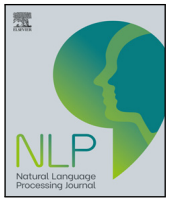
Natural Language Processing Journal 3 (2023) 100006

Contents lists available at ScienceDirect

Natural Language Processing Journal

journal homepage: www.elsevier.com/locate/nlp



Detecting abusive comments at a fine-grained level in a low-resource language☆

Bharathi Raja Chakravarthi [a,*], Ruba Priyadharshini [b], Shubanker Banerjee [c], Manoj Balaji Jagadeeshan [d], Prasanna Kumar Kumaresan [e], Rahul Ponnusamy [f], Sean Benhur [g], John Philip McCrae [e]

[a] School of Computer Science, University Of Galway, Galway, Ireland
[b] Department of Mathematics, Gandhigram Rural Institute-Deemed to be University, Tamil Nadu, India
[c] ADAPT Centre, Data Science Institute, University Of Galway, Galway, Ireland
[d] Birla Institute of Technology and Science, Pilani, Pilani, Rajasthan, India
[e] Insight SFI Research Centre for Data Analytics, Data Science Institute, University Of Galway, Galway, Ireland
[f] Techvantage Analytics, Trivandrum, Kerala, India
[g] PSG College of Arts and Science, Coimbatore, Tamil Nadu, India

ARTICLE INFO

Keywords:
Abuse detection
Low-resourced languages
Machine learning
Deep learning models
Tamil

ABSTRACT

YouTube is a video-sharing and social media platform where users create profiles and share videos for their followers to view, like, and comment on. Abusive comments on videos or replies to other comments may be offensive and detrimental for the mental health of users on the platform. It is observed that often the language used in these comments is informal and does not necessarily adhere to the formal syntactic and lexical structure of the language. Therefore, creating a rule-based system for filtering out abusive comments is challenging. This article introduces four datasets of abusive comments in Tamil and code-mixed Tamil–English extracted from YouTube. Comment-level annotation has been carried out for each dataset by assigning polarities to the comments. We hope these datasets can be used to train effective machine learning-based comment filters for these languages by mitigating the challenges associated with rule-based systems. In order to establish baselines on these datasets, we have carried out experiments with various machine learning classifiers and reported the results using F1-score, precision, and recall. Furthermore, we have employed a t-test to analyze the statistical significance of the results generated by the machine learning classifiers. Furthermore, we have employed SHAP values to analyze and explain the results generated by the machine learning classifiers. The primary contribution of this paper is the construction of a publicly accessible dataset of social media messages annotated with a fine-grained abusive speech in the low-resource Tamil language. Overall, we discovered that MURIL performed well on the binary abusive comment detection task, showing the applicability of multilingual transformers for this work. Nonetheless, a fine-grained annotation for Fine-grained abusive comment detection resulted in a significantly lower number of samples per class, and classical machine learning models outperformed deep learning models, which require extensive training datasets, on this challenge. According to our knowledge, this was the first Tamil-language study on FGACD focused on diverse ethnicities. The methodology for detecting abusive messages described in this work may aid in the creation of comment filters for other under-resourced languages on social media.

1. Introduction

A significant increase in the amount of digital information that is being distributed through various social media platforms has occurred in recent years (Karayiğit et al., 2021). Online social networks (OSNs) have grown in importance over recent years, becoming a go-to source for accessing news, information, and entertainment (İrsoy and Cardie, 2014; Dai and Song, 2019). However, despite the numerous benefits of employing OSNs, a growing body of evidence suggests that there is an ever-increasing number of malevolent actors who are exploiting

☆ This work were supported in part by a research grant from Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289_P2 (Insight_2).
* Corresponding author.
E-mail addresses: bharathi.raja@universityofgalway.ie (B.R. Chakravarthi), rubapriyadharshini.a@gmail.com (R. Priyadharshini), shubhanker.banerjee@adaptcentre.ie (S. Banerjee), prasanna.kumaresan@insight-centre.org (P.K. Kumaresan), rahul.ponnusamy@insight-centre.org (R. Ponnusamy), seanbenhur@gmail.com (S. Benhur), john.mccrae@insight-centre.org (J.P. McCrae).

