Interdisciplinary Journal of Information, Knowledge, and Management

An Official Publication of the Informing Science Institute InformingScience.org

IJIKM.org

Volume 19, 2024

ENHANCED DETECTION OF HATE SPEECH IN DRAVIDIAN LANGUAGES IN SOCIAL MEDIA USING ENSEMBLE TRANSFORMERS

Arunachalam V	School of Computer Science and Engineering, Vellore Institute of Technology – Chennai, India	arunachalam3342@gmail.com
Maheswari N *	School of Computer Science and Engineering, Vellore Institute of Technology – Chennai, India	maheswari.n@vit.ac.in

* Corresponding author

ABSTRACT

Aim/Purpose	This study aims to propose an efficient implementation of identifying and detecting offensive and hate comments on social media platforms for Dravidian languages.
Background	There has been a notable increase in hate comments on social media platforms in recent years. Hate language and hate speech are expressions of conflicts that arise between different groups, both within and across civilizations. This toxic behavior has a detrimental impact on individuals, resulting in disagreements, arguments, political conflicts, and even mental health issues such as cyberbullying, depression, and anxiety. In recent years, research on offensive detection has expanded beyond English and Hindi languages to include other languages such as Urdu, Spanish, Arabic, and more. An ongoing trend in modern research involves gathering data from prominent social media platforms such as YouTube, Instagram, and Twitter to train models in proposed studies.
Methodology	The objective of this work is to identify and detect offensive or hateful comments on social media platforms dedicated explicitly to the Dravidian languages - Tamil, Kannada, and Malayalam. The dataset, HASOC-Offensive Language Identification track in Dravidian Code-Mix FIRE 2021, undergoes several proposed preprocessing procedures on the YouTube comments. Tokens were created to represent the attributes. The pre-trained models utilizing transformers

Accepting Editor Geoffrey Z. Liu | Received: July 7, 2024 | Revised: October 2, October 10, October 21, November 2, November 15, 2024 | Accepted: November 18, 2024.

Cite as: V, Arunachalam., & N, Maheswari.. (2024). Enhanced detection of hate speech in Dravidian languages in social media using ensemble transformers. *Interdisciplinary Journal of Information, Knowledge, and Management, 19,* Article 39. https://doi.org/10.28945/5403

	were trained using tokenized data. The efficacy of different binary classifiers has been assessed and analyzed using the embedded vectors derived from the models. The mBert with the classifier model CATBOOST with GSCV achieved F1 scores for Tamil, Malayalam, and Kannada are 0.94, 0.98, and 0.82, respectively. Precision and recall values for Tamil, Malayalam, and Kannada are 0.94, 0.98, and 0.82, and 0.94, 0.98, and 0.83, respectively.
Contribution	This research produced an approach to improve the results and F1-score by performing an improved preprocessing method using stemming and stopword removal. The native and English-coded comment data used in this work were not discussed much in the previous works.
Findings	The findings of this study indicated that the preprocessing work combined both the native language data and the English-coded language. It describes the improved models and performance of the results in order to detect hate speech.
Recommendations for Practitioners	This study is expected to be valuable for social media platforms and other review sites to separate offensive comments from good and bad ones. This is also valuable for content creators and users in improving their ideas and content and also helps in preventing cyber harassment and bullying.
Recommendations for Researchers	This study discussed the use of stemming and stopword removal and how it improved the detection of hate comments for both native and codemix comments in social media.
Impact on Society	This work helps the users and community explore the positive side of the internet and allows content creators to share their ideas without undermining the benefits of online interactions.
Future Research	Future research will explore the uniqueness of the linguistics and slang of these three languages. Apart from these languages, future research will be conducted on North-Indian Indian languages like Bengali, Marathi, and Gujarati.

Keywords	BERT, NLP, F1-Score, Dravidian languages, Code-mix, Hate comment, Muril, mBERT

INTRODUCTION

Social media platforms have transformed global communication and content sharing, presenting both opportunities and challenges in the digital sphere (Rehan et al., 2023). The comment section is an essential feature of these platforms, providing users with a dynamic forum to share their thoughts and participate in discussions. Despite the exchange of ideas, there is a troubling trend of harmful comments, such as hate speech and harassment, that diminish the positive aspects of online interactions (Sengupta et al., 2022). Despite these challenges, comments are essential for building community and promoting discussion. It is crucial to tackle the widespread offensive content to create a secure digital space where users are shielded from derogatory comments and can engage in productive discussions. Social media companies can continue to play the important role of promoting meaningful engagement and a positive online environment by actively addressing hate speech and inappropriate language.

The proliferation of hate speech on social media is widely recognized to have negative consequences for society at large. Given its widespread reach and lack of regulation, addressing this issue poses a considerable challenge. In numerous nations, hate speech on social media is unlawful unless it specifically targets a particular group or encourages criminal activity (Subramanian, Adhithiya, et al., 2022).

Hate comments negatively affect content quality and user experience on social media platforms, diminishing the intended message and discouraging productive interaction. Offensive language impedes idea exchange, deteriorating meaningful conversations and reducing content influence. This hostile environment challenges audience engagement and retention, alienating supportive users and discouraging active participation. As users disengage due to the toxic atmosphere, the online community's vibrancy, growth, and diversity are compromised, hindering overall content retention for creators.

Hate comments significantly impact user experience, leading to discomfort, frustration, and psychological distress. Such remarks undermine the sense of community on social media, discouraging users from fully engaging in conversations and stifling the exchange of ideas. Persistent exposure to offensive content fosters a toxic environment, deterring new users and posing challenges for platform administrators in maintaining an inclusive space while effectively identifying and removing harmful content.

Hate language refers to offensive, vulgar, profane, insulting, or threatening material directed at someone. Manually detecting abusive content on social media platforms like Instagram, Reddit, and Twitter is no longer feasible (Rehan et al., 2023). So, the research involves developing a solution for swift and efficient identification of offensive content and words in the diverse linguistic landscape of Indian languages. The model needs to be systematically adjusted to effectively identify inappropriate words and content, thereby achieving the highest possible F1 score by considering the complex linguistic nuances and cultural context specific to Indian languages.

The Dravidian languages encompass a linguistic familial group predominantly utilized in the southern regions of India and in the territories of Sri Lanka, Pakistan, Nepal, and Afghanistan. These languages exhibit distinct grammatical, lexical, and morphological characteristics that set them apart from the Indo-European languages spoken in northern India. Tamil, Telugu, Kannada, and Malayalam are major Dravidian languages with unique literary traditions and cultural heritages (Rajalakshmi et al., 2023). Developing models for Dravidian languages is difficult and requires a nuanced approach due to various key factors. The limited availability of data for these languages presents a significant challenge for training robust models. Additionally, the current data frequently experiences class imbalance in labeled datasets, which makes model training and evaluation more challenging. The data poses a distinctive challenge due to its combination of native language and code-mix formats commonly found in digital communication.

This research aims to reduce abusive comments for Indian Dravidian languages on social media platforms. Harmful content detection is a very important part of keeping social media sites safe. These systems read through comments to find and remove potentially harmful content. This can be done using natural language processing and machine learning. This proactive approach not only keeps users from seeing offensive comments but also creates a space where helpful conversation can grow. Making offensive content detection tools better all the time shows a dedication to creating a digital world that cares about users' health and promotes positive online interaction.

The structure of the work is described in the following. The next section presents a comprehensive examination of the existing research and associated research on detecting and categorizing offensive and hate language. An overview of the dataset employed in the present study is then provided. The following section is dedicated to the examination of transformer and classifier models, as well as the methodologies utilized for the detection of hate speech in Dravidian languages. A thorough examination of the behavior and outcomes of the models is then undertaken, accompanied by a comparative evaluation with alternative methodologies. The final section provides a comprehensive overview of the study and delves into possibilities for future investigation on the identification of offensive language in Dravidian languages.

RELATED WORKS

Over the past few years, there has been a noticeable rise in the number of instances in which offensive language has been utilized on various social media platforms. Offensive language and hate speech serve as manifestations of conflicts among diverse groups both within and beyond civilizations (Chakravarthi et al., 2023). In recent years, research on offensive detection has not only been done for English and Hindi languages (Sreelakshmi et al., 2020) but also for other languages like Urdu (Malik, Cheema, & Ignatov, 2023), Spanish (Molero et al., 2023), Arabic (El-Alami et al., 2022), Turkish (Özberk & Çiçekli, 2021), and so forth. A prevailing trend in current research involves collecting data from prominent social media platforms like YouTube and Twitter to train models in proposed studies. The Dravidian languages in India constitute a major language family primarily spoken in the southern part of the country. Tamil, Kannada, and Malayalam are among the languages that have historical roots and similar linguistic characteristics. The majority of studies in this domain employ datasets sourced from these platforms to develop and validate their research frameworks (Chakravarthi et al., 2022).

DRAVIDIAN LANGUAGE

Rajalakshmi et al. (2023) proposed a combination of stemming and stopword removal techniques to preprocess the Tamil text data. Subsequently, they employed various embedding techniques, such as different BERT models and the monolingual model TaMillion, to determine the most effective representation for Tamil text. The methodology employed yielded promising results, as the proposed approach achieved an F1 score of 84% and an accuracy of 86% in its ability to detect offensive content in Tamil YouTube comments accurately. Subramanian, Ponnusamy, et al. (2022) developed models that could effectively identify offensive language in Tamil YouTube comments, with a particular focus on code-mixed data. The study attained a precision rate of 81.651% in machine learning using the K-Nearest Neighbors (KNN) algorithm. However, when employing transformer-based models, XLM-RoBERTa (Large) exhibited superior performance, achieving an accuracy of 88.532%.

A recent study by Hande et al. (2022) examined sentiment analysis and the identification of offensive language in code-mixed YouTube comments in Tamil, Malayalam, and Kannada languages. The study employed a multi-task learning technique to tackle the problem of inadequate labeled data for code-mixed datasets created by users. The F1-Score of the mBERT model was found to be the highest in Kannada, Malayalam, and Tamil, with respective scores of 66.8%, 90.5%, 59%, 70%, and 62.1%, 75.3%. The deep ensemble framework proposed by Roy et al. (2022) integrates conventional machine learning, deep learning, and transformer-based models to accurately differentiate between offensive and non-offensive text in code-mixed languages. In comparison to the baselines, the deep ensemble framework demonstrates superior performance by attaining weighted F1 scores of 0.802 for Malayalam and 0.933 for Tamil code-mixed datasets.

