	R	F1-Score	P	R	F1-Score	
None-of-the-above	0.00	0.00	0.00	0.25	0.03	0.05	0.00	0.00	0.00	0.00	0.00	0.00	36
Misogyny	0.50	0.12	0.20	0.33	0.12	0.18	0.33	0.12	0.18	0.33	0.12	0.18	8
Misandry	0.00	0.00	0.00	1.00	0.09	0.17	0.67	0.18	0.29	0.75	0.27	0.40	11
Counter-speech	0.74	0.36	0.48	0.69	0.57	0.62	0.73	0.61	0.66	0.64	0.56	0.59	104
Xenophobia	0.00	0.00	0.00	0.40	0.08	0.14	0.50	0.21	0.29	0.50	0.08	0.14	24
Homophobia	0.67	0.98	0.80	0.72	0.97	0.83	0.74	0.96	0.83	0.73	0.96	0.83	346
Hope-Speech	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1
Transphobic	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2
Not-Tamil	0.00	0.00	0.00	0.00	0.00	0.00	0.80	0.14	0.24	1.00	0.07	0.13	29

4) Accuracy: The accuracy of classification models is one of the factors to consider while evaluating them. Informally, accuracy refers to the suggested model's proportion of correct predictions, which is determined as in (7).

$$\text{Accuracy} = (TP+FP) / \text{Total Predictions} \quad (7)$$

Fig. 3 represents the accuracy obtained by NB, LR, RF and SVM learning models. The result shows that the highest accuracy of 68.92 is obtained by SVM for the given dataset.

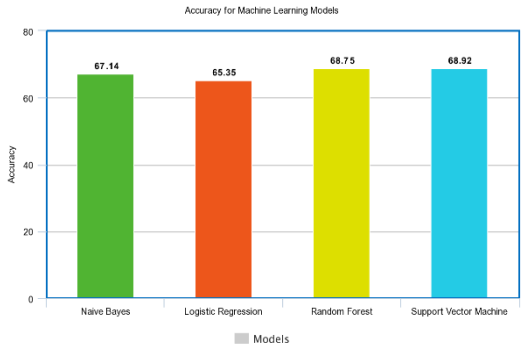


Fig. 3. Performance evaluation of machine learning models based on accuracy.

Fig. 4 denotes the accuracy of CNN, LSTM, Bi-LSTM deep learning models after extracting significant features using word embedding. The result shows that among the three mentioned models, the LSTM deep learning model performs better with an accuracy of 71.67%.

From the deep learning models, the highest accuracy obtained from two models such as CNN and LSTM are combined to generate the hybrid model. The attention mechanism is implemented in between Convolutional layers and LSTM layers in a hybrid model. The highest accuracy machine learning model such as SVM and deep learning model such as LSTM is compared with the hybrid model in Fig. 5. The result shows that the best accuracy of 75.98 is obtained by the attention mechanism-based hybrid CNN-LSTM model.

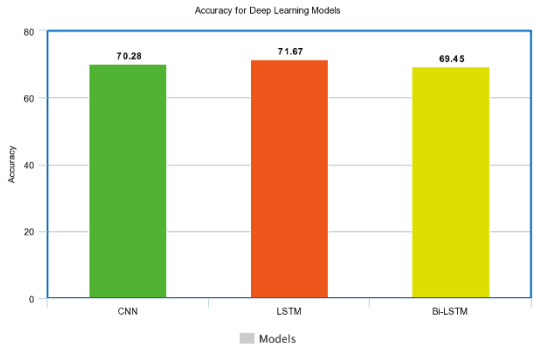


Fig. 4. Performance evaluation of deep learning models based on accuracy.

