IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 December 2023, accepted 14 January 2024, date of publication 5 February 2024, date of current version 9 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3358811

RESEARCH ARTICLE

Detection of Hate Speech and Offensive Language CodeMix Text in Dravidian Languages Using Cost-Sensitive Learning Approach

K. SREELAKSHMI[1], B. PREMJITH[1], BHARATHI RAJA CHAKRAVARTHI[2], AND K. P. SOMAN[1]
[1]Amrita School of Artificial Intelligence, Amrita Vishwa Vidyapeetham, Coimbatore 641112, India
[2]School of Computer Science, University of Galway, Galway, H91 TK33 Ireland

Corresponding author: B. Premjith (b_premjith@cb.amrita.edu)

ABSTRACT Recently, the emergence of social media has opened the way for online harassment in the form of hate speech and offensive language. An automated approach is needed to detect hate and offensive content from social media, which is indispensable. This task is challenging in the case of social media posts or comments in low-resourced CodeMix languages. This paper investigates the efficacy of various multilingual transformer-based embedding models with machine learning classifiers for detecting hate speech and offensive language (HOS) content in social media posts in CodeMix Dravidian languages that belong to the low-resource language group. Experiments were conducted on six sets of openly available datasets in Kannada-English, Malayalam-English and Tamil-English languages. The objective is to identify a single pre-trained embedding model that commonly works well for HOS tasks in the above mentioned languages. For this, a comprehensive study of various multilingual transformer embedding models, such as BERT, DistilBERT, LaBSE, MuRIL, XLM, IndicBERT, and FNET for HOS detection was conducted. Our experiments revealed that MuRIL pre-trained embedding performed consistently well for all six datasets using Support Vector Machine (SVM) with Radial Basis Function (RBF) kernel. In a set of experiments conducted on six datasets, the highest accuracy results for each dataset are as follows: DravidianLangTech 2021 achieved 96% accuracy for Malayalam, 72% accuracy for Tamil, and 66% accuracy for Kannada. For HASOC 2021 Tamil, the accuracy reached 76%, and for HASOC 2021 Malayalam, it reached 68%. Additionally, HASOC 2020 demonstrated an accuracy of 92% for Malayalam. Moreover, we performed an in-depth error analysis and a comparative study, presenting a tabulated summary of our work compared to other top-performing studies. In addition, we employed a cost-sensitive learning approach to address the class imbalance problem in the dataset, in which minority classes get higher classification weights than the majority classes. The weights were initialized and fine-tuned to obtain the best balance between all the classes. The results showed that incorporating the cost-sensitive learning strategy avoided class bias in the trained model. In addition to the aforementioned points, a significant contribution of our research presented in this paper is introducing a novel annotated test set for Malayalam-English CodeMix. This new dataset serves as an extension to our existing data, known as the Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages (HASOC) 2021 Malayalam-English dataset.

INDEX TERMS Natural language processing, CodeMix, hate speech, offensive language, bidirectional encoder representations from transformers, language-agnostic BERT sentence embedding, multilingual representations for Indian languages, machine learning, IndicBERT.

The associate editor coordinating the review of this manuscript and approving it for publication was Maria Chiara Caschera.

I. INTRODUCTION

The emergence of social media platforms has helped people communicate across borders and opine easily [29], [36].

It has not only paved the way for networking and information exchange but also resulted in the proliferation of hate and offensive content. Under International Human Rights Law, there is no universal definition of hate and offensive speech (HOS). However, in [10] and [42], it is stated that HOS is the advocacy or incitement in any form, defamation, hatred, or vilification of a person or group, along with insulting or abusing anyone on the grounds of their colour, race, religion, caste, gender, sex, or financial status, which is pervasive on social media platforms. In most cases, hate speech contains much offensive content, which soberly damages our society by leading to civil war, undermining vulnerable groups, and even spoiling new research inventions like chatbots, which learn from the inputs it experience [46].

Automatic detection of hate and offensive content from social media posts/comments is very relevant in recent times due to the adverse effect of the increasing amount of HOS content on social media. Automatic HOS detection is a relevant area of work mainly because the spread of HOS in social media can lead to the usage of offensive words by children, can result in religious or community conflicts and even lead to the failure of conversational chatbots due to a lack of knowledge for the bots to identify offensive words. Various research has been conducted to detect HOS using Natural Language Processing (NLP) approaches. However, the complexity of the language and the nature of the social media comments and posts caused the ML/DL models to detect the HOS effectively. The challenge increases in the case of CodeMix Dravidian languages due to the usage of vocabulary from multiple languages, the usage of different language scripts, and non-standard grammar, spelling and abbreviation variation. These factors affected the development of a gold-standard annotated corpus, hence the research in detecting HOS contents [13]. However, shared tasks for offensive language identification from Kannada, Malayalam and Tamil by Chakravarthi et al. [13], [14] paved the way for more research on CodeMix Dravidian languages. HOS detection began by using manually extracted features namely punctuation count, emoticon count, negation words, lexicon words [11], [33], [78], followed by machine learning classifiers. Further research used character N-grams, word N-grams, term frequency-inverse document frequency (TF-IDF), Bag of Words vectors (BOW) features [45], [60]. The advent of neural network-based embedding algorithms motivated the researchers to use pre-trained domain-specific embedding such as Word2vec [55], fastText [56], GloVe [62] for generating word vectors followed by Machine learning/Deep learning classifiers [30], [72], [73] for detecting HOS contents. The availability of multilingual pre-trained models and their efficient performance on various NLP tasks increased the use of transformer models for HOS detection [27], [38], [49], [50], [63].

This paper investigates the performance of various multilingual transformer embedding models with Machine Learning classifiers for detecting HOS contents from social media text in Kannada-English, Malayalam-English and Tamil-English languages. We conducted experiments using six different corpora collected from various shared tasks, containing the Dravidian script and native language words written using the Roman script (CodeMix). Unlike various neural network embedding models trained on a single language, multilingual models have the advantage of a single model that can handle the characteristics of multiple languages together, which motivates us to focus on generating sentence embedding using multilingual models rather than an embedding model trained exclusively for one language. Transformer-based embedding models are trained on both subword-level and word-level information, which also helps the models learn representations for Out-of-Vocabulary (OOV) words. Furthermore, it is observed that the class imbalance problem in the data has not yet been addressed in the literature, which could affect the performance of the ML/DL models. Therefore, a cost-sensitive learning approach was incorporated in the classifiers, making them not biased towards any particular class. In addition, an extension of the Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages (HASOC) 2021 Malayalam-English dataset is proposed in this paper, which is an annotated CodeMix Malayalam-English HOS corpus.

In summary, the main contributions of this paper are as follows:

- A comprehensive study of various multilingual transformer embedding models for detecting HOS from CodeMix social media texts in Dravidian languages.
- A single multilingual embedding model that performs well on all CodeMix datasets of the three languages was identified.
- Cost-sensitive learning approach was used to deal with the class imbalance problem in the dataset to avoid the class-bias.
- An annotated HOS detection CodeMix Malayalam-English dataset was developed
- Error analysis and performance comparison of our work with the state-of-the-art approach.

The rest of the paper is organised as follows: Section II highlights the previous work done in this area; Section III gives the details of the dataset used and the experiments and the results are provided in Section IV. Section V gives a thorough discussion of the error analysis conducted and the paper is concluded in section VI.

II. RELATED WORK

Research in HOS detection and classification has advanced in the past decade. Attentiveness to this field has increased as the influence and user adoption of social media and social platforms have expanded. Research in HOS identification mainly focuses on two approaches: those based on hand-made features such as punctuation count, emoticon count, negation words, or lexicon features, and those on neural

network-based pre-trained embeddings such as fastText, GloVe and transformers [4].

Most of the HOS detection works on social media comments and posts are highly challenging due to their non-standard formats in spelling and grammar [28]. Generally, the comments and posts in Dravidian languages appear in CodeMix form [8]. Several works have recently been reported for detecting HOS from social media texts in the CodeMix Dravidian language. This section reviews the research on HOS detection in CodeMix social media texts.

Apart from English, there were also significant research contributions in HOS detection in European languages using machine learning [18], [19], [23], [39], [78]. Initial implementations used TF-IDF scores [57], BOW Vectors [44], [51], [52], [69], N-grams [26], [58], [78], meta-information such as user account information and network structure information [19], [59], [70] for representing the text. These features are fed to various machine-learning classifiers to detect HOS contents. The popularity of deep learning algorithms attracted interest in HOS detection due to their ability to automatically learn input representations which can be used for detecting HOS [1], [6], [32], [34], [61], [81]. The negative impact of the posted contents, their possible severe consequences and the lack of annotated data paved the way for many academic events and shared tasks on HOS detection. Some of the tasks are:

- The shared task on aggression identification included in the First Workshop on Trolling, Aggression and Cyberbullying [47].
- The first, second and third editions of the Workshop on Abusive Language [65].
- The first and second edition of GermEval Shared Task on the Identification of Offensive Language [74].
- The MEX-A3T track at IberLEF 2019 on Authorship and Aggressiveness Analysis [3].
- The PolEval 2019 shared Task 6 on Automatic Cyberbullying Detection in Polish Twitter [64].
- The first edition of the HASOC track at FIRE 2019 on HOS and Offensive Content Identification in Indo-European Languages [54].
- The SemEval shared subtask 5 on the detection of HOS against immigrants and women (HatEval) [9] and subtask 6 on Identifying and categorizing offensive language in social media (OffensEval) [80].

A. DRAVIDIAN CodeMix

Recently, there has been an increase in research focus on HOS detection in CodeMix Dravidian languages, particularly Kannada-English, Malayalam-English and Tamil-English. However, the diverse nature of the grammar, polysemous words and unavailability of the tools and annotated data limited the research in Dravidian languages [2], [16]. Shared tasks on offensive language identification in Kannada, Malayalam Tamil [13], [14] and contribution of annotated data by researchers [15], [24], [40] opened the way for more research on Dravidian languages.

Most of the papers found in the literature regarding the HOS detection in Dravidian languages were related to the teams participating in shared tasks on offensive language identification in Kannada, Malayalam and Tamil DravidianLangTech 2021 and HASOC-Dravidian-CodeMix shared tasks [12], [48]. Most of these works used transformer models because of the availability of multilingual pre-trained models, their capability to capture context information, and the ease of fine-tuning. The top-performing teams used various deep learning and transformer methodologies. Saha et al. [67] used various models, namely the XLM-RoBERTa-large model, a fusion model with a Bidirectional Encoder Representations from Transformers- Convolutional Neural Network (BERT-CNN) where a single classification head was trained on the concatenated embedding from different BERT and CNN models. The BERT models were initialized with fine-tuned weights. The CNN models were trained on skip-gram word vectors using fastText. They also experimented using MuRIL and IndicBERT pre-trained specifically on low resource languages. Balouchzahi et al. [7] used a COOLI Ensemble model, which takes the CountVectors of words and character sequence as features and classifies them using a voting classifier with three estimators Multi-Layer Perceptron, extreme Gradient Boosting, Logistic Regression. Tula et al. [75] used an ensemble model of DistilmBERT and ULMFiT and inverse weighting and focal loss strategies to solve the class imbalance issue. Vasantharajan and Thayasivam [76] used pre-trained multilingual transformer models and used Negative Log Likelihood (NLL) Loss with class weights and self-adjusting dice loss to resolve the class imbalance issue. A few researchers have worked on the comparison of pre-trained Embedding to identify Hate Speech [5]. The paper compares BERT [25], XLNet [79], DistilBERT [68], RoBERTA [21] and Ensemble model for classification. Hande et al. trained multi-task learning models for Sentiment analysis and offensive language [35]. The other works in this area are on Offensive Language Identification using Pseudo-labeling [37] and an approach that uses selective translation and transliteration techniques to reap better results by fine-tuning and ensembling multilingual transformer [76]. Sivalingam and Thavareesan [71] used Support Vector Machine, random forest, k- Nearest Neighbour and Naive Bayes classifiers with chi-square, BOW, TF-IDF feature representation techniques. Apart from these, there are works on pre-trained models specific to Indian Languages. Raj Dabre et al. and team developed a multilingual, sequence-to-sequence pre-trained model IndicBART on 11 Indian languages and English. The model is smaller in size than mBERT but gives comparative results for Neural Machine Translation and summarization [22]. From the literature review, we observed that various pre-trained models are explored on the different datasets, including class imbalanced data.

TABLE 1. Detailed dataset description. The dataset statistics including the train-test count, the details about the dataset source, the scripts involved and the labels.

Dataset Name	Training set	Test set	Labels	Script
HASOC 2020 Malayalam [53]	3200	400	Not_offensive , Offensive	Roman+Devanagari
HASOC 2021 Malayalam [17]	5000	999	HOF, NOT	Roman
HASOC 2021 Tamil [17]	4000	808	NOT, OFF	Roman
DravidianLangTech 2021 Malayalam [14]	18009	2001	Not_offensive, Offen-sive_Targeted_Insult_Group, Of-fensive_Targeted_Insult_Individual, Offensive_Untargeted, notMalayalam	Roman+Devanagari
DravidianLangTech 2021 Tamil [14]	36366	4075	Not_offensive, Offen-sive_Targeted_Insult_Group, Of-fensive_Targeted_Insult_Individual, Offensive_Targeted_Insult_Other, Offensive_Untargeted, notTamil	Roman+Devanagari
DravidianLangTech 2021 Kannada [14]	6994	778	Not_offensive, Offen-sive_Targeted_Insult_group, Of-fensive_Targeted_insult_Individual, Offensive_Targeted_Insult_Other, Offensive_Untargeted, not-Kannada	Roman+Devanagari

Our literature review revealed that no recognized pre-trained model is available for detecting HOS in CodeMix Dravidian languages. It inspired us to investigate different multilingual transformer embeddings for HOS detection on six datasets from three language pairs: Kannada-English, Malayalam-English, and Tamil-English. Moreover, we aimed to find a single multilingual embedding that performs effectively on all CodeMix datasets across the three languages. Furthermore, it was noted that most of the existing data exhibits an imbalance. It motivated us to utilize a cost-sensitive learning strategy to address the issue. In addition, we noticed a scarcity of annotated corpus, which inspired us to create a novel annotated Malayalam-English HOS corpora.

III. DATASET DESCRIPTION

We conducted our experiments on six datasets belonging to three Dravidian CodeMix languages, Kannada-English, Malayalam-English and Tamil-English, collected from the shared task on Offensive Language Identification in Kannada, Malayalam and Tamil, HASOC track at FIRE 2020: Hate Speech and Offensive Content Identification in Indo-European Languages and HASOC track at FIRE 2021:Offensive Language Identification for Dravidian Languages in CodeMix Text. Detailed dataset description is given in Table 1. The classwise dissemination of the datasets is specified in Table 2.

A. IN-HOUSE TESTSET

The lack of annotated data is still a hindrance to research in the area of HOS. So, we have contributed Malayalam-English annotated HOS detection data for shared tasks HASOC 2020 Malayalam and HASOC 2021 Malayalam. As a part of this work, we collected 1000 Malayalam-English CodeMix YouTube comments to validate our top-performing model. We removed all the comments from the collected comments that were not Malayalam-English CodeMix. These comments were then used to create a dataset for the offensive language classification task. This test set is an extension of the HASOC 2021 Malayalam dataset [17]. In this work, we annotated the data into two classes - hate and Non-hate, with each class having 670 and 330 test sentences, respectively. One example from each class is given below.