Malik, Nazarova, et al. (2023) introduced a novel approach for dataset conversion in their study on Multilingual Hope Speech Detection (MHSD). The research developed a theoretical framework that utilized transfer learning and fine-tuning techniques on RoBERTa models to detect messages related to hope in both English and Russian languages. By integrating Russian-RoBERTa into the translation-based approach, remarkable results were achieved, with a 94% accuracy rate and an 80.24% F1 score. The utilization of deep learning for code-mix Malayalam language was examined by Pillai and Arun (2024), with a specific focus on feature fusion and detection approaches. The aforementioned methodology resulted in an accuracy rate of 0.956, a sensitivity rate of 0.939, and a specificity rate of 0.978. Mozafari et al.'s (2022) research centered on the identification of features in a cross-lingual few-shot situation, employing a meta-learning approach. The performance of the meta-learning models MAML and Proto-MAML was found to be significantly superior to that of transfer learning-based models. Mridha et al. (2021) proposed L-Boost, which combines modified AdaBoost with a BERT transformer, to detect offensive content in the Bengali language. The study highlights the increasing prevalence of informal, unorganized, and offensive written content in Bengali and Banglish

on various social media platforms. Wadud et al. (2022) introduced LSTM-BOOST, which achieves an accuracy of 88% in identifying Bengali text. In Lora et al.'s (2023) study, attention is redirected toward the identification of sarcasm using a transformer-based generative adversarial model. This model demonstrates an accuracy of 77.1% and 97.2% when applied to the "Ben-Sarc" and "BanglaSarc" datasets, respectively.

Recent Innovation Methods

Recent strides in offensive comment detection involve innovative methods like combining CNN with BERT (Mehta et al., 2022), integrating LSTM with BERT (Kumar et al., 2021), and utilizing cross-lingual embeddings and a stacked encoder architecture (Sundar et al., 2022). These approaches leverage transfer learning techniques, using pre-trained language models from the iNLTK library, including ULMFiT, to fine-tune language models and enhance accuracy and efficiency in offensive comment detection. Several researchers have also focused on developing models for detecting homophobic, offensive comments on social media. A novel dataset was created by Lavanya and Navaneethakrishnan (2022) to tackle the dearth of knowledge and engagement with the LGBTQ community in the Tamil language. The article by Chakravarthi (2024) examines the identification of homophobic and transphobic language within YouTube comments on the same language. Comparably, the task of addressing Malayalam and Hindi is also addressed by Kumaresan et al. (2023). A summary is presented in the Appendix.

METHODS

The objective of the study is to identify and detect hateful or offensive remarks on social media platforms specifically focused on the Dravidian languages, namely Tamil, Kannada, and Malayalam. The YouTube comments undergo a series of preprocessing procedures on the dataset (Chakravarthi et al., 2022). The dataset was divided into a 70:30 ratio, with 30% going toward testing and 70% going toward training.

Tokens were created to represent the attributes. The transformer-based pre-trained models underwent training using tokenized data. Using both English-coded and native language data enables transformer models to learn cross-linguistic representations, capturing how similar concepts are expressed across languages. This leads to more generalized embeddings, improving the model's performance in multilingual and mixed-language tasks. Additionally, given the linguistic similarities among Dravidian languages, models trained on one language can be transferred to another, producing effective results to a certain extent. The evaluation and examination of the performance of different binary classifiers have been conducted by utilizing the embedded vectors derived from the models. Figure 1 illustrates the structure and architecture of the proposed methodology. Figure 2 displays the algorithm.

Data Preprocessing

In this section, an explanation is provided regarding the preprocessing techniques employed on the Dravidian text data. The techniques encompass the elimination of stopwords and the implementation of stemming. A more advanced stemming algorithm was utilized to analyze the morphology of these languages due to their complex structure. The preprocessing steps involve grouping similar offensive labels into a single category and eliminating the last non-Dravidian language label, resulting in a binary classification for the dataset. In preprocessing English text, several steps are undertaken: removal of punctuations and extra spaces, elimination of special characters while converting to lowercase, and removal of stopwords. Additionally, emoticons are converted to text using the demoji library. These processes streamline the text for further analysis, ensuring consistency and reducing noise in the data. For Tamil, Malayalam, and Kannada languages, the words with minor spelling mistakes and grammatical errors were rectified. Also, some of the English-coded words were translit-

erated to native language words to improve the meaning and content of the comment, thereby improving the text representation. The dataset was divided into a 70:30 ratio, with 30% going toward testing and 70% going toward training.

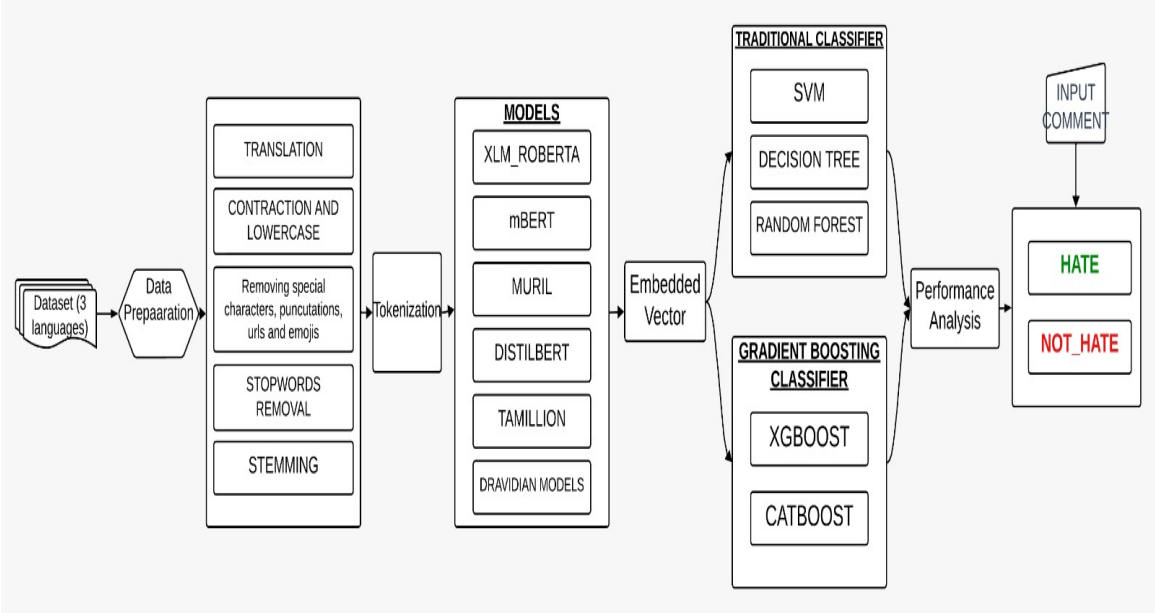

Figure 1. Architecture of the proposed method

Algorithm 1: An enhanced approach to identify the hate comments in Dravidian languages from social media using Ensemble Transformers and Classifiers

```
1.  procedure Hate-Comment-Detection (tamil-dataset, malayalam-dataset, kannada-dataset)
2.      D1 <- PreProcessing(tamil-dataset);          #preprecessing of Tamil dataset
3.      D2 <- PreProcessing(malayalam-dataset);      #preprocessing of Malayalam dataset
4.      D3 <- PreProcessing(kannada-dataset);        #preprocessing of Kannada dataset
5.      F1-Score-Tamil [];                           #to store F1 score for Tamil
6.      F1-Score-Malayalam [];                       #to store F1 score for Malayalam
7.      F1-Score-Kannada [];                         #to store F1 score for Kannada
8.      Models= [XLM-Roberta, Muril, DistilBERT, mBERT, Tamillion, Tamil-BERT, Malayalam-BERT,
    Kannada-BERT];                                   #Models in List
9.      for 0<=i<4 do
10.         F1-Score-Tamil[i] <- Classification (D1, Models[i]);
11.  F1-Score-Malayalam[i] <- Classification (D2, Models[i]);
12.  F1-Score-Kannada[i] <- Classification (D3, Models[i]);
13.         i <- i+1;
14.     End for;                                     # for loop to get the F1-scores of the
    models
15.     F1-Score-Tamil[4] <- Classification (D1, Models[5]);   #F1-score for Tamil-BERT
16.     F1-Score-Malayalam[4] <- Classification (D2, Models[6]);   #F1-score for Malayalam-BERT
17.     F1-Score-Kannada[4] <- Classification (D3, Models[7]);     #F1-score for Kannada-BERT
18.  F1-Score-Tamil[5] <- Classification (D1, Models[4]);       #F1-score for Tamillion
19. end procedure
```

```
1.  procedure PreProcessing (Dataset)
2.      D1 <- Cleaning1 (Dataset);      # Removing URL, punctuations, emoji, special characters
3.      D2 <- Cleaning2 (D1);           # Lowercase conversion and contraction
4.      D3 <- Stopwords (D2);           # Removing Stopwords
5.      D4 <- Stemming (D3);            # Stemming
6.      Return D4;
7.  end procedure
```

```
1.  procedure Classification (Dataset D, model)
2.      Token <- AutoTokenizer (model); #AutoTokenizer to assign tokenizer to the model
3.      Model <- fine-tuning (D, model, 70-30); # fine-tuning the model in 70:30 ratio
4.      Confusion-Matrix <- generate-results (Model); #generate confusion matrix
5.      F1-Score <- compute-F1 (Confusion-Matrix); #computing F1-score from confusion matrix
6.      Return F1-Score;
7.  end procedure
```

Figure 2. Algorithm of the proposed method

Stopwords removal

Stopwords are commonly eliminated words that occur frequently in a given language while performing tasks related to natural language processing. Stopwords are typically characterized by their lack of substantial meaning or their limited contribution to the overall comprehension of a text. By removing stopwords, computational resources can be directed towards more significant words and phrases, leading to a substantial decrease in irrelevant information in the dataset. By reducing the size of the dataset, this procedure not only enhances computational efficiency but also enhances the performance and F1 score of the model and the work. Tasks such as sentiment analysis, topic modeling, and information retrieval improve their accuracy by giving priority to relevant content while disregarding unnecessary words.