https://doi.org/10.1016/j.nlp.2023.100006
Received 7 February 2023; Received in revised form 12 April 2023; Accepted 12 April 2023

these networks to spread hate speech and cause harm to other individuals (Ritter et al., 2011; Mencarini, 2018). These individuals thrive on upsetting the established norms of a specific platform and cause emotional distress to other users (Märtens et al., 2015; Mandl et al., 2019; Urbaniak et al., 2022).

OSNs such as YouTube, Facebook, Twitter, and other social media platforms promote interactions between individuals all over the world (Castaño-Pulgarín et al., 2021). Unfortunately, not all of these interactions are favorable in nature. As the vast majority of YouTube videos are accompanied by comment sections, the platform is particularly conducive to toxic conduct. In contrast to other online social networks, there is no "befriending" or permission granting procedure to govern who may comment on a video on YouTube. Users can easily engage in this unpleasant and dangerous behavior, which frequently leads to the initiation of other negative activities, much like a disease spreading (Kavitha et al., 2020). While toxic conduct is a sad trademark of online engagement, some circumstances have the potential to enhance the incidence of this type of interaction.

Manually monitoring social media content is impossible due to the large amount of data generated every minute and the limited time available. Instead, systems for automatic detection of offensive/aggressive/toxic content are used (Davidson et al., 2017; Fortuna and Nunes, 2018). In order to do this, various studies in the field have investigated supervised approaches, which consider hate speech detection as a binary classification issue (abusive/hateful vs. not abusive/not hateful) or as a hierarchical level issue (abusive/hateful vs. not abusive/not hateful) (Beddiar et al., 2021; Fortuna et al., 2021). Several research works on the detection of hate speech and inflammatory language have been undertaken over the years, employing both traditional machine learning (ML) and deep learning techniques (Poletto et al., 2021). The vast majority of them, on the other hand, have focused on identifying certain types of undesirable information and removing them from the internet completely. As a consequence, they focus on the development of models and features for tackling specific problems in this domain Arango et al. (2019).

Despite the fact that there is no globally acknowledged definition of hate speech, the most frequently used definition was used in this study. The phrase "abusive content" refers to any type of communication that is abusive, insulting, intimidating, and/or incites violence or discrimination against an individual or a vulnerable group based on their ethnic origin, gender, sexual orientation, or religious affiliation (Haas, 2012; Burnap and Williams, 2015; Waseem and Hovy, 2016; Pamungkas et al., 2021). Examples of topics include misogyny (García-Díaz et al., 2021), racism, xenophobic sentiments (Keum and Ahn, 2021), homophobia (Barragán-Medero and Pérez-Jorge, 2020), and transphobia. Depending on the issue, hostile material is directed toward specific targets that reflect the community (individuals or groups) that is being targeted. For example, when the subject's focus is racism, both black and white individuals may be potential targets, whereas women may be potential targets when the topical focus is misogyny or sexism (Sap et al., 2019; Zeinert et al., 2021).

Given the growing number of instances of the use of abusive language and online harassment on social media, which are motivated by racism, homophobia, sexist stereotypes, prejudices, and intolerance and can result in incidents of violence, discrimination, and persecution both online and offline, our contribution was devoted to advancing our understanding of online abusive behaviors in the case of low-resourced Tamil language. For the first time, we have presented research on fine-grained abusive language detection in the low-resourced Tamil (ISO 639-3: tam) language that focuses on a more fine-grained understanding of abusive language. We created a corpus with fine-grained annotation of abusive comments in Tamil and Tamil–English code-mixed data. To date, there have been no attempts to tackle the challenge of detecting misogyny in the Tamil language; aside from a few studies conducted by our team on homophobia and offensives, this corpus is the first and only marked-up collection created for such a purpose. We detected abusive comments by classifying YouTube comments using a variety of ML approaches ranging from traditional classifiers to deep learning models such as long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997), convolutional neural networks (CNN) (LeCun et al., 1989), and state-of-the-art language modeling as well as by proposing the adaptive ensemble method.