- **Non-hate**: A comment is annotated as Non-hate if it doesn't contain any offensive words or is not sarcastically insulting a person or a group.
  Text: ***Cbz nalla quality ulla bike aanu..Silencer ilaki pokunnathu oru safety mechanism aanu..Impact kurakkan..***
  Translation: *Cbz is a good quality bike. Silencer getting removed is a part of safety mechanism to reduce the impact.*
- **Hate**: Comments with offensive or abusive language which is used to insult people or group are labelled as Hate.
  Text: ***Yep…ennittum telegram inne free aayi kaanunna oolakal***
  Translation: *Yep.. Still seeing telegram free fools.*

1) ANNOTATION

The dataset annotation was done by three annotators who were proficient in Malayalam and English. The annotation was based on whole sentence meaning, usage of words, sarcasm in the speech and emoticons. All three annotators separately annotated the whole test set. The annotated dataset is in the form of an excel sheet, with the first column having the texts and the second column having the tags.

TABLE 2. Class-wise distribution of the datasets.

S.No	Dataset Name	Labels	Distribution
1	HASOC 2020 Malayalam	Not_offensive	2961
		Offensive	639
2	HASOC 2021 Malayalam	HOF	2786
		NOT	3213
3	HASOC 2021 Tamil	OFF	2400
		NOT	2408
4	DravidianLangTech 2021 Malayalam	Not_offensive	17697
		Offensive_Targeted_Insult_Group	176
		Offensive_Targeted_Insult_Individual	290
		Offensive_Untargeted	240
		not-Malayalam	1607
5	DravidianLangTech 2021 Tamil	Not_offensive	29482
		Offensive_Targeted_Insult_Group	2704
		Offensive_Targeted_Insult_Individual	2708
		Offensive_Targeted_Insult_Other	528
		Offensive_Untargeted	3342
		not-Tamil	1677
6	DravidianLangTech 2021 Kannada	Not_offensive	4397
		Offensive_Targeted_Insult_Group	418
		Offensive_Targeted_Insult_Individual	628
		Offensive_Targeted_Insult_Other	153
		Offensive_Untargeted	278
		not-Kannada	1898

2) INTER ANNOTATOR AGREEMENT

Inter Annotator Agreement measures the accord between the annotators during data annotation. All three annotators follow the same set of guidelines for annotation. After the annotation by the three annotators, the tag for each comment is chosen by the majority voting scheme. That is, for a comment, the label that the majority of the annotators choose is assigned to it. The annotation is validated using Cohen's kappa score, [20], a statistical measure for assessing the reliability of agreement between a fixed number of raters over 2. Cohen's Kappa score for the proposed corpus is 0.8967. Equation 1 shows the formula used to calculate Cohen's Kappa score, where $\bar{P}$ is the observed agreement, and $\bar{Pe}$ is the expected agreement.

$$k = \frac{\bar{P} - \bar{Pe}}{1 - \bar{Pe}} \tag{1}$$

Since, the number of annotators is three the expected agreement $\bar{Pe} = 1/3$.

IV. EXPERIMENTS AND RESULTS

This section explores the capabilities and the limitations of the different transformer embedding and machine learning approaches for hate speech detection using the workflow given in Figure 2. We considered six sets of Indian CodeMix datasets which consist of a single sentence that either fall into two main categories, offensive and not-offensive, or into multiple classes offensive-targeted-insult-individual, offensive-targeted-insult-group, offensive-targeted-insult-other, offensive untargeted and not-in-intended-language. The work investigates various multilingual pre-trained transformer-based models to find the best fit for Dravidian CodeMix datasets of Kannada, Malayalam and Tamil. In this section, we talk about several multilingual pre-trained transformer-based models. We obtained the transformer embedding using https://www.sbert.net/. The transformer models, the experiments conducted, and the results are explained in detail in the algorithm 1 and workflow diagram 1.

As described in the dataset description section, our dataset consists of CodeMix and Indic script sentences with emoticons and punctuation. These emoticons and punctuation were not removed, and the dataset was not subjected to any other preprocessing as they can contribute to the emotion the sentence carries.

The first step in the workflow is to extract features. Since the data is diverse and multilingual, multilingual transformer models trained on a large Wikipedia dump with a sizeable vocabulary were used. We used the python framework SentenceTransformers to convert these sentences to numbers. Transformer models such as BERT, DistilBERT,

Algorithm 1 An Algorithm Explaining the Workflow Used for Building a HOS Detection Classifier. Features Are Extracted Using Multilingual Models and Machine Learning Classifier Is Built Using Training Set

[1]

Input: X = CodeMix social media text in the languages Kannada-English, Malayalam-English and Tamil-English Texts with labels.

Output: Trained machine learning models

Multilingual Transformer models M = {BERT, Distil-BERT, XLM-RoBERTa, LaBSE, MuRIL, FNet, IndicBERT }

Machine Learning Classifiers C={Random Forest,Linear Regression, Naive Bayes, K-Nearest neighbour, Decision Tree, Adaboost }

for *x in X* **do**
  Read each text
  for *m in M* **do**
    Generate embedding
    Assign class weights using the below given formula

$$\frac{N}{n \times b} \tag{2}$$

    where, N = Total Number of text in the dataset, n = Number of classes and b = Total number of occurrences of each class in the dataset
    for *c in C* **do**
      Train the machine learning classifier
      Save the model
    end
  end
end

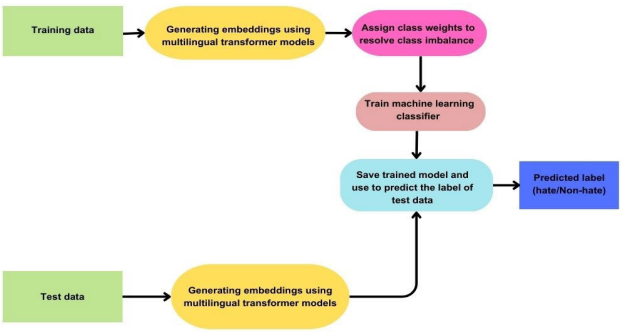

FIGURE 1. An illustration of the workflow. The dataset is divided into train and test sets and the embeddings of the data are obtained. machine learning classifier is trained and used to predict the labels.

XLM-RoBERTa, LaBSE, MuRIL, FNet and IndicBERT are used to extract embedding from the sentences.

A. TRANSFORMER MODELS

Most NLP tasks, for instance, Machine Translation (MT), Topic Classification, Named Entity Recognition (NER), etc., require capable models connecting the previous information to the present. For example, predicting the next word from the previous words in a sentence or present video frames from the previous frame, etc. Recurrent neural network (RNN) and Long short-term memory (LSTM) are such neural networks capable of remembering previous information. However, these models face the vanishing gradient problem whenever such long-term dependencies are involved. Since these are seq2seq models, they handle the sentences word by word or character by character forming a hindrance towards parallelization. Especially in the case of long sentences, these models tend to forget the information of the initial positions. These flaws of RNN and LSTM led to the introduction of the Transformer model [77]. Figure 2 gives the illustration of the transformer architecture.

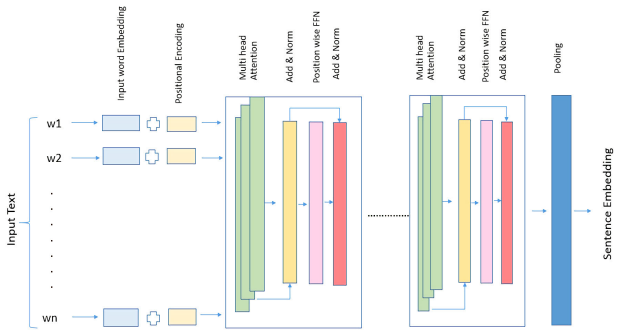

FIGURE 2. An illustration of the transformer architecture. It has 12 blocks each consisting of Multi head Attention, Add and Norm, point wise FFN and Add and Norm. The embedding of the input words w1, w2 are combined with the positional encoding and fed to the block.

The transformer is the first-ever model that entirely depends on self-attention to obtain the input representation. As illustrated in Figure 2. It consists of encoder-decoder stacks. The encoder consists of a multi-head self-attention layer and a feed-forward layer. In addition to those layers, the decoder consists of masked multi-head attention. Around each of these sub-layers of both encoder and decoder, a residual connection is employed, followed by normalization. Apart from this, the multi-head attention in the decoder takes the encoder output as well [77].

Attention mainly involves six steps.

1) Calculation of the vectors Query, Key, and Value. These vectors are calculated by multiplying the word embedding by three matrices we trained during the training process. The architecture is designed in such a way that the embedding vector is of length 512, and the Query, Key and Value vectors have a smaller dimension of 64.
2) Finding the self-attention score. This score is calculated by taking the dot product of the query vector of one word to the key vector of every other word. For a particular word in a sentence, this score gives how much focus has to be given to other words of the input sentence.

3) Divide each score with 8, which is the square root of the dimension of the key vector.
4) These values are passed through a softmax layer, which converts them to a positive value.
5) Multiply the softmax score with each value vector, which forms the weighted value vector.
6) Sum the weighted value vectors.

1) MULTI-HEAD ATTENTION

The model uses several attention layers parallelly, forming the multi-head attention. It is mainly used to attend all the positions in the previous layer. So when attention is calculated multiple times, we get multiple z matrices for a single sentence. These matrices are concatenated and then fed to the feed-forward network.

2) POSITIONAL ENCODINGS

The absence of recurrence or convolutional layers is overcome by the positional encodings. It feeds the information about the tokens' relative or absolute position, which takes care of the succession of sequence order in the model.

B. BERT

BERT is a case-sensitive transformer model pre-trained on a large unlabelled Wikipedia corpus of 104 languages using deep bidirectional representations [25]. BERT-base multilingual (mBERT-base) architecture has 12 layers, 768 hidden sizes and 12 attention heads, making a total of 177 million parameters. It has been trained on 11 NLP tasks with two main objectives:

1) Masked language modeling (MLM) BERT architecture enables the model to learn the bidirectional representation of the sentence, in contrast to other Language models, which are either trained from forward or backward, the sequential deep learning models, or autoregressive text generation models such as GPT. Even so, the word might unintentionally spot itself due to its bidirectional nature. To prevent this the model masks 15% of the input words and the objective is to predict the masked words.
2) Next-Sentence Prediction (NSP) Language models' failure to accurately represent the relationship between two sentences is one of their fundamental weaknesses. However, this is crucial for NLP tasks like question answering and interference with natural language. The task of Next Sentence Prediction is carried out to get around this. In this, two masked sentences are combined as pretraining inputs by the model. Sometimes they line up with sentences that are adjacent to one another in the original text, and other times they don't. The next step is for the model to determine if the two sentences are in order.

BERT can be fine-tuned on tasks such as sequence classification, token classification or question answering that use the whole sentence (potentially masked) to make decisions. We used the "bert-base-multilingual-cased" from hugging face for our experiments.

1) DistilBERT

DistilBERT is a smaller language representation model, a distilled version of the BERT base multilingual model. It can perform all the tasks a BERT model can do but at twice the speed of mBERT-base. It is a case-sensitive model with six layers, 768 dimensions and 12 attention heads which makes a total of 134 million parameters [68].

2) XLM-RoBERTa

XLM-R is a transformer-based masked multilingual language model trained on 100 languages. It was trained using more than two terabytes of filtered CommonCrawl data. It has significantly improved performance on various cross-lingual transfer tasks and outperformed the mBERT model. XLM-R has 24 layers and 16 attention heads, making a total of 550 million parameters [21].

3) LaBSE

LaBSE was proposed [31] to produce a language-agnostic sentence embedding by following the multilingual BERT model. It is trained on 109 languages.

4) MuRIL

Multilingual Representations for Indian Languages (MuRIL) is a transformer-based multilingual language model built for Indian languages. The model is trained using both translated and transliterated document pairs of 16 Indian languages and English. It uses BERT architecture which is pre-trained from scratch using datasets collected from Wikipedia, CommonCrawl, PMINDIA, and Dakshina. It has outperformed mBERT on a lot of downstream tasks [43].

5) IndicBERT

IndicBERT is a multilingual model trained based on the ALBERT model. It is trained on IndicCorp of 12 Indian languages, including Assamese, Bengali, English, Gujarati, Hindi, Kannada, Malayalam, Marathi, Oriya, Punjabi, Tamil, Telugu and evaluated on IndicGLUE. Compared to other multilingual models, namely IndicBERT mBERT, XLM-R IndicBERT has fewer parameters [41].

6) FNet

FNet is a transformer model pre-trained on an English dataset for masked language modeling and next sentence prediction. It is very similar to other transformer models except for using Fourier transforms instead of attention. It is trained on a massive corpus of English raw texts without labels. The model can be used for many downstream tasks such as classification.

The models mentioned above were chosen because they were trained on multilingual data. Among these models, LaBSE, MuRIL and IndicBERT are explicitly trained in

Indian languages but not on CodeMix text. The majority of our data is in Roman script. Hence we chose FNet, which is an English pre-trained model. In addition, FNet is smaller in size and hence computationally efficient. Our dataset being CodeMix with Roman and Indic script, we used them to get the embedding using the transformer models BERT, DistilBERT, XLM-RoBERTa, LaBSE, MuRIL and IndicBERT. At the same time, we transliterated the data to English before using FNet to extract the embedding.

C. DISCUSSION OF RESULTS

As shown in the workflow algorithm 1, the third step is to classify. For classification, we chose the following set of Machine learning classifiers. Random Forest, Linear Regression, Naive Bayes, K-Nearest neighbor, Decision Tree, Adaboost, SVM Rbf, SVM Linear, SVM Ploy.

In total, we conducted 63 experiments on six different datasets. The results of the experiments conducted are given in Tables 3, 4, 5, 6, 7, 8. We used the metrics Accuracy, Precision (macro), Recall (macro), F1-score (macro) and F1-score (weighted) to evaluate our models. One of the major issues with available data is the class imbalance. We resolved the dataset imbalance issue during the experiment using a cost-sensitive learning approach in which the class weights are computed using Equation 3.

$$w_{ci} = \frac{|X|}{n\,|c_i|} \tag{3}$$

where w_{ci} is the class weight for the class c_i, $|X|$ is the total number of data points in the corpus, n is the total number of classes, and $|c_i|$ is the total number of data points in the class c_i. This would assign a large weight to the minority classes and small weights to the majority classes, subsequently restricting the classifiers to bias towards the majority classes.

The code for the experiments is given in the GitHub repository.

On observing the results, we noted that for Malayalam DravidianLangTech data DistilBERT, MuRIL, LaBSE gave comparatively high accuracy and F1-score; for Tamil DravidianLangTech data DistilBERT, MuRIL gave comparatively high accuracy and F1-score and for Kannada DravidianLangTech data MuRIL gave high accuracy and F1-score. These three datasets had sentences in Roman as well as Indic scripts, and they also had other non-Indian language sentences. Out of these, for two datasets, DistilBERT was able to perform well because of its masking model, which grabbed the complete sentence meaning in addition to it being trained in more than 100 languages. For HASOC Malayalam 2021 and HASOC Tamil 2021 data, which are in pure Roman script and CodeMix, MuRIL gave the highest accuracy and F1-score. For HASOC Malayalam 2020, LaBSE and MuRIL gave comparatively high accuracy and F1-score. This data, having both Roman and Devanagari script LaBSE gave high performance due to its Dual encoder structure, which considers source text and targets text simultaneously as input. This way of training the LaBSE model on Indian scripts helped the model grab the meaning of CodeMix texts.

Compared to all the other models, MuRIL gave a consistent performance on all the datasets irrespective of language or script. We also obtained comparable results to the state-of-the-art models. This is due to the MuRIL model's pre-training on parallel and monolingual segments. The model was trained using monolingual data collected from Wikipedia and Common Crawl corpora for 17 Indian languages and parallel corpora obtained by translation and transliteration of the monolingual corpora mentioned above. MuRIL exploits the characteristics of the transformer model to generate the embeddings of words in Indian languages.