The total number of stopwords varies across different languages. For instance, the list of stopwords in English includes words that are used frequently, such as "a" and "to," whereas in languages such as Tamil or Hindi, the list may include terms that are unique to those languages and reflect their grammar and syntax. Here is an example of stopwords removal in a Tamil sentence: "என் பள்ளியில் பல மாணவர்கள் உள்ளன." (Translation: "There are many students in my school.") Now, if we remove the stopwords like "என்" (my), "பல" (many), and "உள்ளன" (there are), we get a shorter, clearer sentence: "பள்ளியில் மாணவர்கள்" (Translation: "Students in school"). By eliminating stopwords, the algorithm can enhance its concentration on the significant substance of the text. A total of 125 stopwords were eliminated from the Tamil language, 57 from the Malayalam language, and 152 from the Kannada language as a result of this specific work.

Stemming

The process of stemming is a fundamental technique employed in the field of natural language processing (NLP) to preprocess textual data. Stemming is the act of reducing words to their fundamental or foundational form, known as the stem. This is accomplished through the elimination of suffixes and prefixes. The aforementioned procedure enables the standardization of words that possess comparable semantics but exhibit distinct grammatical structures, thereby optimizing the examination of textual information. Words like "running," "run," and "ran" would all be simplified to the original stem "run."

Stemming is a process that typically involves eliminating prefixes or suffixes from words to derive their base form, which may be a shortened word that may not necessarily be legitimate in the given language. In the Tamil language, for example, the stemmer can convert the word "படிக்கிறான்"

(which means "reading") into "படி" (the stem form), which is not a valid word on its own. A good example of Tamil stemming where stemming produces meaningful Tamil words is the word "அறிவுக்கலன்" (meaning "educationist" or "scholar"). When subjected to a Tamil stemming algorithm, the word can be simplified to the stem "அறிவு" which signifies "knowledge" or "learning". Although truncated, the word "அறிவு" still retains its significance and validity in Tamil, effectively conveying the essence of the original term associated with education or knowledge. This demonstrates the ability of stemming to generate meaningful Tamil words even when they are reduced to their root form.

Stemming is a linguistic technique used to reduce the various forms of a word to its root form. Inflection refers to the modification of a word to convey different grammatical features, including tense, case, voice, aspect, person, number, gender, and mood (Rajalakshmi et al., 2023). Stemming simplifies vocabulary and enhances the efficiency of tasks like text classification, information retrieval, and sentiment analysis by reducing words to their basic root form. Despite the potential advantages stemming from Dravidian languages, particularly Kannada and Malayalam, it is uncommon and rarely implemented. The morphological analyzer (Kunchukuttan, 2020) is an advanced tool that was used for this work, and it conducts an in-depth analysis of word morphology using linguistic rules and language-specific resources. Linguistic morphology is the study of the grammatical elements of words, including stems, affixes, and inflections, which allows for a more comprehensive comprehension of word formation.

Feature Representation

Bidirectional Encoder Representations from Transformers (BERT) stands as a groundbreaking milestone in the realm of NLP, intricately intertwined with the Transformer architecture. The Transformer architecture, introduced by Vaswani et al. (2017), transformed NLP by utilizing self-attention mechanisms to enable models to assess the importance of different words in a sentence while processing. BERT enhances this innovative structure by incorporating bidirectional comprehension to understand text in both left and right contexts. Its ability to process text non-sequentially represents a significant advancement from previous models, allowing it to grasp complex contextual nuances essential for a comprehensive understanding of language.

The BERT architecture comprises fundamental elements, such as the transformer encoder layers, which can be accurately characterized through mathematical equations that depict the self-attention mechanism and feed-forward neural network layers.

BERT operates through a series of important phases for text classification. The input text undergoes tokenization, a process that involves dividing it into distinct tokens. These tokens can represent either individual words or subwords. The inclusion of special tokens ([CLS] and [SEP]) serves to indicate the beginning and end of sentences. The input is subjected to tokenization and subsequently encoded using the BERT model, which comprises several transformer encoder layers. BERT improves the representation of each token in its layers through self-attention mechanisms, enabling it to effectively capture contextual relationships between words. A classification head is typically appended to BERT after encoding, with the [CLS] token representation commonly utilized for classification. This token holds the contextual details of the entire input sequence, which helps in precise classification. The complete BERT model adjusts its parameters to improve performance by fine-tuning labeled data specific to the classification task. Backpropagation is utilized in fine-tuning to minimize a loss function and enhance the model's text classification accuracy by iteratively refining it. The model analyses tokenized new text inputs. The output is used by the classification head to predict class labels or probabilities. BERT's transformer architecture allows it to understand complex contextual nuances, making it an effective text classifier when fine-tuned with labeled data. BERT's versatility and expertise have made it an NLP research leader in text understanding and categorization.

Transformer models

The following transformer models were employed to derive the embedded vectors.

MBERT

mBERT is an expanded version of the original BERT model, designed to tackle the complexities of processing multilingual text. The model is trained on a dataset consisting of the 104 most commonly spoken languages globally, using the masked language modeling objective. The process of multilingual pre-training improves mBERT's ability to identify and understand linguistic features and structures in various languages. The model acquires the ability to generate language-independent representations that can effectively handle text in different languages, irrespective of linguistic differences or complexities. mBERT, which functions as a universal language encoder, can represent text from different languages in a common feature space. In different languages, mBERT can generalize and transfer knowledge, ensuring consistent and strong performance.

DISTILBERT

DistilBERT is a distilled version of the BERT model, designed to be more lightweight and computationally efficient while retaining much of BERT's performance. The model is created by Hugging Face and is designed to minimize the computational resources needed to implement large transformer-based models in practical settings. DistilBERT's primary innovation lies in its distillation process, wherein the original BERT model is condensed into a more compact form while preserving its fundamental knowledge and functionalities. Knowledge distillation, parameter pruning, and quantization are employed techniques to achieve this compression. The model is a condensed and optimized iteration of the original BERT model, specifically engineered to achieve faster inference speeds and reduced memory consumption. This makes it ideal for use on devices with limited resources or in environments with restricted computational capabilities. The careful distillation process preserves the most informative and relevant features from pre-training on large text corpora. For this work, an increased number of epochs and a learning rate of <= 3e-5 were performed to match the performance and efficiency of other models.

XLM_ROBERTA

The XLM-RoBERTa model is a variant of the RoBERTa model that has been developed by Facebook AI. The utilization of the transformer architecture in XLM-RoBERTa involves the integration of multiple layers of self-attention mechanisms and feed-forward neural networks. Nevertheless, XLM-RoBERTa distinguishes itself through its comprehensive pre-training on extensive multilingual corpora, encompassing a wide array of languages from various regions across the globe. The model undergoes extensive pre-training to acquire diverse and universal representations, enhancing its ability to comprehend and analyze text in diverse linguistic environments. XLM-RoBERTa's "cross-lingual" feature allows it to transfer acquired knowledge and representations from one language to another. XLM-RoBERTa acquires the ability to capture universal linguistic features and patterns shared among languages through training on multilingual data. For this work, the base variation of the model was selected and used for training.

MURIL

The multilingual language model known as MURIL has been developed by Google AI with a specific emphasis on capturing representations that are specifically designed for Indian languages. MURIL stands out from other multilingual models due to its specific emphasis on Indian languages such as Hindi, Bengali, Tamil, Telugu, and other languages spoken in the Indian subcontinent. Significant variations exist among languages in terms of vocabulary, syntax, morphology, and script systems, thereby presenting challenges for conventional multilingual models that may not possess a comprehensive understanding of these nuances. The model undergoes training using a comprehensive corpus of textual data encompassing a range of Indian languages to enhance its capacity to comprehend and analyze text written in Indian languages.

TAMILLION

TaMillion is a version of the BERT model that has been trained solely on a Tamil corpus. As a result, it can comprehend and produce text in the Tamil language alone. This is the second iteration of a Tamil language model that was trained with the help of ELECTRA, a resource provided by Google Research. A total of 482 megabytes of Wikipedia dumps and 11 gigabytes of IndicCorp Tamil have been utilized in the training process of the model as of October 1, 2020. Based on the information provided (Rajalakshmi et al., 2023), this model has been pre-trained on a TPU for a total of 224,000 steps.

DRAVIDIAN REGION MODELS

Dravidian region models are exclusively developed and improved by fine-tuning the pre-existing BERT model using the monolingual corpus of Dravidian languages, specifically Kannada, Malayalam, and Tamil. The work utilized Tamil-Bert, Kannada-Bert, and Malayalam-Bert models from l3cube-pune and Hugging Face (Joshi, 2022) for their respective languages.

Optimizer

AdamW optimizer is used for all the models in this work. An adaptation of the popular Adam optimizer, AdamW optimizer, addresses weight decay's interaction with adaptive learning rates by adding a weight decay mechanism. Weight decay reduces overfitting by adding a component to the optimization objective to discourage large weights. This can cause suboptimal performance due to its interaction with the adaptive learning rate mechanism. AdamW separates weight decay from adaptive learning rate, ensuring it only affects updated parameters. Adam W's weight decay-Adam optimizer integration is unique. Traditional weight decay methods penalize high weights and avoid overfitting with a regularisation term in the optimization objective. In adaptive optimization algorithms like Adam, the weight decay term may disrupt the adaptive learning rate mechanism, resulting in poor performance. AdamW solves this by separating weight decay from adaptive learning rate, ensuring that weight decay only affects updated parameters.

Classifier Models

In this study, downstream classifier models are used to train on embedded data. These models function as binary classifiers, meaning that they take the embedded data as input and classify them as either "Hate" or "Not-Hate." The training data and the testing data are utilized to evaluate the model's performance. Afterward, the results that were obtained are recorded for analysis. The transformer models are utilized as embedded vectors to acquire training and testing data for the classifiers.

SVM

Support Vector Machines (SVM) work by finding the optimal hyperplane that separates data points from different classes in a multidimensional space, maximizing the margin between the closest points, known as support vectors. The goal is to achieve strong performance on unseen data. In this investigation, SVM converts embedded data into numerical feature vectors using techniques like Bag-of-Words or TF-IDF. The Radial Basis Function (RBF) kernel is used to map data into a higher-dimensional space, enabling the model to capture non-linear relationships. SVM then categorizes vectors into 'hate' and 'not hate,' optimizing the hyperplane to maximize separation and minimize classification errors. The model adjusts parameters, including those specific to the kernel, based on labeled data for accurate hate speech detection.