By conducting tests on abusive comment detection at a fine-grained level, we hoped to address the following research questions (RQs):

RQ1. : Should abusive language detection be addressed as a binary class (abusive/not-abusive) or a fine-grained (misogyny, homophobia, transphobia, xenophobia, counter speech, hope-speech, or not-abusive) problem?
We analyzed the dataset using both conventional abusive and non-abusive classes as well as using the fine-grained approach with seven classes. We found that the former provides better results, but the latter provides more accurate information and will help in the interpretation of the text. Both the annotation approaches have been described in Figs. 2 and 3.

RQ2. : What are the most predictive features to distinguish between fine-grained abusive content detection in low-resource Tamil social media comments? Furthermore, is the use of deep learning methods beneficial for this task?
We experimented with TFIDF, bag of words (BoW), and FastText Tamil with classical machine learning models; in the end, we found that TFIDF and BoW perform better than FastText. Furthermore, we also conducted experiments using deep learning based models to assess their suitability for the task at hand.

In summary, the contributions of our paper are summarized as follows:

1. We have presented a fine-grained comment-based approach to detecting abusive language in YouTube comments in Tamil and code-mixed Tamil–English (low-resource language).
2. We undertook a thorough comparison of ML algorithms with pre-trained word embedding-based deep learning techniques for abusive comment detection in Tamil and code-mixed Tamil–English.
3. We validated the statistical significance of the results generated by the classifiers using paired sample t-test.

The contributions of this paper have been summarized in Fig. 1. The paper's overall structure is followed by explaining the related work in Section 2. The creation of the dataset and ethics followed in creation, along with data statistics, are presented in Section 3. Section 4 and Section 5 describe the methodology to establish baselines on the introduced datasets and experiment settings. In Section 6, the results of different models are compared and discussed with statistical tests. The limitation of the work is discussed in Section 7. Finally, the paper concludes and future direction in Section 8.

2. Related work

Automatic identification of online abusive content is not an easy undertaking, all the more so given the fine line that exists between abusive language and free expression (Khairy et al., 2021). For instance, the usage of curse words may become a concern in abusive content detection, as their presence may result in false positives, such as when they are employed in a non-abusive manner for comedy, emphasis, and catharsis or to express informality since the comments are context dependent. However, when utilized in an abusive setting, they can also serve as a significant indicator of hate speech (Pamungkas et al., 2020a).

Defining the vocabulary of offensive content tends to be just as tough as determining what might be offensive to a certain individual when it comes to phrases with offensive content. In conventional