D. COMPARATIVE STUDY WITH THE STATE-OF-THE-ART RESULTS

In this subsection, we compare the results obtained by our approach with the state-of-the-art models. The comparison details are given in Table 9. The state-of-the-art results are taken from the overview papers of each shared task. Out of the four datasets, our approach showed comparable results with the state-of-the-art models in three datasets without any data translation. In DravidianLangTech 2021 Malayalam data, our MuRIL embedding+Machine Learning classifier got the same F1-score compared to the top-performing ULMFiT model of the state-of-the-art model. However, our other embedding BERT+Machine Leaning, DistilBERT+Machine Leaning, and XLM-R+Machine Leaning had better performance than the BERT, DistilBERT, XLM-R based classifiers of the state-of-the-art work. For DravidianLangTech 2021 Tamil data, the MuRIL embedding+Machine Learning classifier and XLM-R embedding+Machine Learning classifier performed better than their MuRIL and XLM-R classifiers. For DravidianLangTech 2021 Kannada data, our MuRIL embedding+Machine Learning classifier crossed the results obtained by the MuRIL classifier of the state-of-the-art work. In addition, for HASOC 2020 Malayalam data, our approach got extremely high performance compared to the state-of-the-art model. This indicates that BERT-based embedding with Machine Learning classifiers has the upper hand compared to BERT-based classifiers in HOS from Dravidian language tasks.

E. VALIDATION ON IN-HOUSE DATA

The performance of the collected test set was validated using the MuRIL models trained on the three binary datasets. The details of this are given in Table 10.

V. ERROR ANALYSIS

This section provides a comprehensive error analysis of the experimental outputs, exploring the misclassification errors encountered by the model during the classification task across diverse datasets in different languages. The evaluation is based on a weighted F1-score for performance comparison.

TABLE 3. Results for Kannada DravidianLangTech dataset.

Embedding	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	
					Macro	Weighted
BERT	SVM RBF	67.9	44.5	38.9	40.2	66
	SVM poly	67.2	44.1	38.7	39.8	65
	SVM linear	59.5	35.7	37	36.2	60
	Random Forest	64.5	39.8	26.2	26.6	58
	Logistic Regression	54.1	38.9	47.3	39.9	57
	KNN	64.8	46.1	37.7	39.2	63
	Decision Tree	50.1	26.6	25.9	26.2	50
	Adaboost	60.8	29.9	28.3	26.6	56
	Naive Bayes	49.1	32.3	40.3	31.9	50
DistilBERT	SVM RBF	67.6	42.2	39.2	40	66
	SVM poly	67.1	42	38.7	39.6	65
	SVM linear	61.4	39.1	42.2	40.2	63
	Random Forest	66.7	39.9	28.2	28.5	60
	Logistic Regression	53.3	37.9	42.2	37.7	57
	KNN	65.7	41.5	37.3	38.6	64
	Decision Tree	53	30.8	30.7	30.6	53
	Adaboost	63.2	31.3	30.3	29	59
	Naive Bayes	49.2	33.2	39.4	31.5	48
LaBSE	SVM RBF	68.3	51.1	29.6	41.9	66
	SVM ploy	69.2	55.9	38.1	40.7	66
	SVM linear	51.5	37.6	44.4	38	55
	Random Forest	62.3	54.2	26.7	27.9	55
	Logistic Regression	49.7	37.8	45.4	38.1	53
	KNN	64.8	45.7	37.8	39.6	62
	Decision Tree	50	27.8	28.3	27.9	50
	Adaboost	55.1	30.6	30.9	30.2	53
	Naive Bayes	45.4	35.6	45.6	36.4	48
MuRIL	SVM RBF	65.2	42.1	47.7	43.8	66
	SVM poly	65.6	42.1	47.5	44	66
	SVM linear	52.6	39.9	47.3	39.2	54
	Random Forest	66.2	40.3	27.6	28	60
	Logistic Regression	50.5	34.2	40.9	33.7	50
	KNN	68.8	38.3	37.8	37.3	66
	Decision Tree	51.3	29.5	29.4	29.3	52
	Adaboost	63.8	34.7	32.9	32.1	60
	Naive Bayes	51.9	32.6	37.9	32.6	52
XLM	SVM RBF	65	40.6	37.2	38.4	63
	SVM poly	64.1	41.9	41.7	41.4	64
	SVM linear	59.5	36.4	38.6	37.2	60
	Random Forest	63.5	51.4	26.8	28.8	56
	Logistic Regression	51.9	36.5	41.7	37	55
	KNN	61.6	43.1	32.5	35	58
	Decision Tree	49.7	28.7	27.2	27.8	50
	Adaboost	60.2	31.5	33.1	31.8	58
	Naive Bayes	45	35.3	44	34.3	46
Indic BERT	SVM RBF	68.5	43.9	39.5	40.6	66
	SVM poly	67.9	45	40.3	41.2	66
	SVM linear	59.5	36.6	40.2	37.8	61
	Random Forest	65.8	40	26.5	25.8	58
	Logistic Regression	51.3	38.1	43.6	37.4	55
	KNN	65.3	42.6	34.3	34.9	63
	Decision Tree	49	23.6	23.6	23.6	49
	Adaboost	60	29.8	26.1	23.5	54
	Naive Bayes	45.6	27.8	32.9	28	47
FNET	SVM RBF	44.9	36	41.3	34.6	47
	SVM poly	46.1	35.9	41.8	34.8	48
	SVM linear	47.2	35.6	41.5	35	49
	Random Forest	64.8	38	27.8	28.8	59
	Logistic Regression	37.9	30.1	37.2	29	38
	KNN	62.6	44.6	30.7	32.8	59
	Decision Tree	51	29.4	30.5	29.8	51
	Adaboost	59.8	31.2	30.8	30.3	57
	Naive Bayes	26.2	24	26.7	10.7	23

TABLE 4. Results for Malayalam DravidianLangTech dataset.

Embedding	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	
					Macro	Weighted
BERT	SVM RBF	96.3	92	67.1	76.2	96
	SVM poly	96.2	89.8	67.9	76.2	96
	SVM linear	93.2	63.4	59.1	60.3	93
	Random Forest	95.3	98.5	63.1	75	95
	Logistic Regression	83.1	40.5	59.3	45.9	86
	KNN	95.6	90.9	65.1	74.4	95
	Decision Tree	92.2	63.1	66.5	64.6	92
	Adaboost	89.5	39	34.8	35	88
	Naive Bayes	31.1	28	46.6	21.2	40
DistilBERT	SVM RBF	96.2	89.9	69.7	77.2	96
	SVM poly	96	86.1	69.9	76.3	96
	SVM linear	94	72.8	55.8	61.9	96
	Random Forest	96	98.7	64.7	76.2	96
	Logistic Regression	81.6	40.1	60.9	45.6	85
	KNN	95.9	90.9	68.3	76.8	96
	Decision Tree	92.8	66.7	65.2	65.9	93
	Adaboost	86.3	40.5	37.3	36	87
	Naive Bayes	15.8	27.8	43.2	15.1	17
LaBSE	SVM RBF	96	92.5	67.7	76.9	96
	SVM poly	95.9	92.9	67.1	76.7	96
	SVM linear	90.6	55.1	27.8	30.5	88
	Random Forest	94.9	98.5	61.9	74.2	94
	Logistic Regression	68.9	33.6	48	35.1	75
	KNN	94	80.7	67.5	72.9	94
	Decision Tree	91.7	67	62.5	64.2	92
	Adaboost	84.8	27.5	25.4	26.2	83
	Naive Bayes	52.6	27.7	49.7	27.1	62
MuRIL	SVM RBF	94.8	84.8	66.5	73.5	95
	SVM poly	94	71	50.8	56.9	93
	SVM linear	91.5	37	28.9	31.2	89
	Random Forest	95.1	98.4	63.7	75.7	95
	Logistic Regression	80.2	43.4	64.5	48.2	83
	KNN	94.7	84.3	67.6	73.8	94
	Decision Tree	92.1	65.2	67.4	66.2	92
	Adaboost	73.7	30.9	32.7	30.6	84
	Naive Bayes	48.6	32	48.6	30.9	59
XLM	SVM RBF	94.8	84.8	66.5	73.5	95
	SVM poly	94.7	79.6	67.7	72.7	94
	SVM linear	92	62.5	57.9	59.7	92
	Random Forest	95.1	98.4	63.7	75.7	95
	Logistic Regression	78.5	42.4	64.1	46.9	82
	KNN	94.7	84.3	67.6	73.8	94
	Decision Tree	91.8	64.8	67.4	66	92
	Adaboost	83.7	30.9	32.7	30.6	84
	Naive Bayes	48.6	32	48.6	30.9	59
Indic BERT	SVM RBF	96.3	89.9	70	77.2	93
	SVM poly	96.3	88.5	70.3	77	96
	SVM linear	87.6	50.2	70.8	56.7	89
	Random Forest	95.8	98.2	64.6	75.9	95
	Logistic Regression	61.5	34.1	65.4	35.6	70
	KNN	95.7	87.5	66.5	74.4	95
	Decision Tree	92.6	65.5	66.1	65.5	93
	Adaboost	84.5	28.1	34.2	30.2	85
	Naive Bayes	25.6	28.1	44.2	19.3	33
FNet	SVM RBF	64.3	34.2	65.2	36.9	59
	SVM poly	65.6	34.9	66.4	38.1	61
	SVM linear	63.5	34	65.1	36.4	63
	Random Forest	95.6	98.4	63.8	75.5	95
	Logistic Regression	36.4	28.5	52.9	23.6	42
	KNN	94.5	87.1	64.4	72.5	94
	Decision Tree	92.7	66.3	65.8	66	93
	Adaboost	86.1	32	34	32.3	85
	Naive Bayes	5.3	24.1	28.9	8.3	25

TABLE 5. Results for Malayalam HASOC 2021 dataset.

Embedding	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	
					Macro	Weighted
BERT	SVM RBF	60.9	75.4	62.4	68.3	62
	SVM poly	60.1	74.6	62.1	67.7	61
	SVM linear	63.3	76.9	65.2	70.6	64
	Random Forest	59.1	77.1	56	64.9	60
	Logistic Regression	62.6	75.9	65.3	70.2	64
	KNN	51.1	71.9	45.2	55.5	52
	Decision Tree	52.8	71	50.8	59.2	54
	Adaboost	59.7	75.4	59.9	66.7	61
	Naive Bayes	57.3	79	50.1	61.3	58
DistilBERT	SVM RBF	59.9	73.9	62.8	67.9	61
	SVM poly	60.9	74.7	63.6	68.7	62
	SVM linear	60.2	74.8	61.9	67.7	61
	Random Forest	60.4	75..8	60.7	67.4	62
	Logistic Regression	61.0	75.1	63.1	68.6	62
	KNN	52.5	71.6	49.2	58.3	54
	Decision Tree	54.7	71.9	54.1	61.7	56
	Adaboost	60.5	75.0	62.2	68.0	62
	Naive Bayes	58.6	77.9	53.9	63.7	60
LaBSE	SVM RBF	62.10	77.9	61.2	68.5	63
	SVM poly	62.10	77.4	61.9	68.8	63
	SVM linear	60.1	76.8	58.5	66.4	61
	Random Forest	61.8	77.1	61.8	68.6	63
	Logistic Regression	62.2	77.4	62.1	68.9	63
	KNN	55.9	75.8	51	60.9	57
	Decision Tree	51.2	68.8	50.7	58.4	53
	Adaboost	60.6	78.9	56.9	66.1	62
	Naive Bayes	59.5	75.7	59	66.3	61
MuRIL	SVM RBF	66.2	80.8	65.5	72.3	67
	SVM poly	65.1	80.1	64.3	71.3	66
	SVM linear	67.1	80.9	67.1	73.4	68
	Random Forest	59.8	75.8	59.4	66.6	61
	Logistic Regression	68	74.6	79.7	77.1	67
	KNN	47.6	73.2	35.6	47.9	48
	Decision Tree	52.7	69.7	52.9	60.2	54
	Adaboost	60.7	77.2	59.3	67.1	62
	Naive Bayes	58.5	77.2	54.7	64	60
XLM	SVM RBF	57.9	75.6	55.6	64	59
	SVM poly	56.2	73.6	54.8	62.8	58
	SVM linear	59.1	75.1	59	66.1	60
	Random Forest	58.2	74.2	58.4	65.3	59
	Logistic Regression	60	75.6	60.1	67	61
	KNN	52.5	71.7	48.9	58.1	54
	Decision Tree	48.4	67.2	46.4	54.9	50
	Adaboost	53.8	70.8	53.8	61.1	55
	Naive Bayes	55.5	77.8	47.7	59.1	57
Indic BERT	SVM RBF	60	76.3	59.1	66.6	61
	SVM poly	59.7	76.4	58.4	66.2	61
	SVM linear	60	78.5	56.1	65.5	61
	Random Forest	56.6	75.7	52.6	62.1	58
	Logistic Regression	62.2	80.2	58.4	67.6	63
	KNN	52.8	73.3	47.3	57.5	54
	Decision Tree	47.5	65.4	47.4	55	49
	Adaboost	56.5	74.6	53.9	62.6	58
	Naive Bayes	53.4	75.9	45.3	56.8	54
FNet	SVM RBF	63.1	76.7	65	70.4	64
	SVM poly	63.7	77.7	64.9	70.7	65
	SVM linear	63.9	77.4	65.8	71.1	65
	Random Forest	53.4	73.7	48.1	58.2	55
	Logistic Regression	58.2	74.7	57.6	65.1	59
	KNN	51.1	67.7	52.6	59.2	53
	Decision Tree	52.1	69.8	51.1	59	54
	Adaboost	53.9	74.2	78.6	58.7	55
	Naive Bayes	52.3	70.1	51.1	59.1	54

TABLE 6. Results for Malayalam HASOC 2020 dataset.

Embedding	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	
					Macro	Weighted
BERT	SVM RBF	91	79	68.1	73.1	91
	SVM poly	90.2	76.2	66.7	71.1	90
	SVM linear	84.8	56.5	66.7	61.1	85
	Random Forest	90.5	100	47.2	64.2	89
	Logistic Regression	81	48.2	73.6	58.2	82
	KNN	85.8	60.3	61.1	60.7	86
	Decision Tree	82.8	52.1	51.4	51.7	83
	Adaboost	85.8	68.3	38.9	49.6	84
	Naive Bayes	67.2	31.2	68.1	42.8	71
DistilBERT	SVM RBF	88.5	69.7	63.9	66.7	88
	SVM poly	89	70.6	66.7	68.6	89
	SVM linear	86.5	59.4	79.2	67.9	87
	Random Forest	89.7	97	44.4	61	88
	Logistic Regression	76.2	40.2	65.3	49.7	78
	KNN	87.3	64.8	63.9	64.3	87
	Decision Tree	86.8	63.8	61.1	62.4	87
	Adaboost	86.5	72.5	40.3	51.8	85
	Naive Bayes	64.2	28.5	65.3	39.7	68
LaBSE	SVM RBF	92.7	89.1	68.1	77.2	92
	SVM poly	91.7	91.5	59.7	72.3	91
	SVM linear	72.9	45.5	76.4	57	81
	Random Forest	90	100	45.8	62.9	89
	Logistic Regression	78	43.5	75	55.1	80
	KNN	89.7	79.2	58.3	67.2	89
	Decision Tree	87	64.7	61.1	62.9	87
	Adaboost	85.5	62.5	48.6	54.7	85
	Naive Bayes	75	38.7	66.7	49	77
MuRIL	SVM RBF	84.8	55.4	77.8	64.7	86
	SVM poly	85.3	57	73.6	64.2	86
	SVM linear	70.5	34.9	73.6	47.3	74
	Random Forest	90	97.1	45.8	62.3	88
	Logistic Regression	69.5	32.9	66.7	44	73
	KNN	88	66.2	68.1	67.1	88
	Decision Tree	84.5	57.4	54.2	55.7	84
	Adaboost	85	60.7	47.2	53.1	84
	Naive Bayes	63.5	27.7	63.9	38.7	68
XLM	SVM RBF	89.7	73.1	68.1	70.5	90
	SVM poly	88.5	68.1	68.1	68.1	89
	SVM linear	83.5	53.4	65.3	58.7	84
	Random Forest	91.5	93.2	56.9	70.7	91
	Logistic Regression	80	46.5	73.6	57	82
	KNN	89.5	76.8	59.7	67.2	89
	Decision Tree	85.5	59.7	59.7	59.7	85
	Adaboost	84.5	59.6	43.1	50	83
	Naive Bayes	73.8	35.4	55.6	43.2	76
Indic BERT	SVM RBF	89.2	76.4	58.3	66.1	89
	SVM poly	90.5	82.7	59.7	69.4	90
	SVM linear	82.8	51.5	70.8	59.6	84
	Random Forest	89.2	1	40.3	57.4	87
	Logistic Regression	78	42.6	63.9	51.1	80
	KNN	88.2	76.6	50	60.5	87
	Decision Tree	85.5	89.5	61.1	60.3	86
	Adaboost	82.8	53.2	34.7	42	81
	Naive Bayes	61.5	24.4	54.2	33.6	66
FNet	SVM RBF	82.5	100	2.8	5.4	75
	SVM poly	82.5	100	2.8	5.4	75
	SVM linear	82.5	100	2.8	5.4	75
	Random Forest	89.7	100	43.1	60.2	88
	Logistic Regression	82.8	100	4.2	8	76
	KNN	86.5	69.6	44.4	54.2	85
	Decision Tree	84.5	57.4	54.2	55.7	84
	Adaboost	85.8	66.7	41.7	51.3	84
	Naive Bayes	73	22.7	20.8	21.7	73

TABLE 7. Results for Tamil DravidianLangTech dataset.