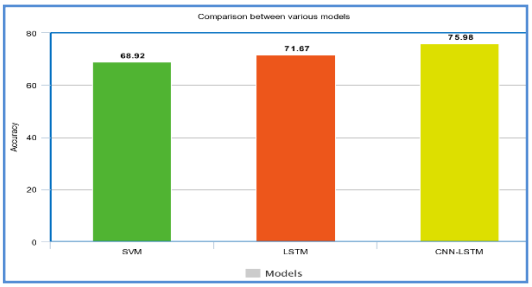
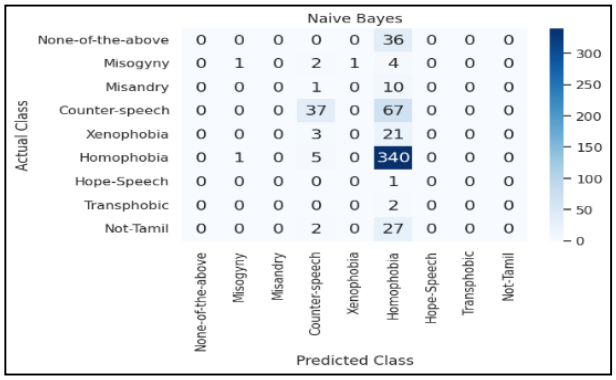
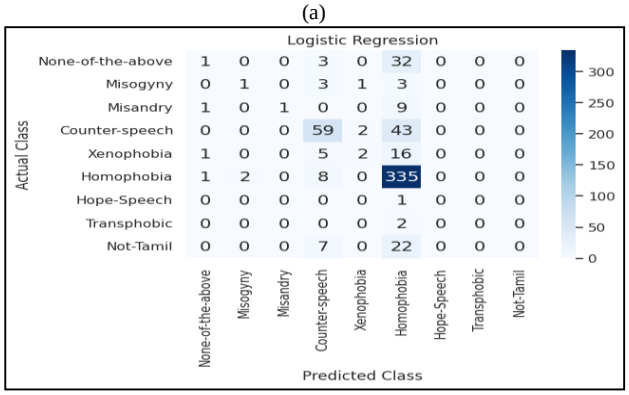


Fig. 5. Comparison of performance evaluation of machine learning, deep learning and attention mechanism-based hybrid CNN-LSTM model.

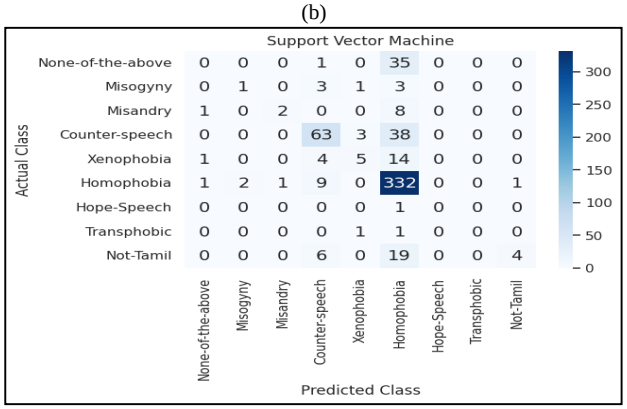
5) Confusion matrix: The confusion matrix is a useful tool for assessing classifiers' abilities to discriminate between data from different classes. In the confusion matrix form, TP and TN are displayed when the classifier is functioning well. FP and FN represent occasions where the classifier is incorrect. A confusion matrix is a table containing the given dataset's 9 classifications and dimensions of 9 x 9. Good accuracy is gained by employing the values along the confusion matrix's diagonal. The confusion matrix for machine learning algorithms like NB, SVM, LR and RF is shown in Fig. 6(a) to 6(d).



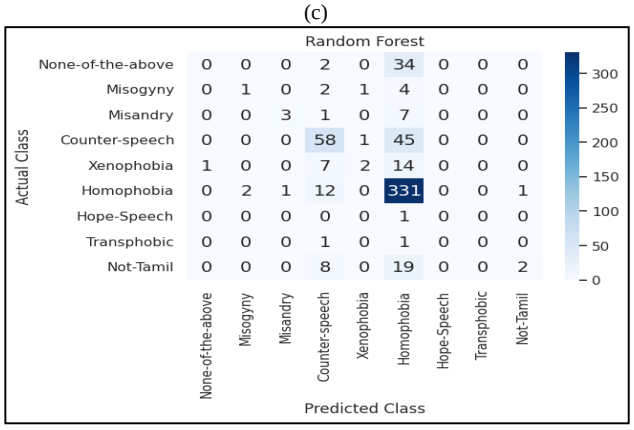
(a)



(b)



(c)

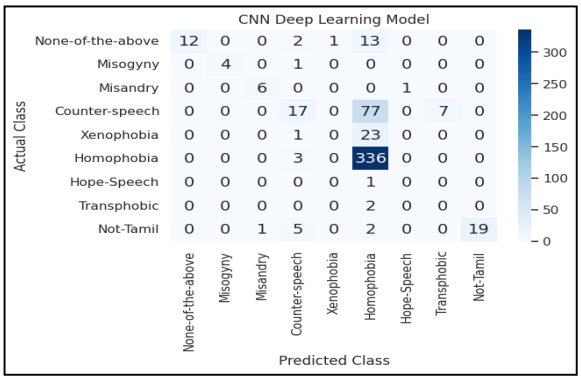


(d)