Decision tree

The decision tree is a versatile tool in machine learning, widely used for prediction and classification tasks. Its main objective is to predict values by deriving decision rules from data attributes. The algorithm operates by having instances start at the root node and progress through nodes based on test attributes, moving through branches until they reach a terminal node. At this point, the decision rules

provide a final prediction or classification. The adaptability and structured decision-making of the decision tree make it valuable in various fields, including finance and healthcare.

Random Forest

Random Forest is a powerful ensemble learning technique commonly used for predictive and classification purposes in the field of machine learning. The main goal is to predict the values of instances by combining forecasts from various decision trees. An instance starts at the root node of each decision tree in the forest during operation and moves through the following nodes according to the classification results from various test attributes. The process involves iterating through all decision trees in the forest and determining the final prediction or classification by combining the individual predictions from each tree using voting or averaging methods. The ensemble technique improves the model's accuracy and adaptability abilities, making Random Forest a widely used option for different real-world scenarios.

XGBOOST

XGBoost is a powerful ensemble learning algorithm widely used for predictive modeling in machine learning. Its primary objective is to enhance predictive accuracy by iteratively building a series of weak learners, typically decision trees, in a boosting framework. Each successive tree focuses on correcting the errors of its predecessors, improving the model's performance over time by reducing misclassifications. Through this iterative process, XGBoost strengthens the overall model by combining the output of multiple trees, leveraging their collective strengths for better accuracy and generalization.

CATBOOST

CatBoost is an effective ensemble learning algorithm known for its exceptional performance in predictive modeling tasks in various fields. CatBoost, similar to XGBoost, seeks to improve predictive accuracy by creating an ensemble of weak learners, primarily decision trees, in a boosting framework. However, one notable difference lies in CatBoost' 's intrinsic ability to handle categorical features seamlessly without requiring prior one-hot encoding or label encoding. This advantage makes the preprocessing stage easier and can enhance model performance, particularly in situations with high-dimensional categorical data. Furthermore, the classifier utilizes advanced methods like ordered boosting and oblivious trees to enhance predictions and improve model performance.

IMPLEMENTATION

The dataset employed in the identification of offensive language comprised code-mixed comments in Dravidian languages obtained from the YouTube platform. The HASOC-Offensive Language Identification track in Dravidian Code-Mix FIRE 2021 supplied the dataset. Over 60,000 comments on YouTube were annotated for sentiment analysis and the identification of offensive language. The dataset comprises an estimated four thousand comments in Tamil-English, approximately seven thousand in Kannada-English, and around twenty thousand in Malayalam-English (Chakravarthi et al., 2022).

Table 1. Dataset statistics

Language	Hate	Not hate
Tamil	9283	28618
Malayalam	706	17697
Kannada	1477	4397

Table 2. Dataset distribution

Language	Training	Testing
Tamil	33685	4216
Malayalam	12882	5521
Kannada	4699	1175

The dataset comprised five labels: not offensive, non-specific offensive content, offensive content directed towards a specific identity group, offensive targeting a specific individual or group, offensive targeting others without specifying an identity group or individual. The sixth category, labeled as "Not in intended language," is assigned to comments that are written in a language different from the intended one. This work excluded the sixth label. The merging of the offensive labels has resulted in a singular label, enabling the task to be approached as a binary classification problem. This approach involves differentiating between the categories of Offensive/Hate and Not-Offensive/Not-Hate. The statistical data for comments labeled Hate and Not-Hate is displayed in Table 1. The amount of data categorized as Not-Hate is considerably larger than the amount categorized as Hate. The division of training and testing data in this study was conducted according to the specifications provided in Table 2.

The study employs the Transformer model in the Python programming language to evaluate its efficacy on a dataset containing collaborative tasks. The developers of the machine learning classifier models utilize the sklearn2 library. The models under consideration utilize pre-trained embeddings and are implemented using the SimpleTransformers (2024) framework in Python 3. The Pandas package is used to handle datasets, while the NumPy package is used for array processing needs. The experiments are conducted in the cloud environment of Google Colab Jupyter Notebook, utilizing T4 and V-100 graphics processing units (GPUs) when performing complex computations. Pandas are commonly employed for dataset management, while NumPy is commonly utilized for the execution of array processing procedures.

The training epochs are 20 for Tamil, 17 for Malayalam, and 15 for Kannada models, except for the Malayalam and Kannada DistilBERT models, which used 20 epochs. To implement the early stopping, a patience of two to four epochs is required. To achieve peak performance, it is necessary to use learning rates that are equal to or lower than 2e-5 while simultaneously maintaining previous learning. The "gradient_accumulation_steps":2 training parameter is used to overcome memory limitations that restrict batch size. Batch size training is used to optimize GPU usage, and that training parameter is used to optimize GPU utilization. Utilizing this configuration, gradients from two mini-batches are combined before the model parameters are updated. This results in an increase in the effective batch size without the need for additional memory utilization. This helps to ensure that training is stable and may assist in improving convergence.

```
                precision    recall  f1-score   support

             0       0.95      0.96      0.96      3193
             1       0.87      0.85      0.86      1023

      accuracy                           0.93      4216
     macro avg       0.91      0.91      0.91      4216
  weighted avg       0.93      0.93      0.93      4216

[[3066  127]
 [ 153  870]]
```

Figure 3. CatBoost without Grid Search Cross-Validation for MBERT Tamil

```
Fitting 3 folds for each of 8 candidates, totalling 24 fits
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.97      0.96      3193
           1       0.91      0.85      0.88      1023

    accuracy                           0.94      4216
   macro avg       0.93      0.91      0.92      4216
weighted avg       0.94      0.94      0.94      4216

Confusion Matrix:
[[3109   84]
 [ 149  874]]
Best Parameters: {'depth': 3, 'iterations': 100, 'learning_rate': 0.1}
```

Figure 4. CatBoost with Grid Search Cross-Validation for MBERT Tamil

To further optimize the XGBoost classifier's performance, Grid Search Cross-Validation (GSCV) was employed for hyperparameter tuning. This method systematically explores a grid of hyperparameters, such as the number of estimators, maximum tree depth, and learning rate, to identify the best combination that maximizes the model's F1 score. Threefold cross-validation was used, splitting the training data into three subsets to assess the model's performance across various training-validation splits. The process of training and evaluating different XGBoost models with varying hyperparameters allowed for fine-tuning the classifier, ultimately enhancing its predictive capability. The optimal model, selected based on the highest F1 score, demonstrated improved generalization and accuracy.

The Grid Search Cross-Validation (GSCV) was also implemented in the CATBoost classifier. This is performed to systematically explore the parameter grid and identify the hyperparameters that yield the highest accuracy. By employing Stratified k-Fold Cross-Validation, a robust evaluation was ensured, avoiding potential biases through the maintenance of class balance across folds. Moreover, incorporating early stopping criteria in model training helped prevent overfitting by stopping iterations when performance leveled off on a distinct evaluation set. The systematic approach led to the selection of the most effective model configuration, which improved F1 scores and enhanced predictive abilities, as shown in the classification report and confusion matrix in Figure 3 and Figure 4.

The source code of the implementation is available in Github.

EVALUATION METRICS

In order to determine the efficiency of the models when combined with their following classifiers, it is essential to assess them using evaluation metrics such as precision, recall, and F1 score. Precision is a measure that calculates the proportion of accurate positive predictions out of all the positive predictions made by the model. It provides valuable information about the model's capacity to minimize false positive predictions, as shown in Eqn. (1). Recall, however, measures the ratio of correctly predicted positive instances by the model out of all the actual affirmative instances, indicating the model's capability to identify relevant cases. The formula is shown in Eqn. (2).

$$Precision = \frac{TP}{(TP+FP)} \tag{1}$$

$$Recall = \frac{TP}{(TP+FN)} \tag{2}$$

The F1 score is a mathematical average that combines precision and recall, offering a clear and quantitative evaluation of a model's capacity to accurately classify instances in both positive and negative categories. The formula is shown in Eqn. (3).

$$F1Score = \frac{2*Precision*Recall}{(Precision+Recall)} \tag{3}$$

The dataset presented an imbalance, with a greater proportion of non-offensive content compared to offensive content. Hence, the metrics of precision, recall, and F1-score are computed by employing the weighted average. This approach guarantees that the impact of each class is accurately represented, taking into account their support or the number of samples. The preference for a weighted F1-score over a simple F1-score arises from its ability to account for the uneven distribution of classes. This methodology aligns with comparable approaches observed in related studies, demonstrating its effectiveness in evaluating the performance of models on datasets with imbalances.

RESULTS

Following the completion of preprocessing, the experiment commenced by training all the models using the available data. A 70:30 ratio was used to divide the dataset, with 70% of the data allocated for training and the remaining 30% for testing. The state of 42 was selected based on its perceived optimality. Research has indicated that a learning rate that is equal to or less than 2e-5 produces superior results in comparison to a learning rate that exceeds 2e-5. Tables 3, 4, and 5 display the outcomes of all the models, including the downstream classifier models. Tables 6, 7, and 8 present a summary of the precision, recall, and F1 scores of the proposed models. Additionally, these tables offer a comparison to the previously published findings for the identical dataset. Due to the imbalanced distribution of data in the dataset, weighted metrics were suitable for comparing models. The CatBoost gradient boosting classifier model was chosen as part of the proposed approach due to its better handling of categorical features and its ability to produce accurate results. It can also demonstrate faster training times when dealing with categorical data and requires less hyperparameter tuning, making it more effective with default settings. Furthermore, its use of symmetric trees enables quicker inference, and it performs better on imbalanced datasets. With efficient GPU support, CatBoost excels in these datasets, enhancing both speed and accuracy. To enhance the performance of the CatBoost model, a proposed Grid Search Cross-Validation was implemented. For the results, the CatBoost with Grid Search Cross-Validation was considered.

Figures 5, 6, and 7 were used to compare the results of various models and their classifiers. The graphs demonstrated that mBERT outperformed other models in Malayalam and Tamil in Precision, Recall, and F1-Score. This is because the linguistic characteristics of these languages are closely related and similar to each other, compared to Kannada. Both the mBERT and Muril models for Kannada demonstrated improved performance. However, when comparing the two models, Muril had a slight advantage over mBERT, as it yielded slightly better results in other classifiers. The values highlighted in bold font represent the most optimal results that have been achieved. According to the results of the baseline model evaluation, our proposed method performed better than other options for every language examined in this research.