Embedding	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	
					Macro	Weighted
BERT	SVM RBF	74.2	43.1	36.6	39.1	72
	SVM poly	73.4	41.8	36.4	38.5	71
	SVM linear	52	32.4	41.3	33.7	59
	Random Forest	73.4	61.4	18.7	17.9	63
	Logistic Regression	49.5	31.6	42.3	32.6	57
	KNN	70.4	39.8	28.4	31.4	67
	Decision Tree	58.7	23.4	24.1	23.7	59
	Adaboost	71.4	23.3	23	20.6	35
	Naive Bayes	34.3	24.6	34.8	22.4	40
DistilBERT	SVM RBF	74.3	45.3	38	40.7	72
	SVM poly	73.8	44.4	37.8	40.4	72
	SVM linear	52.3	33.5	41.9	34.7	59
	Random Forest	74.1	67.8	21.7	22.2	64
	Logistic Regression	49.4	32.2	44.7	33.6	56
	KNN	71.8	42.4	30.3	33.4	68
	Decision Tree	60.5	26.2	26.4	26.3	61
	Adaboost	72.7	34.3	27.2	25.8	65
	Naive Bayes	29.1	24.1	34.7	19.7	33
LaBSE	SVM RBF	75.8	50.1	36	39.8	72
	SVM ploy	75.2	51.3	34.7	38.7	72
	SVM linear	51	31	41.7	32.1	58
	Random Forest	73.3	68	18.3	17.2	63
	Logistic Regression	50.8	30.9	42.7	32	58
	KNN	73.6	43.2	30.8	34.1	69
	Decision Tree	60	22.8	22.9	22.9	60
	Adaboost	71.6	29.2	23.2	22.9	64
	Naive Bayes	51.7	28.9	38.5	29.9	58
MuRIL	SVM RBF	57.2	36	46	38.1	63
	SVM poly	57.9	36	45.3	38.1	63
	SVM linear	51.8	36.1	44.5	36.7	58
	Random Forest	74.7	63.8	24.9	25.6	65
	Logistic Regression	49.1	29.4	39.5	29.9	55
	KNN	73.1	45.3	34.2	37.2	70
	Decision Tree	60.1	27.2	27.2	27.2	61
	Adaboost	72.6	41.4	28.9	26	65
	Naive Bayes	46.5	30.4	37.5	28.8	52
XLM	SVM RBF	70.5	37.1	34.3	35.5	69
	SVM poly	68.2	34.4	34.1	34.1	67
	SVM linear	52.3	31.4	41.4	32.9	59
	Random Forest	73.9	58.5	20.4	20.9	64
	Logistic Regression	49.2	31	41.6	31.8	56
	KNN	70.2	34.6	29.3	31.2	67
	Decision Tree	58.8	23.8	23.9	23.8	59
	Adaboost	71.8	34.9	22.9	22.8	64
	Naive Bayes	43.1	28	35.1	27.5	50
Indic BERT	SVM RBF	71.9	40.6	26.2	38	70
	SVM poly	71.9	41.4	36.7	38.5	70
	SVM linear	50.5	32.5	41.5	33.6	57
	Random Forest	74.4	67.1	23.2	23.7	64
	Logistic Regression	47.5	31.3	43.4	32.2	55
	KNN	72.2	39.4	31.4	33.7	68
	Decision Tree	59.6	25.6	26.1	25.9	60
	Adaboost	73.2	37.9	24	22.8	64
	Naive Bayes	36.5	24.8	33.9	21.6	42
FNet	SVM RBF	44.1	27.9	38.8	28.1	51
	SVM poly	44.6	28.3	39.4	28.6	51
	SVM linear	45.6	28.8	40.3	29.3	52
	Random Forest	73.3	62.5	18.1	16.8	62
	Logistic Regression	28.6	22.9	31.1	19	35
	KNN	70.1	30	21.2	22.4	64
	Decision Tree	57.5	21.1	21.5	21.3	58
	Adaboost	72.6	36.6	21.7	20.4	63
	Naive Bayes	12.6	21.8	24.9	11.8	12

TABLE 8. Results for Tamil HASOC 2021 dataset.

Embedding	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	
					Macro	Weighted
BERT	SVM RBF	73.9	72.4	80.5	76.2	74
	SVM poly	74.5	73.9	78.8	76.3	74
	SVM linear	74.1	71.8	82.9	76.9	74
	Random Forest	69.4	97.3	80	73.1	69
	Logistic Regression	76.4	77.1	76	76	76
	KNN	64.9	65.1	69.8	67.4	68
	Decision Tree	55.7	56.6	63.1	59.7	55
	Adaboost	67.2	66.4	74.8	70.3	67
	Naive Bayes	63.6	64.8	65.7	65.2	64
DistilBERT	SVM RBF	74	73.6	78.3	75.9	74
	SVM poly	74.6	74.6	77.6	76.1	75
	SVM linear	75.7	73.1	84.3	78.3	76
	Random Forest	68.6	65	85.7	73.9	67
	Logistic Regression	76.1	76.9	75.7	75.7	76
	KNN	63.1	62.2	73.8	67.5	63
	Decision Tree	57.1	57.7	65.5	61.3	57
	Adaboost	66.2	64.4	78.3	70.7	66
	Naive Bayes	65.2	62.6	82.1	71.1	64
LaBSE	SVM RBF	65.8	68.7	63.1	65.8	66
	SVM poly	64.6	68.1	60	63.8	65
	SVM linear	72.9	76.9	68.3	72.4	73
	Random Forest	65.7	69.9	59.8	64.4	66
	Logistic Regression	72.8	73.5	73.1	72.7	73
	KNN	61.1	64.6	56	59.9	61
	Decision Tree	54.7	56.9	53.1	54.9	55
	Adaboost	65.5	68.2	62.9	65.4	65
	Naive Bayes	62.7	63.5	66.7	65	63
MuRIL	SVM RBF	76.5	75.8	80.5	78.1	76
	SVM poly	76.1	75.1	81	77.9	76
	SVM linear	71.5	70.2	78.6	74.2	71
	Random Forest	73.4	71.1	82.1	76.2	73
	Logistic Regression	68.9	68.9	69	68.9	69
	KNN	65.8	66	70.7	68.3	66
	Decision Tree	60	61.9	60.2	61	60
	Adaboost	71.9	74.4	70	72.1	72
	Naive Bayes	68.4	68	74.3	71	68
XLM	SVM RBF	65.7	67	67.1	67.1	66
	SVM poly	63.9	65.2	65.5	65.3	64
	SVM linear	67.1	67.7	70	68.9	67
	Random Forest	65.1	66.6	66	66.3	65
	Logistic Regression	68.3	68.3	68.2	68.2	68
	KNN	56.4	56.9	67.1	61.6	56
	Decision Tree	56.2	57.9	57.9	57.9	56
	Adaboost	64.5	65.9	65.7	65.8	64
	Naive Bayes	60.3	60.6	67.4	63.8	60
Indic BERT	SVM RBF	70.5	70.2	75.2	72.6	70
	SVM poly	71.5	71	76.4	73.6	71
	SVM linear	72.6	73	75.2	74.1	73
	Random Forest	68.8	69.4	71.4	70.4	69
	Logistic Regression	74.4	73.9	78.3	76.1	74
	KNN	64.1	67.5	59..8	63.4	64
	Decision Tree	61.4	63.7	59.8	61.7	61
	Adaboost	66.5	68.1	66.7	67.4	66
	Naive Bayes	67.9	70.5	66	68.1	68
FNet	SVM RBF	67.7	67.5	72.9	70.1	68
	SVM poly	69.3	68.7	75.2	71.8	69
	SVM linear	70	69.1	76.7	72.7	70
	Random Forest	62.3	61.9	71.2	66.2	62
	Logistic Regression	64	65.1	66.2	65.6	64
	KNN	59.9	61.4	61.4	61.4	60
	Decision Tree	53.7	55.9	52.1	53.9	54
	Adaboost	63.2	64.6	64.8	64.7	63
	Naive Bayes	53.2	59.2	32.1	41.7	51

TABLE 9. Comparison of the weighted F1-scores of the proposed work and the state of the art work. In the table the BERT+Machine Learning models that excel the BERT-based classifiers are highlighted.

Dataset	Our approach		State of the art approach	
	Embedding	Weighted F1-score (%)	Embedding	Weighted F1-score (%)
DravidianLangTech 2021 Malayalam	MuRIL	**95**	MuRIL	93.01 [66]
	BERT	**96**	BERT	92.88 [66]
	DistilBERT	**96**	DistilBERT	94.65 [66]
	XLM-RoBERTa	**95**	XLM-RoBERTa	92.88 [66]
			ULMFiT	96.03 [66]
DravidianLangTech 2021 Tamil	MuRIL	**70**	MuRIL	61.12 [66]
	BERT	72	BERT	75.56 [66]
	DistilBERT	72	DistilBERT	74.89 [66]
	XLM-RoBERTa	**69**	XLM-RoBERTa	61.12 [66]
			ULMFiT	78.95 [66]
DravidianLangTech 2021 Kannada	MuRIL	**66**	MuRIL	38.9 [66]
	BERT	66	BERT	69.36 [66]
	DistilBERT	66	DistilBERT	70.10 [66]
	XLM-RoBERTa	64	XLM-RoBERTa	68.51 [66]
			ULMFiT	70 [66]
HASOC 2021 Tamil	MuRIL	**76**	word-level+character-level N-gram based TF-IDFTD-IDF+character with Machine Learning classifiers	67.8 [17]
HASOC 2021 Malayalam	MuRIL	**68**	ensemble model which used character N-gram based TF-IDF features with Machine Learning classifiers	76.6 [17]
HASOC 2020 Malayalam	LaBSE	**92**	TD-IDF+character n-gram features with Machine Learning classifiers	78 [53]

TABLE 10. Performance of each embedding trained on three sets of binary data over the In-house test set.

Embedding	Pre-trained data	Accuracy (%)	Precision (%)	Recall (%)	F1Score (%)	
					Macro	Weighted
MuRIL	HASOC 2021 Malayalam	44.6	57.9	63.1	60.4	43
	HASOC 2020 Malayalam	54.8	76.6	46.9	58.1	56
	HASOC Tamil	55.1	76.1	48.1	58.9	56

A. DravidianLangTech DATA

1) KANNADA

All the models applied to this CodeMix data achieved exceptional results. MuRIL embedding with SVM (RBF) classifier obtained the highest results. The error analysis of this data is based on the MuRIL embedding results with the Random Forest classifier. Out of the 2001 test set, only 89 were misclassified.

- The first level error analysis was done on the lengths of the sentences. We checked for the length of the misclassified sentences and the rightly classified sentences. Figures 3 and 4 show the histogram plots of the sentence lengths for misclassified and rightly classified sentences. It is evident from the two figures that the sentence lengths have not affected the predictions as we observe that most of the misclassified sentences and

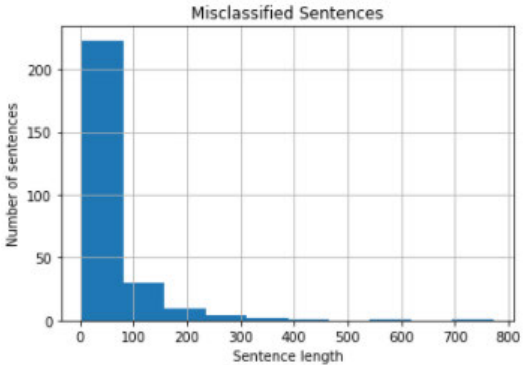

FIGURE 3. The histogram plot of the sentence lengths for the misclassified sentences for the top performing model on DravidianLangTech Kannada data.

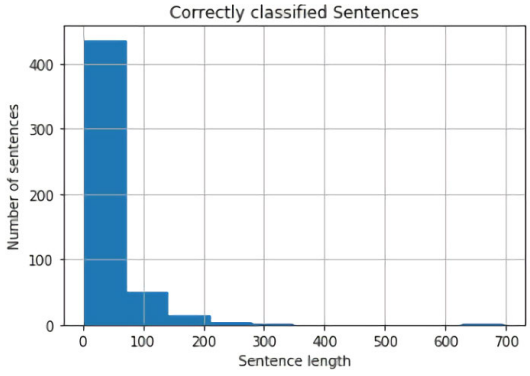

FIGURE 4. The histogram plot of the sentence lengths for the correctly classified sentences for the top performing model on DravidianLangTech kannada data.

the rightly classified sentences have sentence lengths between 25 and 50.

- The data is a mixture of multiple Indian and non-Indian languages written in Roman as well as Language specific text. We manually analysed the effect of language on misclassification by comparing the languages in misclassified and correctly classified sentences. It was observed that the difference in language does not affect the prediction.
- Among the misclassified sentences, we observed that a few of the sentences were mislabelled. Table 11 shows the test sentences that are mislabelled but predicted rightly by the model. Most of the sentences are in pure Roman script but are labeled as Not_offensive but have some offensive content. This is one of the reasons for the model's low performance.
- Certain comments do not have any offensive words but are written in Latin script and do not contain any Kannada words, which have to fall into the non_Kannada class, but the model predicted it as Not_offensive.
- Text: *D boss fans inda full support iden #jaidboss*
  Translation: *D boss fan's full support is there #jaidboss*

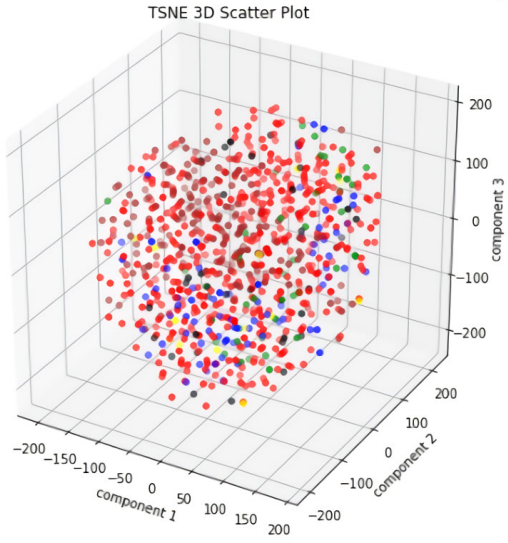

FIGURE 5. 3D TSNE plot for Kannada DravidianLangTech dataset.The figure has the scatter plot for 6 different classes.