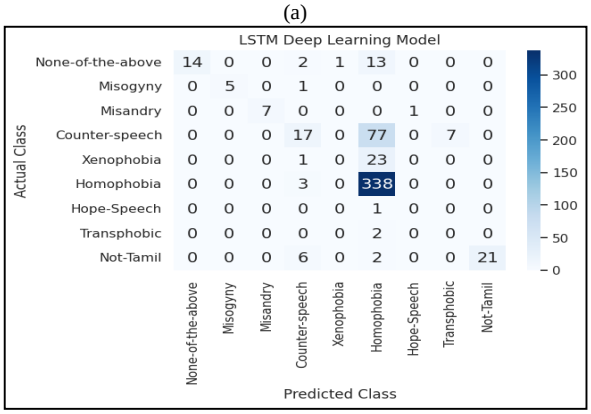
Fig. 6. (a) Confusion matrix - Naïve bayes, (b) Confusion matrix - Logistic regression, (c) Confusion matrix - Support vector machine, (d) Confusion matrix - random forest.

From the confusion matrices in Fig. 6(a) to 6(d), it is understood that out of 346 samples in Homophobia type of abusive comments, the Naïve Bayes models correctly classifies 340 comments, SVM classifies 332 comments, LR classifier 335 comments and RF classifies 331 comments. Fig. 7(a) to 7(c) represents, the confusion matrix obtained by CNN, LSTM and Bi-LSTM deep learning models.

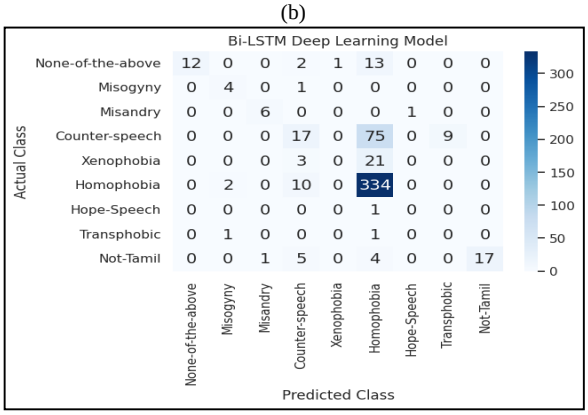
The confusion matrix in Fig. 7(a) to 7(c) shows that out of three deep learning models, the LSTM model outperforms by classifying 402 comments correctly out of 561 comments. Fig. 8 represents the confusion matrix of the proposed attention mechanism-based hybrid CNN-LSTM model. The result shows that 421 comments were correctly classified.



(a)



(b)



(c)

Fig. 7. (a) Confusion matrix - CNN Model, (b) Confusion matrix LSTM Model, (c) Confusion matrix - Bi-LSTM Model.

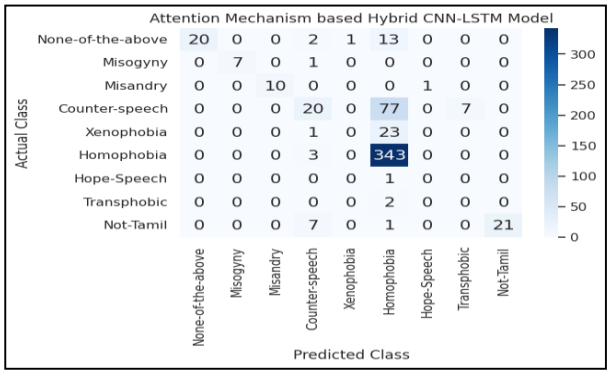


Fig. 8. Confusion matrix - Hybrid CNN-LSTM model.

6) Matthews correlation coefficient: The range of the Matthews Correlation Coefficient (MCC) is [-1,1]. The anticipated values will closely match the actual classification if the value is close to 1, which indicates that the prediction was quite accurate. There is no association between our variables if MCC is equal to 0. Values near -1 imply an inverse relationship between the true and anticipated classes. MCC is calculated using the formula specified as in Eq. (8).

$$MCC = \frac{(TP*TN-FP*FN)}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}} \quad (8)$$

7) Cohen's Kappa: The concept of calculating the agreement between the Predicted and the True Labels—which are viewed as two random categorical variables—is the foundation of Cohen's Kappa. By creating a confusion matrix and computing the marginal rows and marginal column distributions, it is possible to compare two category variables. Kappa value is calculated using the equation as in (9) where p_o is observed probability and p_e is expected or predicted probability which is calculated from the confusion matrix.