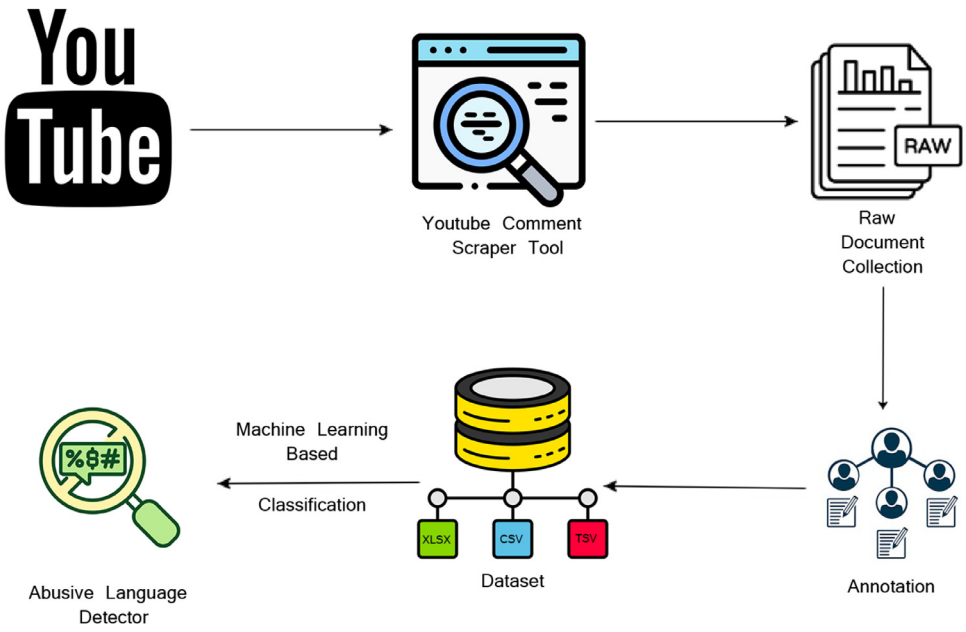


Fig. 1. Gathering Abusive Comments.

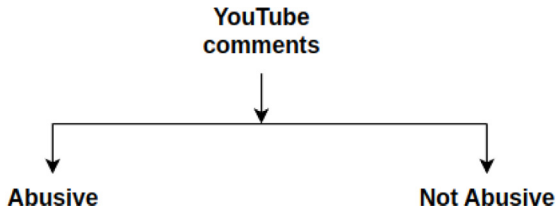


Fig. 2. Comments Hierarchy for BACD.

annotation approaches, a precise definition would be crucial for annotator agreement as well as from the standpoint of accuracy. During the content annotation activities, it is vital for the dataset suppliers that the concepts be perceived in the same way by all participants. When it comes to hate speech, keywords have been most regularly employed to build datasets (Kocoń et al., 2021; Fortuna and Nunes, 2018; Schmidt and Wiegand, 2017). The data collection process for the building of the Offensive Language Identification Dataset (OLID) (Zampieri et al., 2019a), which was utilized in the OffensEval 2019 shared work (Zampieri et al., 2019b), was focused on searching for terms and constructs that are frequently contained in offensive communications as well as analyzing offensive messages. A collection of terms was first utilized to gather tweets, and then some keywords that were not often used in offensive content were eliminated throughout the trail annotation process. In a similar way, the data for the HASOC track dataset (Mandl et al., 2020) was collected through the use of hashtags and keywords that contained offensive material.

For the most part, research works that deal with automatic abusive comment detection rely on supervised techniques to distinguish between abusive and non-abusive content. In the beginning, typical ML algorithms with hard-coded characteristics were used in the initial experiments conducted in this field. A variety of classifiers were employed, including logistic regression (LR) (Ginting et al., 2019), support vector machines (SVM) (Sevani et al., 2021), Naive Bayes (NB) (Fatahillah et al., 2017), decision tree (DT), and random forest (RF) (Nugroho et al., 2019). An extensive set of lexical and syntactic characteristics have been used, including n-grams and the BoW model (TFIDF), along with lexicon-based and TFIDF-based characteristics, stylistic characteristics (e.g., number of characters, punctuation, and text length), and some Twitter-specific characteristics (e.g., the amount of user mentions, hashtags, URLs, information about the user's social networks, and other user characteristics) (Robinson et al., 2018). Automatic hate speech detection has recently been focused on the exploitation of neural models like as LSTM, bidirectional long short-term memory (Bi-LSTM), gated recurrent unit (GRU), and CNN in conjunction with word embedding models such as FastText (Mikolov et al., 2018), word2vec (Mikolov et al., 2013), and ELMo (Peters et al., 2018) in order to find high-frequency words in text (Fortuna and Nunes, 2018; Khairy et al., 2021).