The above sentence doesn't contain any Kannada words hence as per the definition the sentence is labelled as non_Kannada. But it doesn't have any offensive words and it gives a positive sentiment which resulted in the sentence getting misclassified as Not_offensive.
Table 12 gives the examples which has high chance of falling into multiple classes due to the lack of offensive words.

- Text: *Found 806 rashmika mangannas.…*
  Translation: *Found 806 Rashmika monkeys*
  This is a very confusing sentence as the word manganna sounds like "Mandanna". Though the word manganna means monkeys in Kannada the replaced character can be considered a spelling error. Hence the model misclassified it as "Not-offensive".
- Figures 5 and 6 show the dataset's 3D TSNE and PCA scatter plots. The 768-dimensional sentence embedding is mapped to 3D and plotted. It is clear from the Figures that the dataset is clumsy, there are many overlapping data points, and there is no clear separation between the dataset of each class. This has made the classification strenuous.

2) MALAYALAM

All the models applied to this CodeMix data achieved exceptional results. DistilBERT embedding with Random Forest classifier and MuRIL obtained the highest results. The error analysis on this data is done on the results obtained for DistilBERT embedding with Random Forest classifier. Out of the 2001 test set, only 89 were misclassified. This section discusses the misclassification errors of the model for the classification task on various datasets of different languages. The performance is compared using a weighted F1-score.

TABLE 11. These are test sentences from Kannada DravidianLangTech dataset that are mislabelled but predicted rightly by the model.

Example	Label	Prediction
Saryagi rubbidira brother	Not_offensive	Offensive_Targeted_Insult_Individual
Chinndi guru ennu roast madu	Not_offensive	Offensive_Targeted_Insult_Individual
@Like a Phoenix hm avarige ade helbku kushi padli swalp	Not_offensive	Offensive_Targeted_Insult_Group
@Kiccha Vitthal papa bidu guru amele athbitre kashta attar flop ago movie ge support maDecision Treeavre	Offensive_Targeted_Insult_Group	Offensive_Untargeted
Hindi kuthe aahe	Not_offensive	not-Kannada
Super agi ugdiya bro kittogiro bevarsigalige hege video maDecision Treea iri support maDecision Treea irtivi super	Not_offensive	Offensive_Targeted_Insult_Group
RIP kannada film chamber.No proper plannings.	Offensive_Targeted_Insult_Group	Not_offensive
Guru ee desha uddhara agalla bedu bhai indian youth waste bedu	Not_offensive	Offensive_Targeted_Insult_Other
Bro Gadi le hoge nintkoloke heali avn beja bayge barrage bole magange	Not_offensive	Offensive_Targeted_Insult_Individual
Dharsha ge. Rost madu guru pls	Not_offensive	Offensive_Targeted_Insult_Individual

TABLE 12. These are test sentences from Kannada DravidianLangTech dataset that are misclassified as Not_offensive as they do not contain any offensive words but are labelled as non_Kannada as they do not contain any Kannada words.

Text	Label	Prediction
Mee too sir please do consider	not-Kannada	Not_offensive
Good song good making	not-Kannada	Not_offensive
We like shanvi nWe hate sanvi	not-Kannada	Not_offensive
GOOD OBSERVATION	not-Kannada	Not_offensive
Togari tippa all time favorite evergreen	not-Kannada	Not_offensive
Dost excellent video good no one video	not-Kannada	Not_offensive
Wonder full songs	not-Kannada	Not_offensive
Super super super super super super super super super super super super super super super super super super	not-Kannada	Not_offensive
Kannada songs always great....	not-Kannada	Not_offensive
It's like Good bad weird	not-Kannada	Not_offensive

- The first level error analysis was done on the length of sentences. We checked the length of the misclassified sentences and the rightly classified sentences. Figures 7 and 8 show the histogram plots of the sentence lengths for misclassified and rightly classified sentences. It is evident from the two figures that the sentences' length has not affected the predictions as we observe that most of the misclassified sentences and the rightly classified sentences have sentences length between 25 and 50.
- The data is a mixture of multiple Indian and non-Indian languages written in Roman and language-specific text. We manually analsed the effect of language on misclassification by comparing the languages in misclassified and correctly classified sentences. It was observed that the difference in language does not affect the prediction.
- Among the misclassified sentences, we observed that a few of the sentences were mislabelled. Table 13 shows the test sentences that are mislabelled but predicted rightly by the model. Most of the sentences are in pure Roman script but are labeled as non-Malayalam. However, as per the definition, sentences that do not have Malayalam words written in Malayalam script or Latin script are labeled as non-Malayalam [14]. This is one of the reasons for the model's low performance.

TABLE 13. These are test sentences from Malayalam DravidianLangTech dataset that are mislabelled but predicted rightly by the model.

Example	Label	Prediction
Next 200 Crore club Malayalam muvi	Not_offensive	not-Malayalam
192K Views	Not_offensive	not-Malayalam
Happy birthday EKKA MEGA FACE OF INDIAN CINEMA ONE AND ONLY MAMMOOTTY	not-Malayalam	Not_offensive
Padma innaanu kande	Offensive_Untargeted	Not_offensive
Subscribe my channel for latest viral status videos (Song	not-Malayalam	Not_offensive
Wow wow wow	not-Malayalam	Not_offensive
Oops it's 3.21 but you don't feel like it lengthy awesome	not-Malayalam	Not_offensive
2m views 130k likes 12k comments	not-Malayalam	Not_offensive
Thalaivar dialogue All d best mommukka	not-Malayalam	Not_offensive
Lalettan fans like adike support me	not-Malayalam	Not_offensive
All the best ikka from #lalettan fans	not-Malayalam	Not_offensive

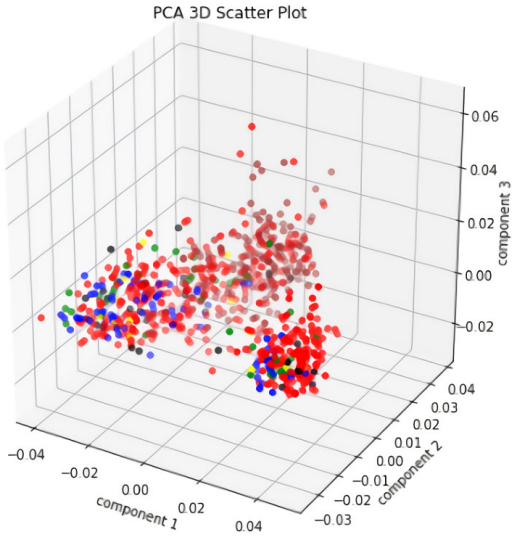

FIGURE 6. 3D PCA plot for Kannada DravidianLangTech dataset.The figure has the scatter plot for 6 different classes.

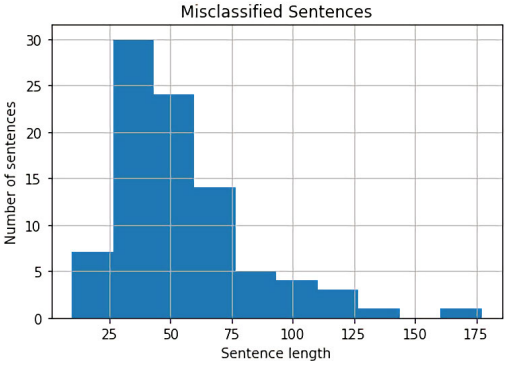

FIGURE 7. The histogram plot of the sentence lengths for the misclassified sentences for the top performing model on DravidianLangTech Malayalam data.

- There are specific comments that do not have any offensive words but are written in Latin script and do not contain any Malayalam words, which have to fall into the non_Malayalam class, but the model predicted it as Not_offensive.

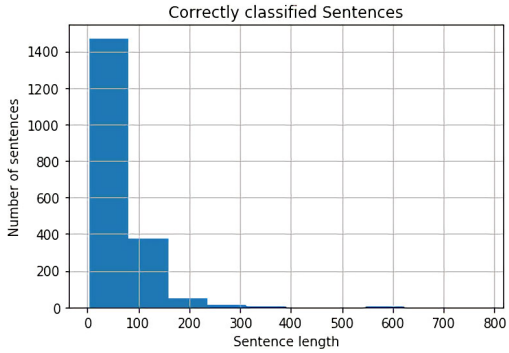

FIGURE 8. The histogram plot of the sentence lengths for the correctly classified sentences for the top performing model.

- Text: *OUR BROTHER IS COMING 'BIG BROTHER'*
  The above sentence does not contain any Malayalam words. So as per the definition, the sentence is labeled as non_Malayalam. But it does not have any offensive words, and it gives a positive sentiment which resulted in the sentence getting misclassified as Not_offensive
  Table 14 gives the examples which has high chance of falling into multiple classes due to the lack of offensive words.
- Text: *ithokke comedy pole und …aa bahubaliyude manam kalayo ?*
  Translation: *This seems like comedy. Will Bagubali's fame be ruined.*
  This comment is labelled as "Offensive_Targeted_Insult_Group" as it is in a way trying to insult a group. But as it doesn't contain any offensive words the comment got misclassified as Not_offensive.
- Text: *Aye kuura trailer oola padam chali mohanlal**************
  Translation: *Yuk bad trailer worst movie dirty mohanlal**************
  In this comment, the author is trying to insult a group of people who worked behind the movie by writing bad comments about the trailer, movie and the actor Mohanlal, so the comment is Offensive_Targeted_Insult_Group, but due to the lack of any offensive words, it is misclassified as Not_offensive.

TABLE 14. These are test sentences from Malayalam DravidianLangTech dataset that are misclassified as Not_offensive as they do not contain any offensive words but are labelled as non_Malayalam as they do not contain any Malayalam words.

Example	Label	Prediction
Trending 1 just 5 hours	not-Malayalam	Not_offensive
similar to Kirik party (kannada movie)	not-Malayalam	Not_offensive
After Pulimurugan Biggest Blockbuster ku Waitiing For Tn Fans	not-Malayalam	Not_offensive
68 yr old young man.	not-Malayalam	Not_offensive
Love frm kannada we need kannada dub	not-Malayalam	Not_offensive
Youtube still hang 210 likes 283 views	not-Malayalam	Not_offensive
Bigil..Kerala Thalapaty fans hit like.	not-Malayalam	Not_offensive
After Pulimurugan Biggest Blockbuster ku Waitiing For Tn Fans	not-Malayalam	Not_offensive
Mammooty + linto kurian + ajai vasudev = swag ka baap	not-Malayalam	Not_offensive
Mamangam on Nov 21st.. in 4 language....	not-Malayalam	Not_offensive

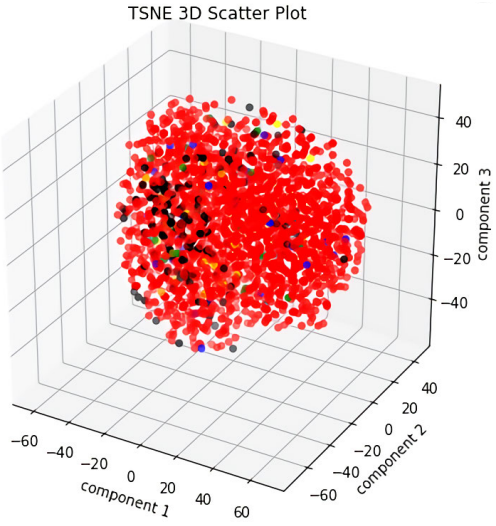

FIGURE 9. 3D TSNE plot for Malayalam DravidianLangTech dataset.The figure has the scatter plot for 5 different classes.

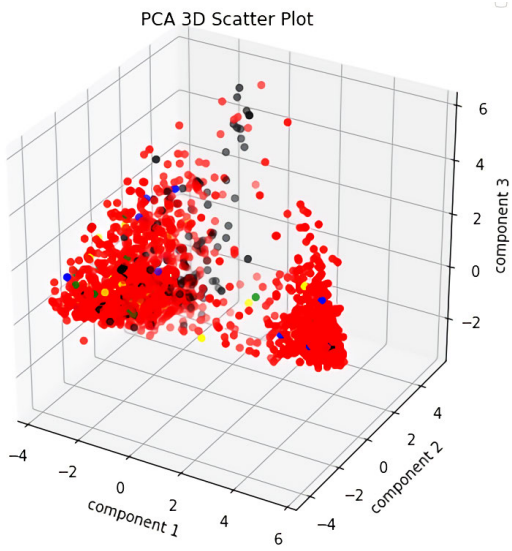

FIGURE 10. 3D PCA plot for Malayalam DravidianLangTech dataset.The figure has the scatter plot for 5 different classes.

- The dataset has sentences that do not use any direct offensive words but are sarcastic or insulting which belong to the offensive classes.
  Text: *Ella oollapadathinteyum stiram cheruva. 8nilayil padam pottum*
  Translation: *cliche ingredient of all flop movies. This movie is going to be a failure*
  Though this comment does not contain any offensive words, the sentence as a whole is meant to insult the director or any person behind that movie. The author gives negative comments about the movie. Hence the sentence is actually "Offensive Targeted Individual" but is misclassified as "Not Offensive."
- It is also observed that the significant misclassification happens to the Not_offensive class as the data is highly imbalanced. The Not_offensive class has a total of 17697 data points, on the other hand, the rest of the data points from other classes sum up to 2313.
- Figures 9 and 10 show the 3D TSNE and PCA scatter plots of the dataset. The 768-dimensional sentence embedding is mapped to 3D and plotted. It is clear from the Figures that the dataset is clumsy, there are many overlapping data points, and there is no clear separation between the dataset of each class. This has made the classification strenuous.

3) TAMIL

The highest results on this data were obtained using DistilBERT embedding with SVM (RBF) classifier and MuRIL embedding. The error analysis of this data is done on the results of this high-performing model.

- The first level error analysis was done on the length of the sentences. We checked for the length of the misclassified sentences and the rightly classified sentences. Figures 11 and 12 show the histogram plots of the sentence lengths for misclassified and rightly classified sentences. It is evident from the two figures that the sentence lengths have not affected the predictions as we observe that most of the misclassified sentences and the rightly classified sentences have sentence lengths between 10 and 100.
- The data is a mixture of multiple Indian and non-Indian languages written in Roman as well as Language

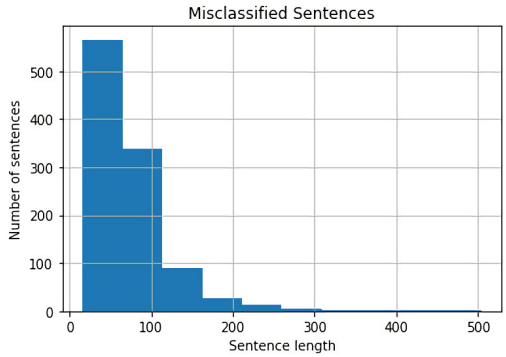

FIGURE 11. The histogram plot of the sentence lengths for the misclassified sentences for the top performing model on DravidianLangTech Tamil data.

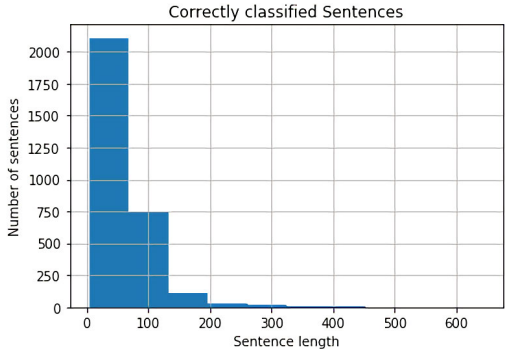

FIGURE 12. The histogram plot of the sentence lengths for the correctly classified sentences for the top performing model on DravidianLangTech Tamil data.

specific text. We manually analysed the effect of language on misclassification by comparing the languages in misclassified and correctly classified sentences. It was observed that the difference in language does not affect the prediction.