Table 3. Evaluation metrics for Tamil hate comment detection

Tamil graph				
Models	**Classifiers**	**Precision**	**Recall**	**F1-Score**
XLM-RoBERTa	SVM	0.94	0.94	0.94
	Decision Tree	0.9	0.9	0.9
	Random Forest	0.92	0.92	0.92
	XGBoost with GSCV	0.94	0.94	0.94
	CATBoost with GSCV	0.94	0.94	0.94

Tamil graph				
Models	**Classifiers**	**Precision**	**Recall**	**F1-Score**
Tamil-BERT	SVM	0.94	0.94	0.94
	Decision Tree	0.91	0.91	0.91
	Random Forest	0.93	0.93	0.93
	XGBoost with GSCV	0.94	0.94	0.94
	CATBoost with GSCV	0.94	0.94	0.94
Muril	SVM	0.94	0.94	0.94
	Decision Tree	0.91	0.91	0.91
	Random Forest	0.92	0.93	0.93
	XGBoost with GSCV	0.94	0.94	0.94
	CATBoost with GSCV	0.94	0.94	0.94
mBERT	SVM	0.94	0.94	0.94
	Decision Tree	0.92	0.92	0.92
	Random Forest	0.93	0.93	0.93
	XGBoost with GSCV	0.95	0.95	0.95
	CATBoost with GSCV	**0.94**	**0.94**	**0.94**
DistilBERT	SVM	0.9	0.9	0.9
	Decision Tree	0.85	0.85	0.85
	Random Forest	0.88	0.88	0.88
	XGBoost with GSCV	0.9	0.9	0.9
	CATBoost with GSCV	0.9	0.9	0.9
Tamillion	SVM	0.88	0.89	0.88
	Decision Tree	0.83	0.83	0.83
	Random Forest	0.86	0.86	0.86
	XGBoost with GSCV	0.9	0.9	0.9
	CATBoost with GSCV	0.9	0.9	0.9

Table 4. Evaluation metrics for Malayalam hate comment detection

Malayalam graph				
Models	**Classifiers**	**Precision**	**Recall**	**F1-Score**
XLM-RoBERTa	SVM	0.97	0.98	0.97
	Decision Tree	0.98	0.98	0.98
	Random Forest	0.98	0.98	0.98
	XGBoost with GSCV	0.98	0.98	0.97
	CATBoost with GSCV	0.98	0.98	0.97

Malayalam graph				
Models	**Classifiers**	**Precision**	**Recall**	**F1-Score**
Malayalam-BERT	SVM	0.96	0.96	0.94
	Decision Tree	0.95	0.95	0.95
	Random Forest	0.96	0.96	0.96
	XGBoost with GSCV	0.96	0.96	0.94
	CATBoost with GSCV	0.96	0.96	0.94
Muril	SVM	0.97	0.98	0.97
	Decision Tree	0.98	0.98	0.98
	Random Forest	0.97	0.98	0.97
	XGBoost with GSCV	0.97	0.98	0.97
	CATBoost with GSCV	0.97	0.98	0.97
mBERT	SVM	0.98	0.98	0.98
	Decision Tree	0.98	0.98	0.98
	Random Forest	0.98	0.98	0.98
	XGBoost with GSCV	0.98	0.98	0.97
	CATBoost with GSCV	**0.98**	**0.98**	**0.98**
DistilBERT	SVM	0.97	0.97	0.97
	Decision Tree	0.98	0.97	0.97
	Random Forest	0.97	0.97	0.97
	XGBoost with GSCV	0.97	0.97	0.97
	CATBoost with GSCV	0.97	0.97	0.97

Table 5. Evaluation metrics for Kannada Hate comment detection

Kannada graph				
Models	**Classifiers**	**Precision**	**Recall**	**F1-Score**
XLM-RoBERTa	SVM	0.81	0.82	0.81
	Decision Tree	0.81	0.81	0.81
	Random Forest	0.81	0.81	0.81
	XGBoost with GSCV	0.81	0.82	0.81
	CATBoost with GSCV	0.81	0.81	0.81
Kannada-BERT	SVM	0.81	0.82	0.82
	Decision Tree	0.81	0.8	0.81
	Random Forest	0.81	0.81	0.81
	XGBoost with GSCV	0.81	0.82	0.82
	CATBoost with GSCV	0.81	0.81	0.81

Kannada graph				
Models	**Classifiers**	**Precision**	**Recall**	**F1-Score**
Muril	SVM	0.82	0.82	0.82
	Decision Tree	0.81	0.81	0.81
	Random Forest	0.82	0.82	0.82
	XGBoost with GSCV	0.82	0.82	0.82
	CATBoost with GSCV	**0.82**	**0.83**	**0.82**
mBERT	SVM	0.81	0.82	0.81
	Decision Tree	0.83	0.82	0.82
	Random Forest	0.81	0.82	0.82
	XGBoost with GSCV	0.82	0.82	0.82
	CATBoost with GSCV	0.82	0.82	0.82
DistilBERT	SVM	0.8	0.81	0.8
	Decision Tree	0.81	0.8	0.81
	Random Forest	0.8	0.81	0.8
	XGBoost with GSCV	0.8	0.81	0.8
	CATBoost with GSCV	0.8	0.81	0.81

Table 6. List of rankings based on F1-Score, Precision, and Recall for Tamil

Models	Precision	Recall	F1-Score	Rank
Proposed Work - MBERT + CatBoost with GSCV	**0.94**	**0.94**	**0.94**	**1**
Deep ensemble framework - DNN, XLM-Roberta, Distilbert (Roy et al., 2022)	0.9	0.9	0.9	2
MPNet + CNN (Chakravarthi et al., 2023)	0.97	0.76	0.85	3
Muril + Stemming+ Stopwords Removal (Rajalakshmi et al., 2023)	0.84	0.86	0.84	4
CNN + XLM-Roberta-base + Custom XLM-Roberta (Saha et al., 2021)	0.78	0.78	0.78	5
Ensemble of mBERT, XLM-Roberta, and ULMFiT (Kedia & Nandy, 2021)	0.75	0.79	0.77	6
XLM_Roberta + DPCNN (Zhao & Tao, 2021)	0.75	0.77	0.76	7
mBert and XLM-Roberta (Li, 2021)	0.74	0.77	0.75	8

Table 7. List of rankings based on F1-Score, Precision, and Recall for Malayalam

Models	Precision	Recall	F1-Score	Rank
Proposed Work – MBERT + CatBoost with GSCV	**0.98**	**0.98**	**0.98**	**1**
MPNet + CNN (Chakravarthi et al., 2023)	0.98	0.97	0.98	1
CNN + XLM-Roberta-base + mBert-base (Saha et al., 2021)	0.97	0.97	0.97	2
Ensemble of mBERT, XLM-Roberta, ULMFiT (Kedia & Nandy, 2021)	0.97	0.97	0.97	2
Ensemble mBert and XLM-Roberta (Jayanthi & Gupta, 2021)	0.97	0.97	0.96	3

Models	Precision	Recall	F1-Score	Rank
Customized pre-trained models – Bert, Indicbert, XLM-Roberta (Ghanghor et al., 2021)	0.94	0.95	0.95	4
mBert and XLM-Roberta (Li, 2021)	0.93	0.94	0.94	5
XLM_Roberta + DPCNN (Zhao & Tao, 2021)	0.91	0.94	0.92	6
Deep ensemble framework - DNN, Muril, BERT (Roy et al., 2022)	0.775	0.77	0.769	7

Table 8. List of rankings based on F1-Score, Precision, and Recall for Kannada

Models	Precision	Recall	F1-Score	Rank
Proposed Work – Muril + CatBoost with GSCV	**0.82**	**0.83**	**0.82**	**1**
MPNet + CNN (Chakravarthi et al., 2023)	0.76	0.75	0.76	2
Ensemble mBert and XLM-Roberta (Jayanthi & Gupta, 2021)	0.73	0.78	0.75	3
CNN + mBert-base + Custom XLM-Roberta (Saha et al., 2021)	0.76	0.76	0.74	4
Ensemble of mBERT, XLM-Roberta, and ULMFiT (Kedia & Nandy, 2021)	0.71	0.74	0.72	5
Customized pre-trained models – Bert, Indicbert, XLM-Roberta (Ghanghor et al., 2021)	0.7	0.75	0.72	5
mBert and XLM-Roberta (Li, 2021)	0.7	0.75	0.72	5
XLM_Roberta + DPCNN (Zhao & Tao, 2021)	0.75	0.74	0.69	6

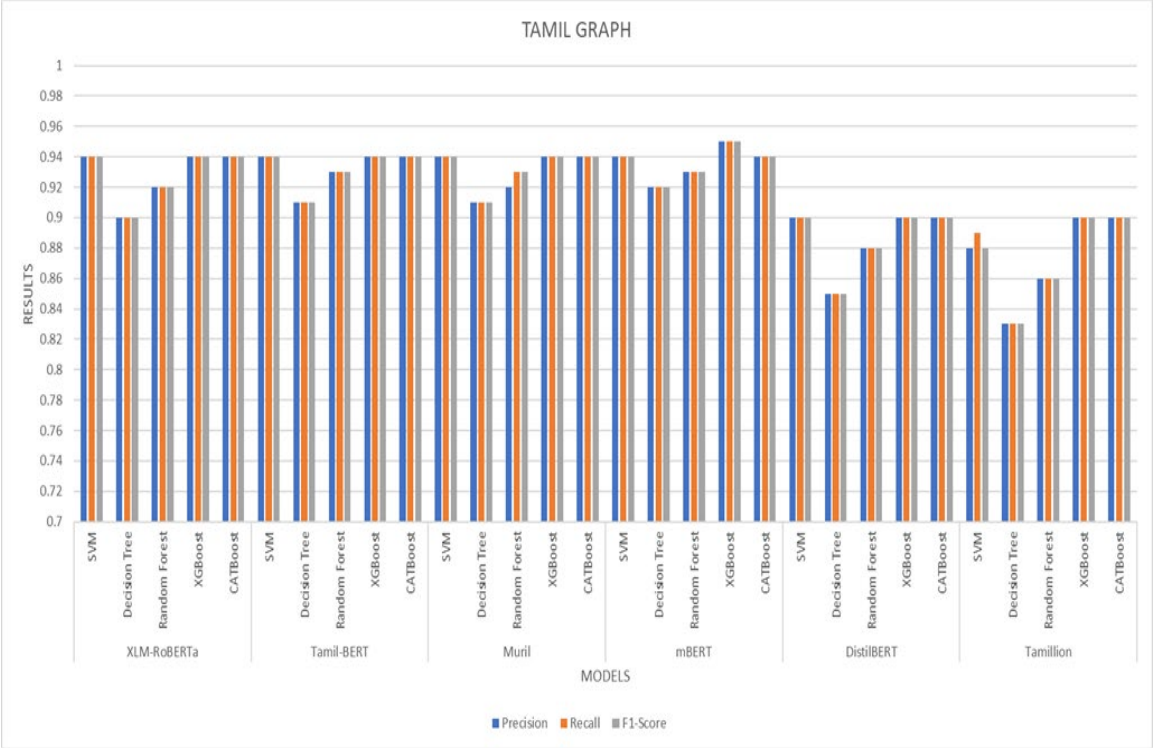

Figure 5. Evaluation of the effectiveness of models employed for the Tamil language