$$Kappa = \left(\frac{p_o-p_e}{1-p_e}\right) \quad (9)$$

8) Error rate: The error rate is calculated by subtracting accuracy from 1 and it is calculated using equation as in (10). For example, the sample sentence "இது பெண்மையை பயம் முறுத்தும் புரிதலற்ற பேச்சு....(This is an incomprehensible talk that scares women ....) is the abusive comment of counter speech category. However, due to the context related to women, it is wrongly classified as a misogyny type.

$$Error\ rate = 1 - accuracy \quad (10)$$

The MCC, kappa value and error rate of all the machine learning models are represented in Table V. Out of all the mentioned machine learning models, SVM obtained the highest MCC value of 0.69 and Kappa value of 0.67. Based on accuracy, the error rate is calculated and it is less for both SVM and Random Forest.

TABLE V. MCC, KAPPA AND ERROR RATE OF MACHINE LEARNING MODELS

Machine Learning Model	MCC	Kappa	Error rate
Naïve Bayes	0.66	0.58	0.32
Support Vector Machine	0.69	0.67	0.31
Logistic Regression	0.64	0.64	0.34
Random Forest	0.67	0.64	0.31

V. CONCLUSION AND FUTURE WORK

In this research work, a hybrid CNN-LSTM model with an attention mechanism has been effectively presented to detect and categorize offensive comments into subcategories including misandry, counter-speech, xenophobia, misogyny, hope-speech, homophobia, transphobia, not-in-Tamil, and none of the above. This research work initially uses a variety of machine learning models, including NB, LR, SVM and RF along with a variety of feature extraction methods, including word level TF IDF, count vector, N-Gram vector, and character level vector. SVM with character level vector based feature extraction method achieves the best accuracy of 68.92%. Word embedding-based deep learning models including CNN, LSTM, and Bi-LSTM were applied to the dataset in order to further increase accuracy. The experimental findings demonstrated that significant features are automatically retrieved, and individual CNN, LSTM models achieve an accuracy of 70.28% and 71.69%, respectively. This result motivates us to develop a hybrid model by combining CNN with LSTM. The suggested work utilizes a novel attention mechanism between CNN and LSTM layers which is called the hybrid CNN-LSTM model to extract more pertinent features from the dataset and achieve superior accuracy of 75.98%. The proposed work detects abusive comments from code-mixed data by applying state-of-the-art machine learning, and deep learning models. Further, the accuracy is improved by applying the proposed attention mechanism-based hybrid CNN-LSTM model. In the future, transfer learning methodology for the identification and classification of abusive remarks may be utilized to improve accuracy further.

REFERENCES

[1] K. Subramaniam, W. Wider, A. Vasudevan, N. Khan, and A. Kohli, "Transitions of value creation from traditional media to social media architecture," Online Journal of Communication and Media Technologies, 13(4), e202356, 2023.

[2] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, N. Jose, S. Suryawanshi, E. Sherly, and J. P. McCrae, "Dravidiancodemix: Sentiment analysis and offensive language identification dataset for Dravidian languages in code-mixed text," Language Resources and Evaluation, 56(3), pp. 765-806, 2022.

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

[4] R. Priyadharshini, B. R. Chakravarthi, S. C. Navaneethakrishnan, T. Durairaj, M. Subramanian, K. Shanmugavadivel, and P. K. Kumaresan, "Findings of the shared task on Abusive Comment Detection in Tamil," In Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages. Association for Computational Linguistics, May 2022.

[5] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[6] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," arXiv preprint arXiv:1910.01108, 2019.

[7] V. Singh, A. Varshney, S. S. Akhtar, D. Vijay, and M. Shrivastava, "Aggression detection on social media text using deep neural networks," In Proceedings of the 2nd workshop on abusive language online (ALW2), pp. 43-50, October, 2018.

[8] A. Hasan, S. Moin, A. Karim, and S. Shamshirband, "Machine learning-based sentiment analysis for twitter accounts," Mathematical and computational applications, 23(1), pp. 11, 2018.

[9] M. Huang, Y. Cao, and C. Dong, "Modeling rich contexts for sentiment classification with lstm," arXiv preprint arXiv:1605.01478, 2016.

[10] J. Zhao, S. Mudgal, and Y. Liang, "Generalizing word embeddings using a bag of subwords," arXiv preprint arXiv:1809.04259, 2018.