- Among the misclassified sentences, we observed that a few of the sentences were mislabelled. Table 15 shows a few test sentences that are mislabelled but predicted rightly by the model. This is one of the reasons for the model's low performance.
- Some of the wrong predictions are that the non-Tamil sentences do not have any offensive content. These sentences are misclassified as not offensive. Table 16 shows a few sentences that are not-Tamil but do not contain any offensive words and are hence misclassified as Not_offensive.
- Text *Rajin political entry dailog 1996/2016 10 years one dailog naaku baag nachindi ——— Offensive_Targeted_Insult_Individual*
  Translation: *Political entry dialog of Rajnikanth between 1996-2016. I liked it.*
  The above sentence is written in Telugu, so it is labeled as ''not-Tamil.'' This sentence does not have any offensive words; instead, it has positive words such as 'I liked it,' but the sentence has a sarcastic meaning which was captured by the model and hence misclassified as Offensive_Targeted_Insult_Individual.
- Figures 13 and 14 show the 3D TSNE and PCA scatter plots of the dataset. The 768-dimensional sentence embedding is mapped to 3D and plotted. It is clear from the Figures that the dataset is clumsy there are many overlapping data points, and there is no clear separation between the dataset of each class. This has made the classification strenuous.

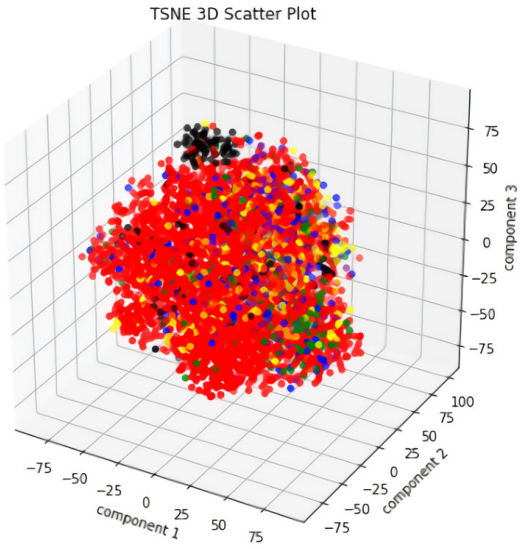

FIGURE 13. 3D TSNE plot for Tamil DravidianLangTech dataset.The figure has the scatter plot for 6 different classes.

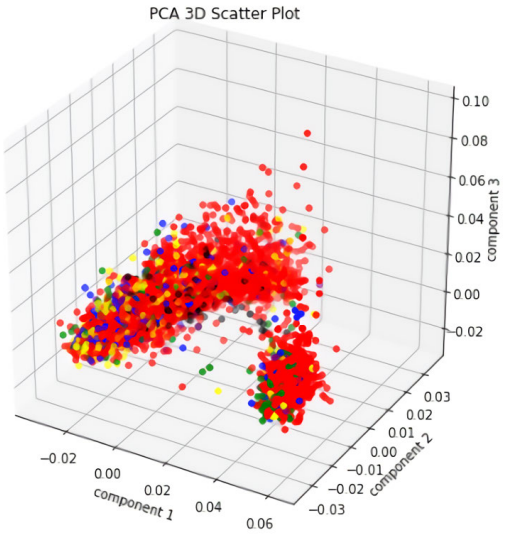

FIGURE 14. 3D PCA plot for Tamil DravidianLangTech dataset.The figure has the scatter plot for 6 different classes.

B. HASOC DATA

1) MALAYALAM 2021

- Initially, the sentences were analysed based on the length of the sentences. On comparing the length of

TABLE 15. These are test sentences from Tamil DravidianLangTech dataset that are mislabelled but predicted rightly by the model.

Example	Label	Prediction
Surya Arya and Mohan Lal wow Kaka Kaka look perfect	Not_offensive	not-Tamil
Mugamoodi part 2	Offensive_Targeted_Insult_Individual	Not_offensive
Why she is looking like sushma sawaraj ji	Offensive_Targeted_Insult_Group	Not_offensive
rendu varushama mukki mukki intha karumaththa than edutha-nungala	Not_offensive -	Offensive_Untargeted
KGF chapter 1 - like Kaithi - like	Offensive_Targeted_Insult_Other	Not_offensive
better luck next time! :D :D :D	Offensive_Targeted_Insult_Other	Not_offensive
karthi enna kidu look aanu.poli mass padamanu.narenum thakarthu	Offensive_Targeted_Insult_Other	not-Tamil
Arjun sir and sk padam yeppadi irukkum	Offensive_Targeted_Insult_Other	Not_offensive
Devidiya punda... eva yaru da gommala	Offensive_Targeted_Insult_Group	Offensive_Untargeted
better luck next time! :D :D :D	Offensive_Targeted_Insult_Other	Not_offensive
Ina pa like low va iruku	Offensive_Targeted_Insult_Other	Not_offensive
Ithokke dislike adikkan vendi kure myrukal	Offensive_Targeted_Insult_Other	not-tamil
Thirumavalavan groups mattum thaan katharitu irukanunga	Offensive_Targeted_Insult_Group	Not_offensive

TABLE 16. These are test sentences from Tamil DravidianLangTech dataset that are misclassified as Not_offensive as they do not contain any offensive words but are labelled as not-Tamil as they do not contain any Tamil words.

Example	Label	Prediction
Annanu angu keralavilum und katta fan base	not-Tamil	Not_offensive
In the Talab polygon Dabang Nagar Pune	not-Tamil	Not_offensive
OMG.. Goose bumps aa gaye	not-Tamil	Not_offensive
Me no so so in love love an	not-Tamil	Not_offensive
Zero faltu hain lekin 2.0 joss hain	not-Tamil	Not_offensive
Racha rambola tala luv u from hyd	not-Tamil	Not_offensive
Vere Level Padam Ith Pwolikkmm Katta Wi8ng	not-Tamil	Not_offensive
Ith polikkum jayan ravi kidukki Frm kerala Waiting miruthan 2	not-Tamil	Not_offensive
Kangana Ranaut ke fans like here	not-Tamil	Not_offensive
I am Kerala Vijay fan njangal und kudea	not-Tamil	Not_offensive

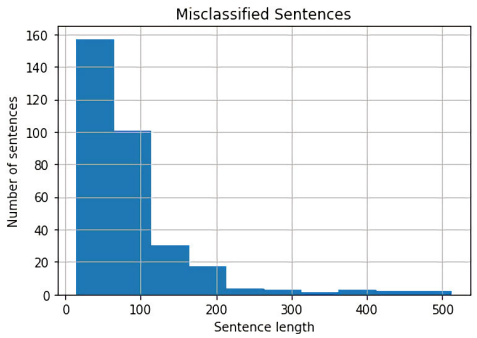

FIGURE 15. The histogram plot of the sentence lengths for the misclassified sentences for the top performing model Malayalam HASOC 2021 dataset.

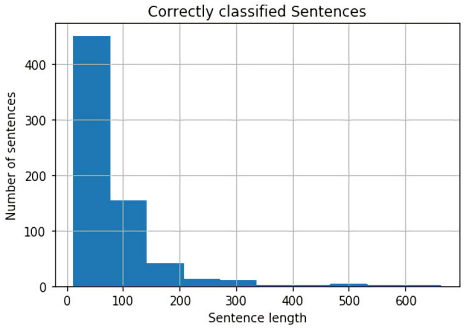

FIGURE 16. The histogram plot of the sentence lengths for the correctly classified sentences for the top performing model of Malayalam HASOC 2021 dataset.

the misclassified sentences and the rightly classified sentences, it was observed that most of the sentence lengths were between 40 and 150. The Figures 15 and 16 show the histogram plots of the sentence lengths for misclassified sentences and the rightly classified sentences.

- The dataset did not have any mislabelling. As a next level, we tried to analyse the data behavior by plotting TSNE and PCA plots. Figures 17 and 18 show the 3D TSNE and PCA scatter plots of the dataset. From the plots, we can observe that no clusters formed, and most of the data points are close or above each other, which makes the classification difficult.
- The next level of analysis was based on word frequency. Figure 19 shows the plot of the most frequent 100 words. The plot shows that the words are very close to the straight line. This shows that most of these 100 words fall into both classes, which makes it difficult for the model to classify. Further, on analysing, we observe that in a total of 6156 words, 579 words fall into both classes. A few of the overlapping words are "'oru', 'aa', 'ee', 'ennu', 'user', 'nalla', 'aanu', 'sir', 'anu', 'pole', 'athu', 'e', 'avan', 'kondu', 'ethu', 'kollanam', 'okke', 'poyi', 'ulla', 'thanne', 'amma', 'video', 'onnu', 'alle', 'bro', 'avane', 'alla', 'nee', 'ariyam', 'interview', 'boby', 'vere', 'pinne', 'onnum', 'koodi', 'illa', 'enna', 'undu', 'ningal', 'thalla'".

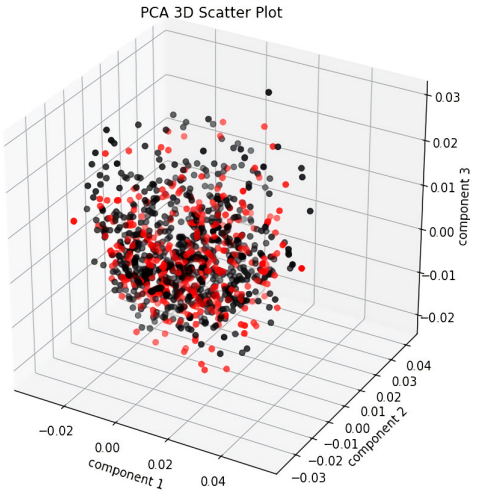

FIGURE 17. 3D PCA plot for Malayalam 2021 HASOC dataset.The figure has the scatter plot for 2 different classes.

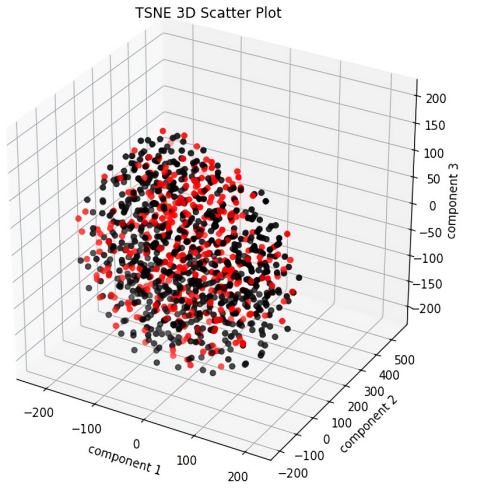

FIGURE 18. 3D TSNE plot for Malayalam 2021 HASOC dataset.The figure has the scatter plot for 2 different classes.

2) MALAYALAM 2020

- Initially, the sentences were analysed based on the length of the sentences. Comparing the length of the misclassified and the rightly classified sentences, it was observed that most of the sentence lengths were between 40 and 150. The Figures 20 and 21 show the histogram plots of the sentence lengths for misclassified sentences and the rightly classified sentences.
- The dataset did not have any mislabelling. As a next level, we tried to analyse the data behavior by plotting TSNE and PCA plots. Figure 22 and Figure 23 shows the 3D TSNE and PCA scatter plots of the dataset. From the plots, we can observe that no clusters were formed, and most data points are close to or above each other, making the classification difficult.
- The next level of analysis was based on word frequency. Figure 24 shows the plot of the most frequent 100 words.

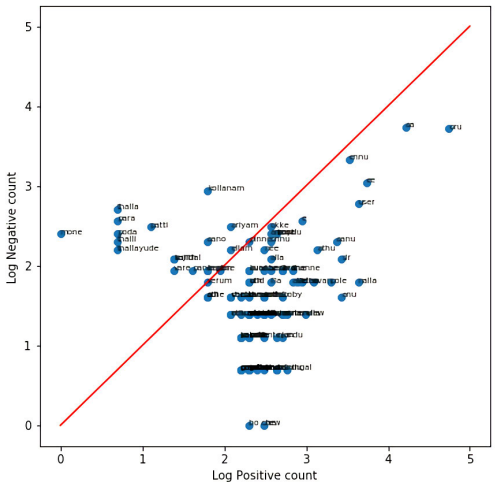

FIGURE 19. Word frequency plot for HASOC Malayalam 2021 dataset.

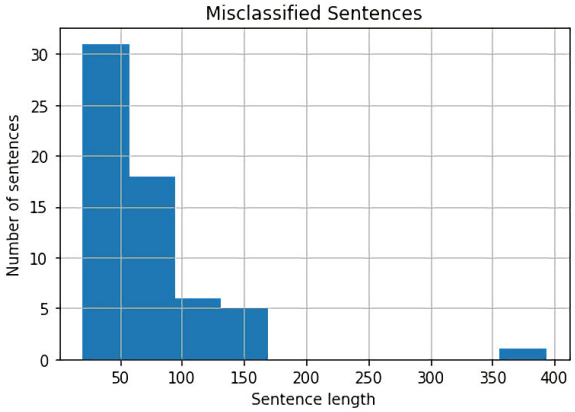

FIGURE 20. The histogram plot of the sentence lengths for the misclassified sentences for the top performing model of HASOC Malayalam 2020 dataset.

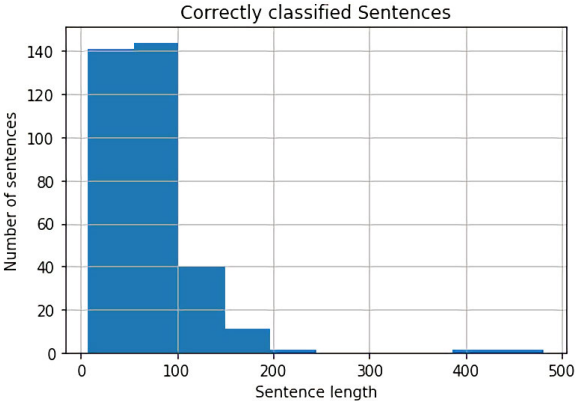

FIGURE 21. The histogram plot of the sentence lengths for the correctly classified sentences for the top performing model of HASOC Malayalam 2020 dataset.

The plot shows that the words are very close to the straight line. This shows that most of these 100 words fall into both classes, which makes it difficult for the

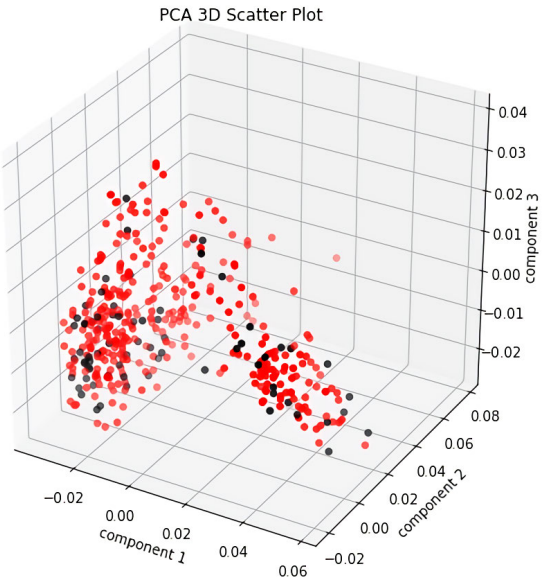

FIGURE 22. 3D PCA plot for Malayalam 2020 HASOC dataset.The figure has the scatter plot for 2 different classes.