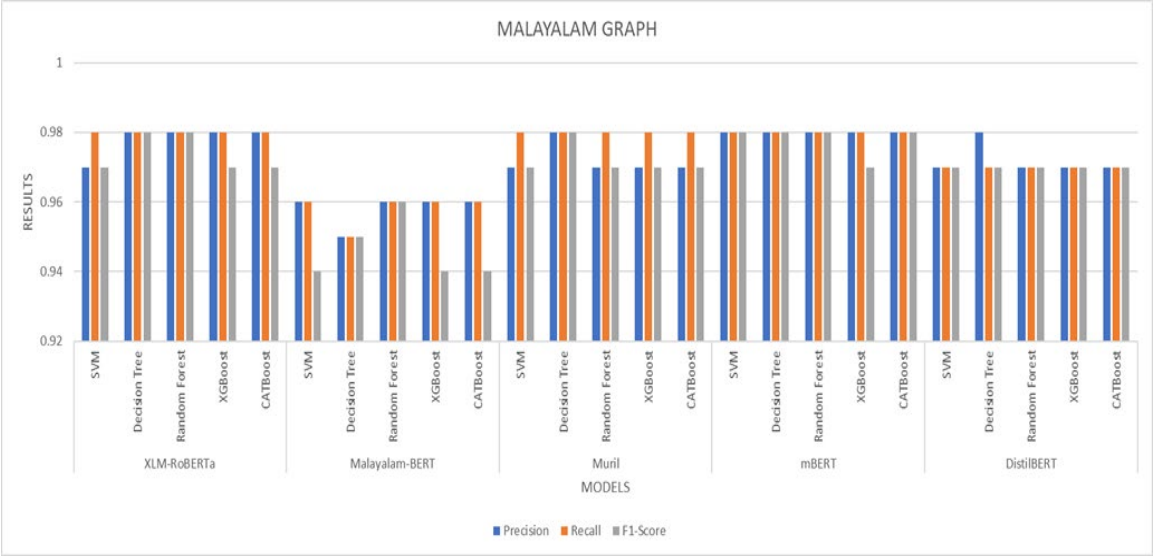

Figure 6. Evaluation of the effectiveness of models employed for the Malayalam language

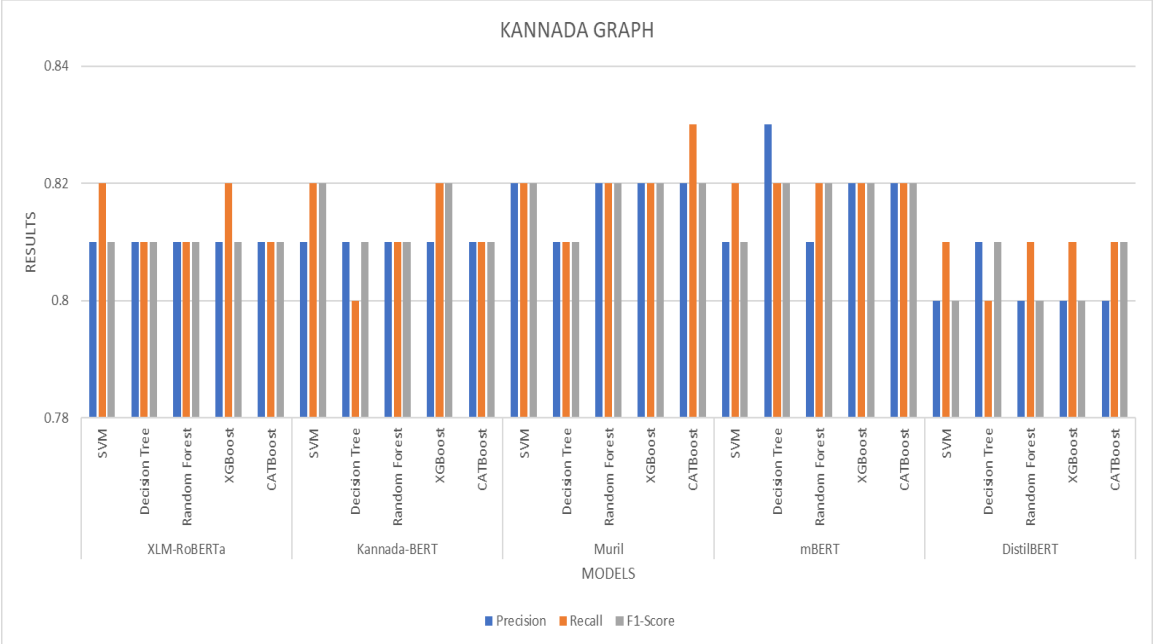

Figure 7. Evaluation of the effectiveness of models employed for the Kannada language

FINDINGS AND DISCUSSION

The evaluation metrics include precision, recall, and *F*1-score, the main factor used for ranking is the *F*1-score. Prior studies achieved the highest F1 scores of 0.85 in Tamil, 0.98 in Malayalam, and 0.76 in Kannada. Research for Malayalam has already achieved the benchmark of 0.98, and this research is able to produce this result. Since both the native and English-coded comments were used for this research, the results were improved in Tamil and Kannada.

Roy et al. (2022) surpassed Tamil by 0.9 in terms of F1-score, but they obtained the lowest score in Malayalam, with a difference of 0.769. Rajalakshmi et al. (2023) achieved the second-highest score in

Tamil, scoring 0.84. The following group (Saha et al., 2021) obtained a score of 0.97 in Malayalam, 0.74 in Kannada, and 0.78 in Tamil. The proposed approach successfully attained the optimal outcome for all three languages across all metrics. Based on the tables, the proposed system exhibits superior outcomes in comparison to the other reported results.

This study helped to prevent or reduce the possibility of the start of cyberbullying and harassment in South Indian languages. The present social media has anti-bullying algorithms in their comment sections. However, it is mostly for English and Hindi languages as the data used to train them were hugely different from the dataset used in this study. This could help them to implement the South Indian language in the present comment sections. Also, since the study used code-mix data, the present English models can be used with the Dravidian models to detect offensive content in the comments, making the detection faster and more accurate.

A key limitation of this study was the absence of recent or new comments in the dataset, particularly affecting the ratio of native to code-mixed comments in Kannada, which was lower compared to Tamil and Malayalam. This lack of updated comments introduced challenges when testing on recent YouTube data, leading to some inaccuracies in the model's performance.

CONCLUSION AND FUTURE WORK

The study has proposed a deep learning-based method for detecting hateful and offensive content in Dravidian text. It is possible to utilize this approach to identify content composed using a blend of native and informal code-mix scripts. The framework being proposed employs straightforward preprocessing techniques, specifically emphasizing the application of stemming to the data to detect offensive content present in the textual data. The models exhibited a notable enhancement in performance after the incorporation of stemming techniques and the removal of stopwords. The tokens are subjected to normalization through the process of stemming, which leads to a more accurate representation of the various tokens comprising the entire text. The morphological analyzer algorithm from the IndicNLP library was used to improve stemming (Kunchukuttan, 2020). The method utilizing a morphological analyzer demonstrates a higher level of significance in terms of performance, which can be attributed to the substantial morphological complexity present in the Dravidian language.

The findings of the experimental study indicate that the MuRIL model demonstrates greater efficacy compared to alternative language models for Tamil, while the mBERT model exhibits superior performance for Malayalam and Kannada. The combination of these text embedding models with CatBoost as the downstream classifier for data training is the most effective, resulting in weighted F1 scores of 0.94, 0.98, and 0.82 for Tamil, Malayalam, and Kannada, respectively. The gradient boosting algorithm known as CatBoost effectively integrates predictions from multiple learners and exhibits remarkable performance and resilience, particularly when dealing with categorical features. In the given task, this classifier demonstrates superior performance compared to other classifiers, rendering it a commendable option for attaining high performance without necessitating supplementary customization.

Utilizing a combination of data stemming, data embedding using MuRIL and MBERT, and CatBoost with Grid Search Cross-Validation for learning can yield optimal results. The primary objective of this research is to investigate the application of stopword removal and stemming techniques in the Dravidian script, with a particular focus on their impact on text representation and classification. The proposed system demonstrates a high level of accuracy in identifying hate content. The achieved F1 scores for Tamil, Malayalam, and Kannada were 0.94, 0.98, and 0.82, respectively. The precision values for Tamil, Malayalam, and Kannada are 0.94, 0.98, and 0.82, while the recall values were 0.94, 0.98, and 0.83, respectively.

A key limitation of this study was the lack of recent comments in the dataset, which reduced its relevance to real-world scenarios. The Kannada dataset, in particular, had a much lower proportion of native-to-code-mixed comments compared to Tamil and Malayalam. This imbalance made it difficult to test the model accurately on new YouTube comments, leading to some errors. Additionally, the smaller amount of data for Kannada contributed to lower F1 scores for this language compared to the others. This data shortage limited the model's ability to generalize well, highlighting the need for more complete datasets in low-resource languages to improve overall performance.

The findings of this study demonstrate a notable enhancement in comparison to current methodologies. The subsequent phase involves further augmentation through the incorporation of more specialized linguistic-based methodologies and the enhancement of stemming techniques. This enhancement is not limited to the three languages aforementioned, but rather extends to other Indian languages, including Telugu and Gujarati.