A significant amount of work dealing with abusive language detection has come from teams that have lately collaborated on tasks such as automatic misogyny identification (AMI) (Fersini et al., 2020). SemEval-2019 saw the introduction of HatEval, a shared task designed to identify abusive posts on Twitter directed at two specific groups of people: immigrants and women (Basile et al., 2019). This was done from the standpoint of a multilingual audience (English and Spanish). It

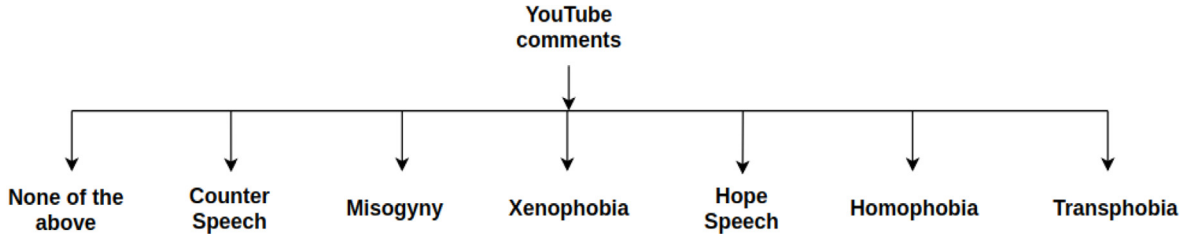


Fig. 3. Comments Hierarchy for FGACD.

was a conventional SVM using a radial basis function (RBF) kernel; the kernel was used in conjunction with Google's Universal Sentence Encoder feature representation (Yang et al., 2020) that produced the top results in the English HatEval competition. In addition to AMI, which was a common responsibility in two distinct evaluation campaigns in 2018 (IberEval and Evalita), hate speech that targets women is another area of emphasis for AMI. In English, classical models for both AMI-IberEval (SVM with numerous handcrafted features) and AMI-Evalita (SVM with several handcrafted features) produced the greatest results (LR coupled with vector representation that concatenates sentence embedding, TFIDF, and average word embedding). Finally, HASOC (Mandl et al., 2019), a shared task at FIRE 2019 that combines hate speech and offensive language identification, encompasses three languages: English, German, and Hindi. All of the aforementioned shared activities included datasets in languages other than English, including Italian, Spanish, Hindi, and German, among other languages. Other languages employed in collaborative projects included Italian (HasSpeeDe (Bosco et al., 2018), which focuses on identifying hate speech toward immigrants) and German (GermEval (Wiegand et al., 2018), which focuses on offensive language identification in German).

HASOC-DravidianCodeMix (Chakravarthi et al., 2020) performed one of the first experiments on offensive language identification in the case of Tamil in 2020, which was followed by a DravidianLangTech shared task (Chakravarthi et al., 2021) where a Tamil dataset of offensive remarks was produced and shared with participants of the shared task. The HASOC-DravidianCodeMix dataset consists of 4000 comments that were gathered from Twitter and the Helo App and combined into one large collection. There are 30,000 comments on YouTube that have been collected for DravidianLangTech's dataset. Several annotators assigned varying ratings of offensiveness to different parts of the dataset. A wide range of systems were evaluated on tasks in the Tamil language using a well-annotated dataset. For Tamil, Malayalam, and English, a researcher (Chakravarthi, 2022a,b) generated a hope speech dataset and undertook a collaborative task to advance the study in a positive manner for each language. Using user-generated comments on YouTube, the author built a Hope Speech dataset for Equality, Diversity, and Inclusion (HopeEDI). It consists of 28,451, 20,198, and 10,705 comments in English, Tamil, and Malayalam, respectively, and it was manually determined whether they contain hope speech or not. This is one of the earliest datasets in Tamil that examines topics related to LGBTQ+ people. These datasets gave researchers the ability to test models using code-mixed Tamil data in order to prevent offensive language from being posted online.

Although studies devoted to the identification of abusive language content in languages other than English have developed in recent years, research on abusive language detection in Tamil is still at an early stage. We are aware of no study on fine-grained abusive language detection in Tamil, except for a few works by our team, all of which dealt with various forms of hate speech to varied degrees in Tamil and other Dravidian languages.