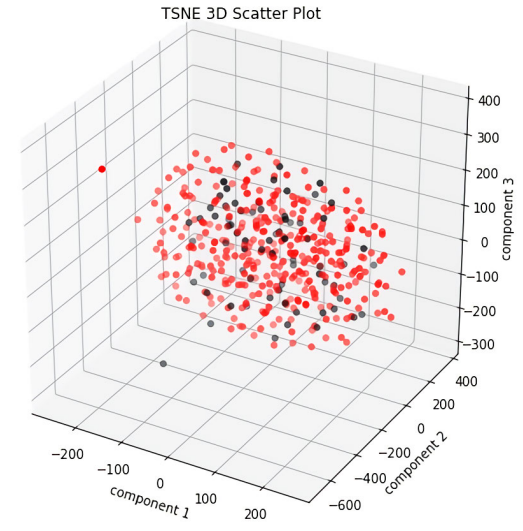

FIGURE 23. 3D TSNE plot for Malayalam 2020 HASOC dataset.The figure has the scatter plot for 2 different classes.

model to classify. Further, on analysing, we observe that in a total of 2012 words, 191 words fall into both classes. A few of the overlapping words are "'ivide', 'pakshe', 'kanda', 'eee', 'release', 'trailor', 'kure', 'comment', 'illa', 'cinemaye', 'hit', 'onnum', 'china', 'collection', 'kaanan', 'ella', 'poyi', 'undo', 'cheyyu', 'kooduthal', 'undallo', 'ne', 'paranju', 'okke', 'poi', 'views', 'million', 'vere', 'polum', 'ningalude', 'look', 'ulla', 'unlike', 'akumo', 'ethu', 'ellam',.

3) TAMIL

- The first level of error analysis was done based on the length of the sentences. Comparing the sentence lengths of the misclassified sentences and the rightly classified sentences showed that most of the sentence lengths were between 60 and 110. The Figures 25 and 26 show the

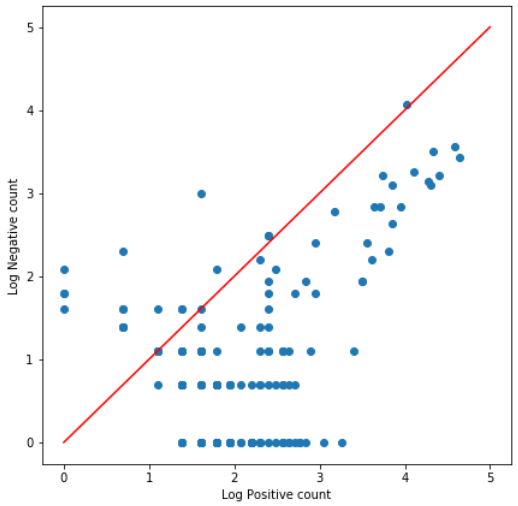

FIGURE 24. Word frequency plot for HASOC Malayalam 2020 dataset.

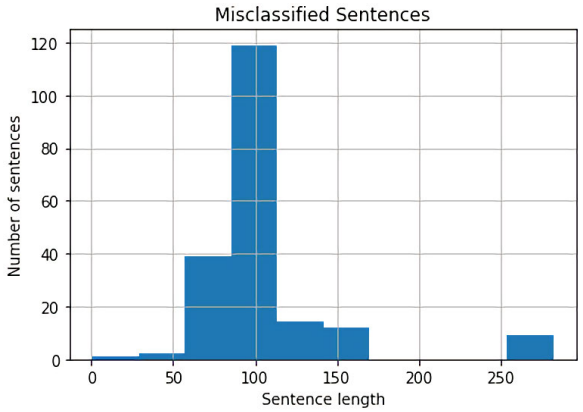

FIGURE 25. The histogram plot of the sentence lengths for the misclassified sentences for the top performing model of Tamil HASOC data.

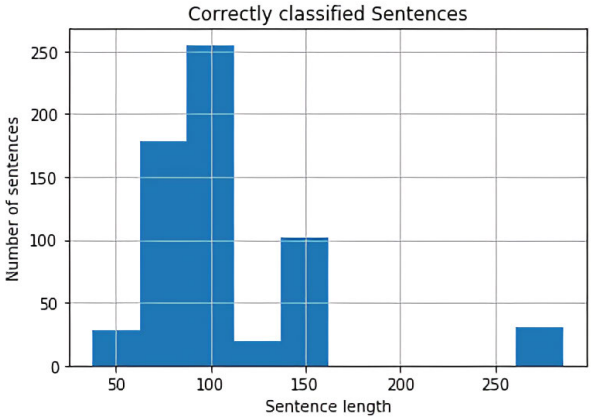

FIGURE 26. The histogram plot of the sentence lengths for the correctly classified sentences for the top performing model of Tamil HASOC data.

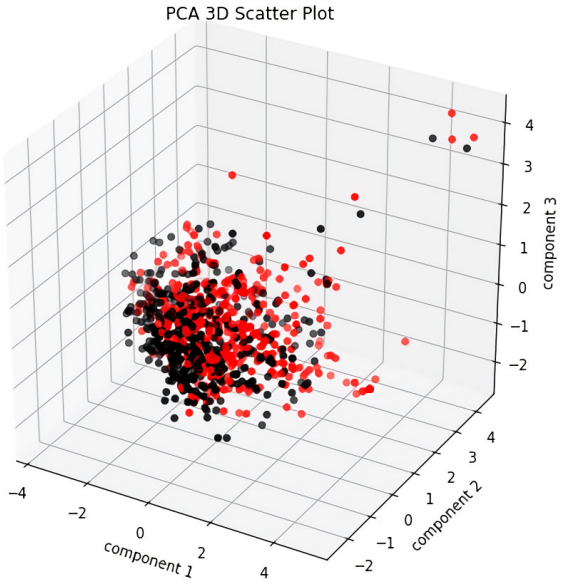

FIGURE 27. 3D PCA plot for HASOC Tamil dataset. The figure has the scatter plot for 2 different classes.

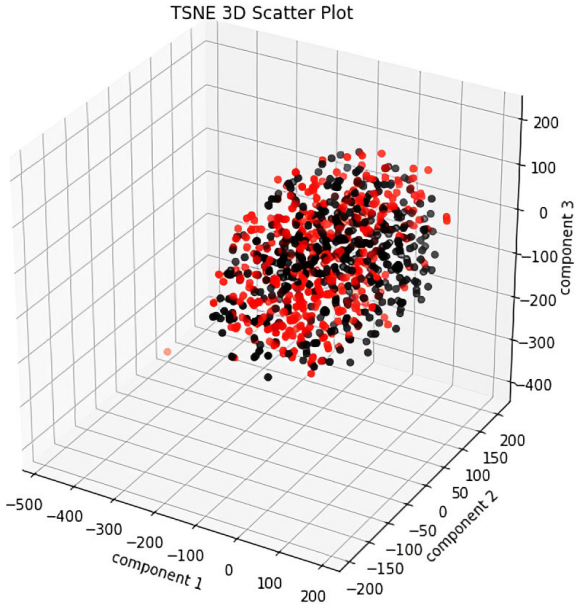

FIGURE 28. 3D TSNE plot for HASOC Tamil dataset. The figure has the scatter plot for 2 different classes.

histogram plots of the sentence length for misclassified sentences and the rightly classified sentences.

- The dataset did not have any mislabelling. As a next level, we tried to analyze the data behavior by plotting TSNE and PCA plots. Figures 27 and 28 show the 3D PCA and TSNE scatter plots of the dataset. From the plots, we can observe that no clusters formed, and most of the data points are close or above each other, which makes the classification difficult.
- The next level of analysis was based on word frequency. Figure 29 shows the plot of the most frequent 100 words.

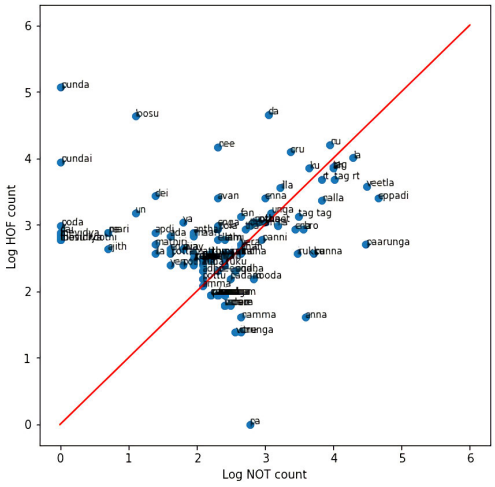

FIGURE 29. Word frequency plot for Tamil HASOC dataset.

The plot shows that the words are very close to the straight line. This shows that most of these 100 words fall into both classes, making it difficult for the model to classify. Further, on analysing, we observe that in a total of 4827 words, 632 words fall into both classes. A few of the overlapping words are "'eppadi', 'la', 'da', 'veetla', 'nu', 'loosu', 'paarunga', 'tag', 'ah', 'tag rt', 'oru', 'rt', 'ku', 'nee', 'nalla', 'illa', 'tag tag', 'panna', 'bro', 'ena', 'enna', 'unga', 'irukku', 'na', 'tweet', 'anna', 'avan', 'intha', 'poi', 'fan', 'thaan', 'dei', 'panni', 'tha', 'enga', 'pola', 'vera', 'dhan'".

From all the above analysis on the three HASOC datasets, we observe that word frequency, sentence length, or mislabelling are not the reasons for the misclassification. This requires further analysis of the linguistics and embedding, which will be done as future work.

VI. CONCLUSION

The increasing spread of abusive language on social media platforms, lack of annotated CodeMix data, and relatively fewer approaches that address CodeMixing in Dravidian languages have impelled us to study various multilingual transformer models and find a single model that works well for CodeMix data. Another major problem in this area is the class imbalance issue. Through this paper, we developed machine learning classifiers for HOS detection using various multilingual transformer-based embedding models by employing a cost-sensitive learning approach to address the class imbalance problem. We compared seven different transformer embedding with Machine Learning classifiers on six CodeMix datasets. Individually observing the performance of embedding on each dataset, DistilBERT was top performing for Tamil DravidianLangTech and Malayalam DravidianLangTech data with F1-scores (weighted) 72 % and 96% respectively, LaBSE for Malayalam 2020 data with a F1-score (weighted) of 92% and MuRIL for Kannada DravidianLangTech, Tamil HASOC and Malayalam

2021 datasets with F1-scores(weighted) 66%, 76% and 68% respectively. Apart from the three datasets, Kannada DravidianLangTech, Tamil HASOC and Malayalam 2021, for which MuRIL gave top performance, the model also had a comparable result on the remaining three datasets. Hence, we observed that MuRIL embedding worked well for all six datasets. We also compared our results with the state-of-the-art models. Out of the compared four datasets, our approach exhibited comparable results with the state-of-the-art work in three datasets without any data translation. We also noticed that MuRIL gave consistent results; hence, we elucidate that MuRIL embedding works well for the CodeMix Dravidian text. Compared with the state-of-the-art works, we obtained better results for two datasets and comparable results for the remaining two. In addition, for all the data, BERT-based embedding with Machine Learning classifiers performed better than BERT-based classifiers. Hence, BERT-based embedding with Machine Learning classifiers has the upper hand over BERT-based classifiers in HOS from Dravidian language tasks. Through the paper, we also introduce a new Malayalam-English CodeMix test set which is an extension of the Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages (HASOC) 2021 Malayalam-English dataset.

REFERENCES

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

[2] S. Anbukkarasi and S. Varadhaganapathy, "Deep learning-based hate speech detection in code-mixed Tamil text," *IETE J. Res.*, pp. 1–6, Mar. 2022.

[3] M. E. Aragon, M. A. A. Carmona, M. M.-Y. Gomez, H. J. Escalante, L. V. Pineda, and D. Moctezuma, "Overview of MEX-A3T at IberLEF 2019: Authorship and aggressiveness analysis in Mexican Spanish tweets," in *Proc. IberLEF@ SEPLN*, 2019, pp. 478–494.

[4] A. Arango, J. Pérez, and B. Poblete, "Hate speech detection is not as easy as you may think: A closer look at model validation (extended version)," *Inf. Syst.*, vol. 105, Mar. 2022, Art. no. 101584.

[5] G. Arora, "INLTK: Natural language toolkit for Indic languages," 2020, *arXiv:2009.12534*.

[6] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," in *Proc. 26th Int. Conf. World Wide Web Companion*, 2017, pp. 759–760.

[7] F. Balouchzahi, B. K. Aparna, and H. L. Shashirekha, "MUCS@DravidianLangTech-EACL2021: COOLI-code-mixing offensive language identification," in *Proc. 1st Workshop Speech Lang. Technol. Dravidian Lang.*, 2021, pp. 323–329.

[8] S. Banerjee, B. Raja Chakravarthi, and J. P. McCrae, "Comparison of pretrained embeddings to identify hate speech in Indian code-mixed text," in *Proc. 2nd Int. Conf. Adv. Comput., Commun. Control Netw. (ICACCCN)*, Dec. 2020, pp. 21–25.

[9] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. R. Pardo, P. Rosso, and M. Sanguinetti, "SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter," in *Proc. 13th Int. Workshop Semantic Eval.* Minneapolis, MN, USA: Association for Computational Linguistics, 2019, pp. 54–63.

[10] D. Benikova, M. Wojatzki, and T. Zesch, "What does this imply? Examining the impact of implicitness on the perception of hate speech," in *Proc. Int. Conf. German Soc. for Comput. Linguistics Lang. Technol.* Cham, Switzerland: Springer, 2017, pp. 171–179.

[11] A. Bohra, D. Vijay, V. Singh, S. S. Akhtar, and M. Shrivastava, "A dataset of Hindi-English code-mixed social media text for hate speech detection," in *Proc. 2nd Workshop Comput. Modeling People's Opinions, Personality, Emotions Social Media*, 2018, pp. 36–41.

[12] B. Raja Chakravarthi, D. Chinnappa, R. Priyadharshini, A. Kumar Madasamy, S. Sivanesan, S. Chinnaudayar Navaneethakrishnan, S. Thavareesan, D. Vadivel, R. Ponnusamy, and P. Kumar Kumaresan, "Developing successful shared tasks on offensive language identification for Dravidian languages," 2021, *arXiv:2111.03375*.

[13] B. R. Chakravarthi, A. Kumar, J. P. McCrae, B. Premjith, K. P. Soman, and T. Mandl, "Overview of the track on hasoc-offensive language identification-dravidiancodemix," in *Proc. FIRE*, 2020, pp. 112–120.

[14] B. R. Chakravarthi, R. Priyadharshini, N. Jose, T. Mandl, P. K. Kumaresan, R. Ponnusamy, R. L. Hariharan, J. P. McCrae, and E. Sherly, "Findings of the shared task on offensive language identification in Tamil, Malayalam, and Kannada," in *Proc. 1st Workshop Speech Language Technol. Dravidian Lang.*, 2021, pp. 133–145.

[15] 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," *Lang. Resour. Eval.*, vol. 56, no. 3, pp. 765–806, Sep. 2022.

[16] B. Raja Chakravarthi, R. Priyadharshini, S. Thavareesan, D. Chinnappa, D. Thenmozhi, E. Sherly, J. P. McCrae, A. Hande, R. Ponnusamy, S. Banerjee, and C. Vasantharajan, "Findings of the sentiment analysis of Dravidian languages in code-mixed text," 2021, *arXiv:2111.09811*.

[17] B. R. Chakravarthi, P. K. Kumaresan, R. Sakuntharaj, A. K. Madasamy, S. Thavareesan, S. C. Navaneethakrishnan, J. P. McCrae, and T. Mandl, "Overview of the HASOC-DravidianCodeMix shared task on offensive language detection in Tamil and Malayalam," in *Proc. Work. Notes FIRE Forum Inf. Retr. Eval.*, 2021, pp. 1–14.

[18] D. Chatzakou, N. Kourtellis, J. Blackburn, E. De Cristofaro, G. Stringhini, and A. Vakali, "Mean birds: Detecting aggression and bullying on Twitter," in *Proc. ACM Web Sci. Conf.*, New York, NY, USA, Jun. 2017, pp. 13–22.