REFERENCES

Chakravarthi, B. R. (2024). Detection of homophobia and transphobia in YouTube comments. *International Journal of Data Science and Analytics, 18*, 49-68. https://doi.org/10.1007/s41060-023-00400-0

Chakravarthi, B. R., Jagadeeshan, M. B., Palanikumar, V., & Priyadharshini, R. (2023). Offensive language identification in Dravidian languages using MPNet and CNN. *International Journal of Information Management Data Insights, 3*(1), 100151. https://doi.org/10.1016/j.jjimei.2022.100151

Chakravarthi, B. R., Priyadharshini, R., Muralidaran, V., Jose, N., Suryawanshi, S., Sherly, E., & McCrae, J. P. (2022). DravidianCodeMix: Sentiment analysis and offensive language identification dataset for Dravidian languages in code-mixed text. *Language Resources and Evaluation*, *56*, 765-806. https://doi.org/10.1007/s10579-022-09583-7

El-Alami, F. Z., El Alaoui, S. O., & Nahnahi, N. E. (2022). A multilingual offensive language detection method based on transfer learning from transformer fine-tuning model. *Journal of King Saud University – Computer and Information Sciences*, *34*(8), 6048-6056. https://doi.org/10.1016/j.jksuci.2021.07.013

Ghanghor, N., Krishnamurthy, P., Thavareesan, S., Priyadharshini, R., & Chakravarthi, B. R. (2021). IIITK@ DravidianLangTech-EACL2021: Offensive language identification and meme classification in Tamil, Malayalam and Kannada. *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages* (pp. 222-229). Association for Computational Linguistics. https://aclanthology.org/2021.dravidianlangtech-1.30/

Hande, A., Hegde, S. U., & Chakravarthi, B. R. (2022). Multi-task learning in under-resourced Dravidian languages. *Journal of Data, Information and Management*, *4*, 137-165. https://doi.org/10.1007/s42488-022-00070-w

Jayanthi, S. M., & Gupta, A. (2021). Sj_aj@ dravidianlangtech-eacl2021: Task-adaptive pre-training of multilingual BERT models for offensive language identification. *arXiv*. https://doi.org/10.48550/arXiv.2102.01051

Joshi, R. (2022). L3Cube-HindBERT and DevBERT: Pre-trained BERT transformer models for Devanagari based Hindi and Marathi languages. arXiv. https://doi.org/10.48550/arXiv.2211.11418

Kedia, K., & Nandy, A. (2021). indicnlp@kgp at DravidianLangTech-EACL2021: Offensive language identification in Dravidian languages. arXiv. https://doi.org/10.48550/arXiv.2102.07150

Kumar, A., Tyagi, V., & Das, S. (2021, December). Detection of offensive language in social networks using LSTM and BERT model. *Proceedings of the IEEE 6th International Conference on Computing, Communication and Automation, Arad, Romania,* 546-548. https://doi.org/10.1109/ICCCA52192.2021.9666342

Kumaresan, P. K., Ponnusamy, R., Priyadharshini, R., Buitelaar, P., & Chakravarthi, B. R. (2023). Homophobia and transphobia detection for low-resourced languages in social media comments. *Natural Language Processing Journal*, 5, 100041. https://doi.org/10.1016/j.nlp.2023.100041

Kunchukuttan, A. (2020). *The IndicNLP Library.* https://github.com/anoopkunchukuttan/indic_nlp_library/

Lavanya, S. K., & Navaneethakrishnan, S. C. (2022, July). Building Tamil text dataset on LGBTQIA and offensive language detection using multilingual BERT. *Proceedings of the International Conference on Inventive Computation Technologies, Nepal,* 489-496. https://doi.org/10.1109/ICICT54344.2022.9850904

Li, Z. (2021, April). Codewithzichao@DravidianLangTech-EACL2021: Exploring multilingual transformers for offensive language identification on code mixing text. *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages* (pp. 164-168). Association for Computational Linguistics. https://aclanthology.org/2021.dravidianlangtech-1.21/

Lora, S. K., Jahan, I., Hussain, R., Shahriyar, R., & Al Islam, A. A. (2023). A transformer-based generative adversarial learning to detect sarcasm from Bengali text with correct classification of confusing text. *Heliyon*, *9*(12), e22531. https://doi.org/10.1016/j.heliyon.2023.e22531

Malik, M. S. I., Cheema, U., & Ignatov, D. I. (2023). Contextual embeddings based on fine-tuned Urdu-BERT for Urdu threatening content and target identification. *Journal of King Saud University – Computer and Information Sciences*, *35*(7), 101606. https://doi.org/10.1016/j.jksuci.2023.101606

Malik, M. S. I., Nazarova, A., Jamjoom, M. M., & Ignatov, D. I. (2023). Multilingual hope speech detection: A robust framework using transfer learning of fine-tuning RoBERTa model. *Journal of King Saud University – Computer and Information Sciences, 35*(8), 101736. https://doi.org/10.1016/j.jksuci.2023.101736

Mehta, M., Gada, D., Sharma, R., Chavan, K., & Kanani, P. (2022, October). Offense detection using BERT and CNN. *Proceedings of the IEEE 3rd Global Conference for Advancement in Technology, Bangalore, India*, 1-5. IEEE. https://doi.org/10.1109/GCAT55367.2022.9971862

Molero, J. M., Pérez-Martín, J., Rodrigo, A., & A. Peñas, A. (2023). Offensive language detection in Spanish social media: Testing from bag-of-words to transformers models. *IEEE Access*, *11*, 95639-95652. https://doi.org/10.1109/ACCESS.2023.3310244

Mozafari, M., Farahbakhsh, R., & Crespi, N. (2022). Cross-lingual few-shot hate speech and offensive language detection using meta learning. *IEEE Access*, *10*, 14880-14896. https://doi.org/10.1109/ACCESS.2022.3147588

Mridha, M. F., Wadud, M. A. H., Hamid, M. A., Monowar, M. M., Abdullah-Al-Wadud, M., & Alamri, A. (2021). L-Boost: Identifying offensive texts from social media post in Bengali. *IEEE Access*, *9*, 164681-164699. https://doi.org/10.1109/ACCESS.2021.3134154

Özberk, A., & Çiçekli, İ. (2021, September). Offensive language detection in Turkish tweets with BERT models. *Proceedings of the 6th International Conference on Computer science and Engineering, Ankara, Turkey,* 517-521. https://doi.org/10.1109/UBMK52708.2021.9559000

Pillai, A. R., & Arun, B. (2024). A feature fusion and detection approach using deep learning for sentimental analysis and offensive text detection from code-mix Malayalam language. *Biomedical Signal Processing and Control*, *89*, 105763. https://doi.org/10.1016/j.bspc.2023.105763

Rajalakshmi, R., Selvaraj, S., & Vasudevan, P. (2023). Hottest: Hate and offensive content identification in Tamil using transformers and enhanced stemming. *Computer Speech & Language*, *78*, 101464. https://doi.org/10.1016/j.csl.2022.101464

Rehan, M., Malik, M. S. I., & Jamjoom, M. M. (2023). Fine-tuning transformer models using transfer learning for multilingual threatening text identification. *IEEE Access*, *11*, 106503-106515. https://doi.org/10.1109/ACCESS.2023.3320062

Roy, P. K., Bhawal, S., & Subalalitha, C. N. (2022). Hate speech and offensive language detection in Dravidian languages using deep ensemble framework. *Computer Speech & Language*, *75*, 101386. https://doi.org/10.1016/j.csl.2022.101386

Saha, D., Paharia, N., Chakraborty, D., Saha, P., & Mukherjee, A. (2021). Hate-Alert@DravidianLangTech-EACL2021: Ensembling strategies for transformer-based offensive language detection. arXiv. https://doi.org/10.48550/arXiv.2102.10084

Sengupta, A., Bhattacharjee, S. K., Akhtar, M. S., & Chakraborty, T. (2022). Does aggression lead to hate? Detecting and reasoning offensive traits in Hinglish code-mixed texts. *Neurocomputing*, *488*, 598-617. https://doi.org/10.1016/j.neucom.2021.11.053

SimpleTransformers. (2024). https://simpletransformers.ai/

Sreelakshmi, K., Premjith, B., & Soman, K. P. (2020). Detection of hate speech text in Hindi-English code-mixed data. *Procedia Computer Science*, *171*, 737-744. https://doi.org/10.1016/j.procs.2020.04.080

Subramanian, M., Adhithiya, G. J., Gowthamkrishnan, S., & Deepti, R. (2022, March). Detecting offensive Tamil texts using machine learning and multilingual transformer models. *Proceedings of the International Conference on Smart Technologies and Systems for Next Generation Computing, Villupuram, India,* 1-6. https://doi.org/10.1109/ICSTSN53084.2022.9761335

Subramanian, M., Ponnusamy, R., Benhur, S., Shanmugavadivel, K., Ganesan, A., Ravi, D., Shanmugasundaram, G. K., Priyadharshini, R., & Chakravarthi, B. R. (2022). Offensive language detection in Tamil YouTube comments by adapters and cross-domain knowledge transfer. *Computer Speech & Language*, *76*, 101404. https://doi.org/10.1016/j.csl.2022.101404

Sundar, A., Ramakrishnan, A., Balaji, A., & Durairaj, T. (2022). Hope speech detection for Dravidian languages using cross-lingual embeddings with stacked encoder architecture. *SN Computer Science*, *3*, Article 67. https://doi.org/10.1007/s42979-021-00943-8

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention is all you need.* arXiv. https://doi.org/10.48550/arXiv.1706.03762

Wadud, M. A. H., Kabir, M. M., Mridha, M. F., Ali, M. A., Hamid, M. A., & Monowar, M. M. (2022). How can we manage offensive text in social media – A text classification approach using LSTM-BOOST. *International Journal of Information Management Data Insights*, *2*(2), 100095. https://doi.org/10.1016/j.jjimei.2022.100095

Zhao, Y., & Tao, X. (2021). ZYJ123@DravidianLangTech-EACL2021: Offensive language identification based on XLM-RoBERTa with DPCNN. *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages* (pp.216-221). Association for Computational Linguistics. https://aclanthology.org/2021.dravidianlangtech-1.29/