[11] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," In Proceedings of the 26th international conference on World Wide Web companion, pp. 759-760, April, 2017.

[12] A. S. Rotaru, and G. Vigliocco, "Constructing semantic models from words, images, and emojis," Cognitive science, 44(4), pp12830, 2020.

[13] O. Sharif, E. Hossain, and M. M. Hoque, "M-bad: A multilabel dataset for detecting aggressive texts and their targets," In Proceedings of the Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situations, pp. 75-85, May 2022.

[14] K. Sreelakshmi, B. Premjith, and K. P. Soman, "Detection of hate speech text in Hindi-English code-mixed data," Procedia Computer Science, 171, pp.737-744, 2020.

[15] S. Mammadli, S. Huseynov, H. Alkaramov, U. Jafarli, U. Suleymanov, and S. Rustamov, "Sentiment polarity detection in Azerbaijani social news articles," In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019), pp. 703-710, September 2019.

[16] V. Shweta Raut, and M. M. Nashipudimath, "Performance Analysis of Feature Selection for Twitter Sentiment Analysis: Classification approach," International Journal of Advanced Research in Computer and Communication Engineering, pp67-74, 2018.

[17] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," IEEE Intelligent systems, 28(2), pp. 15-21, 2013.

[18] M. Subramanian, R. Ponnusamy, S. Benhur, K. Shanmugavadivel, A. Ganesan, D. Ravi, and B. R. Chakravarthi, "Offensive language detection in Tamil YouTube comments by adapters and cross-domain knowledge transfer," Computer Speech & Language, 76, pp. 101404, 2022.

[19] L. Barbosa, and J. Feng, "Robust sentiment detection on twitter from biased and noisy data," In Coling 2010: Posters, pp. 36-44, August 2010.

[20] D. Sitaram, S. Murthy, D. Ray, D. Sharma, and K. Dhar, "Sentiment analysis of mixed language employing Hindi-English code switching," In 2015 International Conference on Machine Learning and Cybernetics (ICMLC), vol. 1, pp. 271-276, IEEE, July D. 2015.

[21] P. Mathur, R. Shah, R. Sawhney, and Mahata, "Detecting offensive tweets in hindi-english code-switched language," In Proceedings Of The Sixth International Workshop On Natural Language Processing For Social Media, pp. 18-26, July 2018.

[22] M. Bhardwaj, M. S. Akhtar, A. Ekbal, A. Das, and T. Chakraborty, "Hostility detection dataset in Hindi," arXiv preprint arXiv:2011.03588, 2020.

[23] H. Mulki, H. Haddad, C. B. Ali, and H. Alshabani, "L-hsab: A levantine twitter dataset for hate speech and abusive language," In Proceedings of the Third Workshop on Abusive Language Online, pp. 111-118, August 2019.

[24] A. Hasan, S. Moin, A. Karim, and S. Shamshirband, "Machine learning-based sentiment analysis for twitter accounts," Mathematical and Computational Applications, 23(1), pp. 11, 2018.

[25] K. Shanmugavadivel, S. H. Sampath, P. Nandhakumar, P. Mahalingam, M. Subramanian, P. K. Kumaresan, and R. Priyadharshini, "An analysis of machine learning models for sentiment analysis of Tamil code-mixed data," Computer Speech & Language, 76, pp. 101407, 2022.

[26] J. A. Leite, D. F. Silva, K. Bontcheva, and C. Scarton, "Toxic language detection in social media for Brazilian Portuguese: New dataset and multilingual analysis," arXiv preprint arXiv:2010.04543, 2020.

[27] N. Ashraf, A. Zubiaga, and A. Gelbukh, "Abusive language detection in youtube comments leveraging replies as conversational context," PeerJ Computer Science, 7, pp. 742, 2021.

[28] M. Jahan, I. Ahamed, M. R. Bishwas, and S. Shatabda, "Abusive comments detection in bangla-english code-mixed and transliterated text," In 2019 2nd International Conference on Innovation in Engineering and Technology (ICIET), pp. 1-6, IEEE, December 2019.

[29] S. Thavareesan, and S. Mahesan, "Sentiment analysis in Tamil texts: A study on machine learning techniques and feature representation," In 2019 14th Conference on industrial and information systems (ICIIS), pp. 320-325, IEEE, December 2019.

[30] S. Thavareesan, and S. Mahesan, "Word embedding-based part of speech tagging in Tamil texts," In 2020 IEEE 15th International conference on industrial and information systems (ICIIS), pp. 478-482, IEEE, November 2020.