We built on the aforementioned factors in our study and emphasized the need of achieving high-quality fine-grained abusive language detection in Tamil.

- To begin, we defined abusive language content broadly as a negative attitude against a group or a person. The difficulty in identifying abusive speech was resolved by developing fine-grained abusive speech taxonomy in Tamil language texts, including misogyny, homophobia, transphobia, xenophobia, counter speech, and hope speech;
- Our corpus was created by annotating occurrences from YouTube comments in Tamil Nadu, India, Singapore, and Sri Lanka, and it is reflective of real-world data in that we do not intentionally ignore nuanced, imprecise, or intermediate examples;
- We concentrated on fine-grained abusive speech identification, and we identified abusive speech directed at each group mentioned in the text and compared it to a binary technique that classifies the existence of abusive speech;
- We undertook a thorough comparison of ML algorithms with different feature extractors and deep learning models for abusive comment detection in Tamil and code-mixed Tamil–English.

3. Data

We collected comments on YouTube videos,[1] as it is the world's most widely utilized medium for expressing an opinion about a video and it is popular among the Tamil diaspora. We compiled videos from famous YouTubers about the LGBTQ+ community, women rights, and what is racism and how to avoid being racist in the hope that people will be more accepting and we will get more positive commentTables. To verify that our dataset had an appropriate amount of misogynistic, xenophobic, homophobic, and transphobic abusive comments, we began with targeted YouTube videos such as those with the words "gay prank", "transgender prank", "legalizing homosexuality", "China vs. India", "COVID", "gay", "transgender", "women demanding dowry", and "women in science, technology, engineering, and management".

While there were some videos highlighting the good aspects of transgenderism, the majority of videos from news and popular networks portrayed transgender individuals as exploiters and instigators of disputes. Similarly, the comments from videos where women were asking for gifts from their parents for marriage were misogynistic and attacked women. It was difficult to locate a YouTube video discussing LGBTQ+ topics in Tamil since they are still considered taboo topics, same-sex marriage has not been legalized, and homosexuality had been outlawed in India until recently. The comments were collected using the *YouTube Comment Scraper* tool.[2] During annotation of these comments, we observed that a significant amount of the comments were in English or code-mixed Tamil–English. We utilized the *langdetect* library[3] to identify various languages and segregate them throughout the data cleaning process. Furthermore, the data was divided into three categories: Tamil, English, and Tamil–English. We retained the Tamil and Tamil–English code-mixed text for annotation. To comply with data privacy requirements, we removed all user-related information from the corpus. We removed superfluous data such as URLs as a part of the dataset preparation process. Each comment was annotated by at least three separate professionally trained annotators who were given a set of comments to annotate. For fine-grained annotation the annotators classified each comment into the classes defined in Tables 4 and 5. For abusive comment annotation, the comments were classified into the classes defined in Tables 2 and 3. Figs. 4 and 5 illustrate examples of Tamil and Tamil–English code mixed comments with fine-grained annotation.

- Misogynistic material is directed toward women or a gendered group that is closely associated with them. To insult women, misogynistic terms are employed. They include terms that are explicitly insulting and derogatory, sucTableh as "slut" or "whore", as well as terms that reflect negativity or hostility against women, such as "Stacy" or "Becky". (Guest et al., 2021)
- Homophobic material is a type of gender-based harassing statement that involves the use of disparaging labels (e.g., "fag" and "homo") or phrases (e.g., "don't be a homo" and "that's so gay") directed against gay, lesbian, bisexual, queer, or gender-nonconforming people (Poteat and Rivers, 2010).
- Transphobic statements are phrases that are used to demean vulnerable transgender people. They include overtly offensive and pejorative terms, such as "tranny", "trannie", "cross-dresser", or

[1] https://www.youtube.com/
[2] https://github.com/philbot9/youtube-remark-scraper
[3] https://pypi.org/venture/langdetect/