APPENDIX: SUMMARY OF RECENT STUDIES

S.no	Paper	Task	Methods	Performance
1.	Shanmugasundaram et al. (2022). Offensive language detection in Tamil YouTube comments by adapters and cross-domain knowledge transfer.	To develop models for identifying offensive language in Tamil YouTube comments, focusing on code-mixed data.	Machine Learning, Fine Tuned Transformers, Adapter Transformers Performed Cross-domain for offensive detection in speech using transformers.	Highest accuracy in ML: KNN- 81.651%. Transformers: XLM-RoBERTa (Large)- 88.532%
2.	(Malik, Cheema, & Ignatov, 2023). Contextual embeddings based on fine-tuned Urdu-BERT for Urdu-threatening content and target identification.	Evaluate the Urdu-BERT model's effectiveness in detecting threatening content and identifying targets in Urdu tweets.	BERT model for detecting threatening content and identifying targets in Urdu tweets. It utilizes contextual semantic embedding	87.5% accuracy and 87.8% F1-score for threatening content identification and 82.5% accuracy and 83.2% F1-score for target identification task.
3.	Molero et al. (2023). Offensive language detection in Spanish social media: Testing from bag-of-words to transformers models.	To describe different machine learning algorithms based on Bag-of-Words and language models to detect offensive language in Spanish messages.	Automated detection of offensive language in Spanish social media texts using a range of models, from classical machine learning with Bag-of-Words to advanced transformer-based models.	The transformer-based models, RoBERTuito and XLM-RoBERTa, obtain the best performance, with RoBERTuito achieving an F1-Score of 0.9011.

S.no	Paper	Task	Methods	Performance
4.	El-Alami et al. (2022). A multilingual offensive language detection method based on transfer learning from transformer fine-tuning model.	Use the Bidirectional Encoder Representations from Transformers (BERT) model to capture the semantics and contextual information within texts for the Arabic language.	The system consists of preprocessing, text representation using BERT models, and classification into offensive and non-offensive categories.	AraBERT shows an accuracy of 90.79% and an F1-score of 93%.
5.	Özberk and Çiçekli (2021). Offensive language detection in Turkish tweets with Bert models.	To discuss the use of natural language processing and BERT for offensive language detection for the Turkish language.	Trained a BERT-based model on a large dataset of offensive and non-offensive text to classify offensive language accurately.	The results showed that the BERT model outperformed other traditional machine-learning models by approx. 80% accuracy in offensive language detection tasks.
6.	Rajalakshmi et al. (2023). H: Hate and offensive content identification in Tamil using transformers and enhanced stemming.	To identify offensive content in Tamil language YouTube comments using machine learning and deep learning techniques.	Performed analysis for different approaches: Without sampling, With Over Sampling, With Under Sampling	The proposed model achieved an F1 score of 84% and an accuracy of 86% for detecting offensive content in Tamil YouTube comments.
7.	Subramanian, Adhithiya, et al., (2022). Detecting offensive Tamil texts using machine learning and multilingual transformer models.	To explain the use of machine learning and multilingual transformer models to detect offensive Tamil texts.	To address the imbalance in the dataset, techniques using the Synthetic Minority Oversampling Technique (SMOTE) were employed.	The results show that BERT, a multilingual transformer model, performs the best with an accuracy of 82% compared to other models.
8.	Hande et al. (2022). Multi-task learning in under-resourced Dravidian languages.	To develop multi-task learning for sequence classification, addressing sentiment analysis and offensive language detection in code-mixed YouTube comments for Tamil, Malayalam, and Kannada.	The proposed work uses a multi-task learning approach with several pre-trained multilingual transformer models to improve the performance of sequence classification models for sentiment analysis and offensive language identification.	The highest F1-Score for sentiment analysis and offensive language identification was achieved by the mBERT model, which scored (66.8%, 90.5%), (59%, 70%) and (62.1%, 75.3%) respectively, for Kannada, Malayalam and Tamil.
9.	Roy et al. (2022). Hate speech and offensive language detection in Dravidian languages using deep ensemble framework.	The authors propose a deep ensemble framework that combines traditional machine learning, deep learning, and transformer-based models to classify offensive and non-offensive code-mixed text.	DistilBERT, DNN, and XLM-RoBERTa for the Tamil dataset, and DistilBERT, DNN, and m-BERT for the Malayalam dataset.	Weighted F1-scores of 0.802 for Malayalam and 0.933 for Tamil code-mixed datasets.
10.	(Malik, Nazarova, et al., 2023). Multilingual hope speech detection: A robust framework using transfer learning of fine-tuning RoBERTa model.	To develop a framework for Multilingual Hope Speech Detection (MHSD) using transfer learning and fine-tuning of RoBERTa models to detect hope speech in both English and Russian languages.	RoBERTa model compared the performance of the proposed pipeline with baselines. Additionally, the study compared the fine-tuned XLM-RoBERTa model against nine state-of-the-art comparable models	94% accuracy and 80.24% F1-score, outperforming the nine comparable classification models

S.no	Paper	Task	Methods	Performance
11.	Pillai and Arun (2024). A feature fusion and detection approach using deep learning for sentimental analysis and offensive text detection from code-mix Malayalam language.	To discuss the feature fusion and detection approach using deep learning for sentimental analysis and offensive text detection in code-mix Malayalam language.	ALBERT tokenization for word splitting and feature extraction. Feature fusion using Shannon entropy and a Recurrent Neural Network (RNN) model.	The proposed Feature fusion +HAN technique produced accuracy - 0.956, Sensitivity - 0.939, Specificity - 0.978
12.	Mozafari et al. (2022). Cross-lingual few-shot hate speech and offensive language detection using meta-learning.	To focus on cross-lingual few-shot hate speech and offensive language detection using a meta-learning approach.	A meta-learning approach was proposed to tackle the challenge of few-shot hate speech and offensive language detection in low-resource languages.	Meta-learning-based models outperform transfer learning-based models in a majority of cases, and that Proto-MAML is the best-performing model.
13.	Mridha et al. (2021). L-boost: Identifying offensive texts from social media posts in Bengali.	To focus on the development of a detection mechanism, L-Boost, designed to identify offensive texts from social media posts in the Bengali language.	Create and assess the L-Boost model for offensive text identification in Bengali social media posts.	Proposed L-Boost model, which combines modified AdaBoost with BERT transformer, is shown an accuracy of 95.11%.
14.	Wadud et al. (2022). How can we manage offensive text in social media – a text classification approach using LSTM-BOOST.	To develop a text classification algorithm named LSTM-BOOST, which employs ensemble learning with a Long Short-Term Memory (LSTM) model	PCA and LSTM networks to each part of the dataset to obtain the most significant variance and reduce the weighted error of the weak hypothesis of the model.	The fastText approach shows the highest 88% for F3 features with the fastText feature extraction method in the proposed LSTM-BOOST architecture.
15.	Lora et al. (2023). A transformer-based generative adversarial learning to detect sarcasm from Bengali text with correct classification of confusing text.	To propose a transformer-based generative adversarial learning approach for detecting sarcasm from Bengali texts.	Generator and Discriminator. The architecture involves modifying the existing GAN-BERT model to suit sarcasm detection in Bengali.	The Bangla BERT-based Generative Adversarial Model achieves the highest accuracy of 77.1% for the "Ben-Sarc" dataset and 97.2% for the "BanglaSarc" dataset.
16.	Mehta et al. (2022). Offense detection using BERT and CNN.	To discuss the use of BERT and CNN for offense detection in user-generated content and propose a model that combines BERT with CNN layers to improve accuracy.	Combines BERT with CNN layers to improve accuracy.	Achieved accuracy using BERT with CNN compared to BERT without CNN of being about 95%.
17	Kumar et al. (2021). Detection of offensive language in social networks using LSTM and BERT model.	To detect offensive language in social networks using deep learning methods, specifically LSTM and BERT models.	Combination of LSTM and BERT models optimized through hyperparameter tuning.	LSTM+BERT method achieved an accuracy of 93%
18	Sundar et al. (2022). Hope speech detection for Dravidian languages using cross-lingual embeddings with stacked encoder architecture.	To develop a multilingual model for detecting hope speech in Dravidian languages using cross-lingual embeddings and a stacked encoder architecture.	Traditional and deep learning approaches, including TF-IDF vectorization, SVM, Naive Bayes, logistic regression, and Bi-LSTM	The model achieved an F1-score of 0.61 for Tamil and 0.85 for Malayalam.

S.no	Paper	Task	Methods	Performance
19	Lavanya and Navaneethakrishnan (2022). Building Tamil text dataset on LGBTQIA and offensive language detection using multilingual BERT.	To address the lack of awareness and connection with the LGBTQ community in the Tamil language, which is limited in terms of data and research.	Explores the use of the Multilingual BERT (mBERT) model for classifying the tweets into "Offensive" and "Non-Offensive" categories.	The mBERT model scores F-score of 93% for the Non-Offensive class and 70% for the Offensive class. This means that the model was able to accurately classify 93% of the Non-Offensive tweets and 70% of the Offensive tweets.
20	Chakravarthi et al. (2023). Detection of homophobia and transphobia in YouTube comments.	To detect homophobic and transphobic language in YouTube comments using machine learning models.	The study utilized a range of machine learning and deep learning techniques to detect homophobic and transphobic content in YouTube comments.	The approximate accuracy: English - 90% Tamil - 83% Tamil-English – 79%
21	Kumaresan et al. (2023). Homophobia and transphobia detection for low-resourced languages in social media comments.	To detect homophobic and transphobic content in social media comments for low-resource languages.	The research uses a hybrid approach, combining traditional machine learning and transformer-based models, to detect homophobic and transphobic content in social media comments.	The best-performing model is XLM-RoBERTa large, which achieves an F1-score of 0.83 on the Malayalam dataset, 0.77 on the Hindi dataset, and 0.86 on the English dataset.

AUTHORS

Arunachalam. V is an Analyst with a degree in B. Tech Computer Science and Engineering (Specialization in AI and Robotics) from Vellore Institute of Technology, Chennai. His research interests are machine learning, web development, and data analysis. He has a background in SQL and Python.

Maheswari. N is currently a Professor at the School of Computer Science and Engineering, Vellore Institute of Technology (VIT), Chennai. She has authored and co-authored a large number of research articles in journals. Her research interests include artificial intelligence and behavioral analytics.