[19] Y. Chen, Y. Zhou, S. Zhu, and H. Xu, "Detecting offensive language in social media to protect adolescent online safety," in *Proc. Int. Conf. Privacy, Secur., Risk Trust Int. Conf. Social Comput.*, Sep. 2012, pp. 71–80.

[20] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, Apr. 1960.

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

[22] R. Dabre, H. Shrotriya, A. Kunchukuttan, R. Puduppully, M. Khapra, and P. Kumar, "IndicBART: A pre-trained model for indic natural language generation," in *Proc. Findings Assoc. Comput. Linguistics: ACL*, 2022, pp. 1849–1863.

[23] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Proc. Int. AAAI Conf. Web Social Media*, vol. 11, 2017, pp. 512–515.

[24] S. L. Devi, "Anaphora resolution from social media text in Indian languages (SocAnaRes-IL): 2nd edition-overview," in *Proc. Forum Inf. Retr. Eval.*, Dec. 2020, pp. 9–13.

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

[26] K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the detection of textual cyberbullying," in *Proc. 5th Int. AAAI Conf. Weblogs Social Media*, 2011, pp. 11–17.

[27] S. Doddapaneni, G. Ramesh, M. M. Khapra, A. Kunchukuttan, and P. Kumar, "A primer on pretrained multilingual language models," 2021, *arXiv:2107.00676*.

[28] S. Dowlagar and R. Mamidi, "Hate speech detection on code-mixed dataset using a fusion of custom and pre-trained models with profanity vector augmentation," *Social Netw. Comput. Sci.*, vol. 3, no. 4, pp. 1–17, Jul. 2022.

[29] S. Edosomwan, S. K. Prakasan, D. Kouame, J. Watson, and T. Seymour, "The history of social media and its impact on business," *J. Appl. Manag. Entrepreneurship*, vol. 1, no. 3, pp. 79–91, 2011.

[30] H. Faris, I. Aljarah, M. Habib, and P. Castillo, "Hate speech detection using word embedding and deep learning in the Arabic language context," in *Proc. 9th Int. Conf. Pattern Recognit. Appl. Methods*, 2020, pp. 453–460.

[31] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang, "Language-agnostic BERT sentence embedding," 2020, *arXiv:2007.01852*.

[32] B. Gambäck and U. K. Sikdar, "Using convolutional neural networks to classify hate-speech," in *Proc. 1st Workshop Abusive Lang. Online*, 2017, pp. 85–90.

[33] N. D. Gitari, Z. Zhang, H. Damien, and J. Long, "A lexicon-based approach for hate speech detection," *Int. J. Multimedia Ubiquitous Eng.*, vol. 10, no. 4, pp. 215–230, Apr. 2015.

[34] T. Gr¨ndahl, L. Pajola, M. Juuti, M. Conti, and N. Asokan, "All you need is 'love' evading hate speech detection," in *Proc. 11th ACM Workshop Artif. Intell. Secur.*, 2018, pp. 2–12.

[35] A. Hande, S. U. Hegde, R. Priyadharshini, R. Ponnusamy, P. K. Kumaresan, S. Thavareesan, and B. R. Chakravarthi, "Benchmarking multi-task learning for sentiment analysis and offensive language identification in under-resourced Dravidian languages," 2021, *arXiv:2108.03867*.

[36] A. Hande, K. Puranik, R. Priyadharshini, S. Thavareesan, and B. R. Chakravarthi, "Evaluating pretrained transformer-based models for COVID-19 fake news detection," in *Proc. 5th Int. Conf. Comput. Methodol. Commun. (ICCMC)*, Apr. 2021, pp. 766–772.

[37] A. Hande, K. Puranik, K. Yasaswini, R. Priyadharshini, S. Thavareesan, A. Sampath, K. Shanmugavadivel, D. Thenmozhi, and B. Raja Chakravarthi, "Offensive language identification in low-resourced code-mixed Dravidian languages using pseudo-labeling," 2021, *arXiv:2108.12177*.

[38] M. A. Hedderich, D. Adelani, D. Zhu, J. Alabi, U. Markus, and D. Klakow, "Transfer learning and distant supervision for multilingual transformer models: A study on African languages," 2020, *arXiv:2010.03179*.

[39] H. Hosseinmardi, S. Arredondo Mattson, R. Ibn Rafiq, R. Han, Q. Lv, and S. Mishra, "Detection of cyberbullying incidents on the Instagram social network," 2015, *arXiv:1503.03909*.

[40] N. Jose, B. R. Chakravarthi, S. Suryawanshi, E. Sherly, and J. P. McCrae, "A survey of current datasets for code-switching research," in *Proc. 6th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Mar. 2020, pp. 136–141.

[41] D. Kakwani, A. Kunchukuttan, S. Golla, A. Bhattacharyya, M. M. Khapra, and P. Kumar, "IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages," in *Proc. Findings Assoc. Comput. Linguistics*, 2020, pp. 4948–4961.

[42] T. Keipi, M. Nasi, A. Oksanen, and P. Räsänen, *Online Hate and Harmful Content: Cross-National Perspectives*. London, U.K.: Taylor & Francis, 2016.

[43] S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, D. K. Margam, P. Aggarwal, R. T. Nagipogu, S. Dave, S. Gupta, S. C. B. Gali, V. Subramanian, and P. Talukdar, "MuRIL: Multilingual representations for Indian languages," 2021, *arXiv:2103.10730*.

[44] S. Koffer, D. M. Riehle, S. Hohenberger, and J. Becker, "Discussing the value of automatic hate speech detection in online debates," in *Proc. Multikonferenz Wirtschaftsinformatik Data Driven X-Turning Data in Value*, Leuphana, Germany, 2018, pp. 1–12.

[45] G. Koushik, K. Rajeswari, and S. K. Muthusamy, "Automated hate speech detection on Twitter," in *Proc. 5th Int. Conf. Comput., Commun., Control Autom. (ICCUBEA)*, Sep. 2019, pp. 1–4.

[46] G. Kovács, P. Alonso, and R. Saini, "Challenges of hate speech detection in social media," *Social Netw. Comput. Sci.*, vol. 2, no. 2, pp. 1–15, Apr. 2021.

[47] R. Kumar, A. K. Ojha, M. Zampieri, and S. Malmasi, "Proceedings of the first workshop on trolling, aggression and cyberbullying (TRAC-2018)," in *Proc. 1st Workshop Trolling, Aggression Cyberbullying*, 2018, pp. 1–14.

[48] P. K. Kumaresan, R. Sakuntharaj, S. Thavareesan, S. Navaneethakrishnan, A. K. Madasamy, B. R. Chakravarthi, and J. P. McCrae, "Findings of shared task on offensive language identification in Tamil and Malayalam," in *Proc. Forum Inf. Retr. Eval.*, Dec. 2021, pp. 16–18.

[49] A. Kunchukuttan, D. Kakwani, S. Golla, A. Bhattacharyya, M. M. Khapra, and P. Kumar, "AI4Bharat-IndicNLP corpus: Monolingual corpora and word embeddings for Indic languages," 2020, *arXiv:2005.00085*.

[50] Y. Kuratov and M. Arkhipov, "Adaptation of deep bidirectional multilingual transformers for Russian language," 2019, *arXiv:1905.07213*.

[51] I. Kwok and Y. Wang, "Locate the hate: Detecting tweets against blacks," in *Proc. 27th AAAI Conf. Artif. Intell.*, 2013, pp. 1621–1622.

[52] S. Malmasi and M. Zampieri, "Detecting hate speech in social media," 2017, *arXiv:1712.06427*.

[53] T. Mandl, S. Modha, A. Kumar, and B. R. Chakravarthi, "Overview of the HASOC track at FIRE 2020: Hate speech and offensive language identification in Tamil, Malayalam, Hindi, English and German," in *Proc. Forum Inf. Retr. Eval.*, Dec. 2020, pp. 29–32.

[54] T. Mandl, S. Modha, P. Majumder, D. Patel, M. Dave, C. Mandlia, and A. Patel, "Overview of the HASOC track at FIRE 2019: Hate speech and offensive content identification in indo-European languages," in *Proc. 11th Forum Inf. Retr. Eval.*, Dec. 2019, pp. 14–17.

[55] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[56] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Advances in pre-training distributed word representations," 2017, *arXiv:1712.09405*.

[57] V. Mujadia, P. Mishra, and D. M. Sharma, "IIIT-hyderabad at HASOC 2019: Hate speech detection," in *Proc. FIRE*, 2019, pp. 271–278.

[58] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive language detection in online user content," in *Proc. 25th Int. Conf. World Wide Web*, 2016, pp. 145–153.

[59] E. Papegnies, V. Labatut, R. Dufour, and G. Linares, "Graph-based features for automatic online abuse detection," in *Proc. Int. Conf. Stat. Lang. Speech Process.* Cham, Switzerland: Springer, 2017, pp. 70–81.

[60] B. Pariyani, K. Shah, M. Shah, T. Vyas, and S. Degadwala, "Hate speech detection in Twitter using natural language processing," in *Proc. 3rd Int. Conf. Intell. Commun. Technol. Virtual Mobile Netw. (ICICV)*, Feb. 2021, pp. 1146–1152.

[61] J. Ho Park and P. Fung, "One-step and two-step classification for abusive language detection on Twitter," 2017, *arXiv:1706.01206*.

[62] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[63] F. M. Plaza-del-Arco, M. D. Molina-González, L. A. Ureña-López, and M. T. Martín-Valdivia, "Comparing pre-trained language models for Spanish hate speech detection," *Exp. Syst. Appl.*, vol. 166, Mar. 2021, Art. no. 114120.

[64] M. Ptaszynski, A. Pieciukiewicz, and P. Dybała, "Results of the PolEval 2019 shared task 6: First dataset and open shared task for automatic cyberbullying detection in Polish Twitter," Inst. Comput. Sci., Polish Acad. Sci., Warsaw, Poland, Tech. Rep., 2019, pp. 89–110.

[65] S. T. Roberts, J. Tetreault, V. Prabhakaran, and Z. Waseem, "Proceedings of the third workshop on abusive language online," in *Proc. 3rd Workshop Abusive Lang. Online*, 2019, pp. 1–16.

[66] J. P. McCrae, S. Banerjee, and B. R. Chakravarthi, "Comparison of pretrained embeddings to identify hate speech in Indian codemixed text," in *Proc. 2nd Int. Conf. Adv. Comput., Commun. Control Netw. (ICACCCN)*, 2020, pp. 21–25.

[67] D. Saha, N. Paharia, D. Chakraborty, P. Saha, and A. Mukherjee, "Hate-Alert@DravidianLangTech-EACL2021: Ensembling strategies for transformer-based offensive language detection," 2021, *arXiv:2102.10084*.

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

[69] S. Sharma, S. Agrawal, and M. Shrivastava, "Degree based classification of harmful speech using Twitter data," 2018, *arXiv:1806.04197*.

[70] V. K. Singh, S. Ghosh, and C. Jose, "Toward multimodal cyberbullying detection," in *Proc. CHI Conf. Extended Abstr. Hum. Factors Comput. Syst.*, May 2017, pp. 2090–2099.

[71] D. Sivalingam and S. Thavareesan, "OffTamil@DravideanLangTech-EASL2021: Offensive language identification in Tamil text," in *Proc. 1st Workshop Speech Lang. Technol. Dravidian Lang.*, 2021, pp. 346–351.

[72] K. Sreelakshmi, B. Premjith, and K. P. Soman, "Amrita cen at HASOC 2019: Hate speech detection in Roman and devanagiri scripted text," in *Proc. FIRE*, 2019, pp. 366–369.

[73] K. Sreelakshmi, B. Premjith, and K. P. Soman, "Detection of hate speech text in hindi-english code-mixed data," *Proc. Comput. Sci.*, vol. 171, pp. 737–744, Jan. 2020.

[74] J. M. Struß, M. Siegel, J. Ruppenhofer, M. Wiegand, and M. Klenner, "Overview of GermEval task 2, 2019 shared task on the identification of offensive language," in *Proc. 15th Conf. Natural Lang. Process. (KONVENS)*. Nuremberg, Germany: German Society for Computational Linguistics, 2019, pp. 354–365.

[75] D. Tula, P. Potluri, S. Ms, S. Doddapaneni, P. Sahu, R. Sukumaran, and P. Patwa, "Bitions@DravidianLangTech-EACL2021: Ensemble of multilingual language models with pseudo labeling for offence detection in Dravidian languages," in *Proc. 1st Workshop Speech Lang. Technol. Dravidian Lang.*, 2021, pp. 291–299.

[76] C. Vasantharajan and U. Thayasivam, "Towards offensive language identification for Tamil code-mixed Youtube comments and posts," *Social Netw. Comput. Sci.*, vol. 3, no. 1, pp. 1–13, Jan. 2022.
[77] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5998–6008.
[78] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter," in *Proc. NAACL Student Res. Workshop*, 2016, pp. 88–93.
[79] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, 32, 2019, pp. 1–11.
[80] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval)," 2019, *arXiv:1903.08983*.
[81] Z. Zhang, D. Robinson, and J. Tepper, "Detecting hate speech on Twitter using a convolution-GRU based deep neural network," in *Proc. Eur. Semantic Web Conf.* Cham, Switzerland: Springer, 2018, pp. 745–760.

K. SREELAKSHMI is currently pursuing the Ph.D. degree in offensive language identification with Social Media Text. She is also an Assistant Professor with the Centre for Computational Engineering and Networking, Amrita Vishwa Vidyapeetham, Coimbatore, India. She has authored several national and international publications. Her research interests include offensive language detection from code-mixed social media text, machine learning, deep learning, and artificial intelligence.

B. PREMJITH is currently an Assistant Professor (Senior Grade) with the Amrita School of Artificial Intelligence, Amrita Vishwa Vidyapeetham, Coimbatore, India. He has published papers in reputed journals and conferences. His research interests include natural language processing, computational linguistics, and computational social science. He secured first place in international competitions on factuality analysis of the text in Iberian languages and abusive language detection in Tamil.

BHARATHI RAJA CHAKRAVARTHI is currently a permanent Lecturer above the bar with the School of Computer Science, University of Galway, Ireland. He involved on multimodal machine learning, abusive/offensive language detection, bias in natural language processing tasks, inclusive language detection, and multilingualism. He has published papers in highly reputed journal articles (*LRE*, *CSL*, *MTAP*, *SNAM*, *JDSA*, *JDIM*, and *IJIM Data Insights*) and multiple international conference papers (COLING, LREC, MTSUMMIT, DSAA, LDK, GWC, AICS, and FIRE). He received the Best Application Paper Award at DSAA 2020 IEEE and ACM-funded conference. He is the Area Chair of the 17th Conference of the European Chapter of the Association for Computational Linguistics 2023 and the General Chair of SPELLL 2022 and 2023. He is an Associate Editor of *Expert Systems with Applications* (Elsevier) and an Editorial Board Member of *Computer Speech and Language* (Elsevier).

K. P. SOMAN is currently the Head and a Professor with the Center for Computational Engineering and Networking (CEN) and the Dean of the Amrita School of Artificial Intelligence, Amrita Vishwa Vidyapeetham, Coimbatore, India. He has more than 25 years of research and teaching experience in artificial intelligence and data science related subjects with the Amrita School of Engineering, Coimbatore. He has around 450 publications to his credit in reputed journals, such as IEEE TRANSACTIONS, IEEE ACCESS, *Applied Energy*, and conference proceedings. He published four books, namely, *Insight Into Wavelets: from Theory to Practice*, *Insight Into Data Mining: Theory and Practice*, *Support Vector Machines and Other Kernel Methods*, and *Signal and Image Processing-The Sparse Way*. He is the most cited author in Amrita Vishwa Vidyapeetham in the areas of artificial intelligence and data science. He was listed among the Top-10 Computer Science Faculty by DST, Government of India, from 2009 to 2013, the Career 360 and MHRD, from 2017 to 2018, and also in the list of the most prolific authors in the world, prepared by Springer Nature.