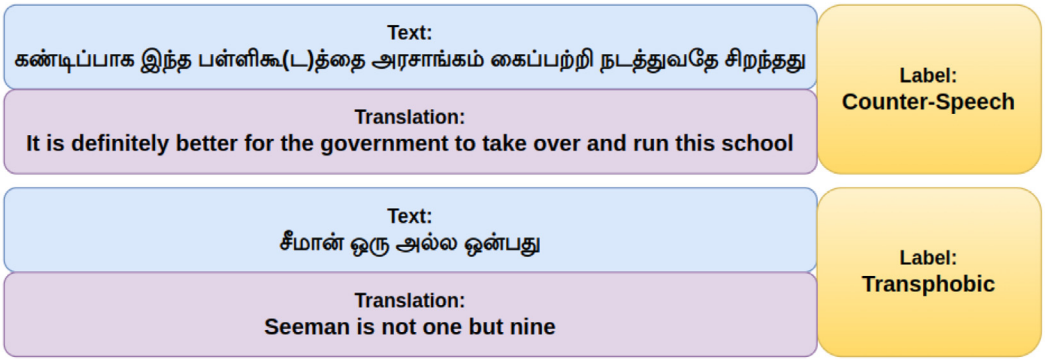


Fig. 4. Sample comments of all the classes in Tamil dataset.

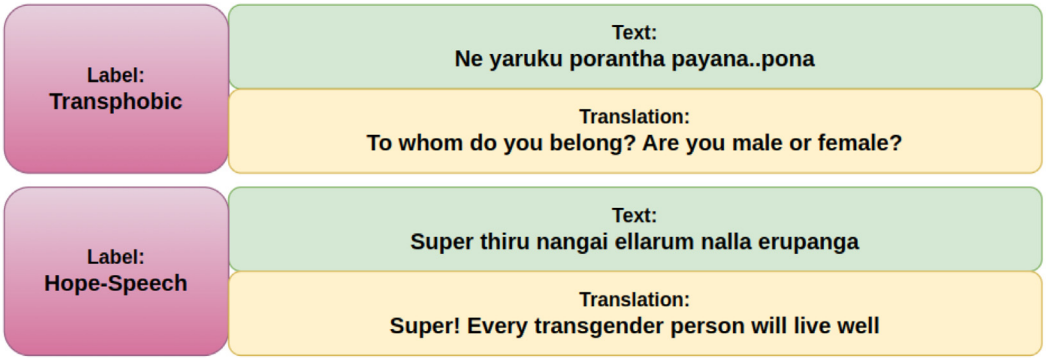


Fig. 5. Sample comments of all the classes in Tamil–English dataset.

"drag", as well as remarks conveying inferred hatred for or wrath toward transgender persons, such as "She-Male", "it", and "9".

- The fear of or hate for what is thought to be alien/foreign or odd is known as xenophobia. It is also known as the dread of the unknown. It includes overtly offensive and pejorative terms, such as "Chinaman", "vadakkan (Northener)", or "Madrasi" as well as remarks conveying inferred hatred of or wrath directed toward different persons.
- Counter speech is content that criticizes, refutes, or brings earlier misogynistic, xenophobic, homophobic or transphobic bigotry in a comment into question. In more specific words, counter speech is a non-aggressive response that provides feedback through fact-based arguments. For example, it may openly denounce earlier mistreatment (Tekiroğlu et al., 2020).
- Hope speech is defined as an upbeat presentation about how people deal with and overcome misfortune. These comments include inspiration from participants' peers and others as well as encouragement, reassurance, advice, and insight. Hope speech is often characterized by empathetic replies (Chakravarthi, 2020).

3.1. Ethics in data annotation

We were committed to upholding ethical standards, which included protecting the privacy and confidentiality of vulnerable individuals. Before forwarding the comments to annotators, we deleted any user IDs and phone numbers. We took great steps to mitigate the risk of individual identification in the data by removing personal information such as names from the dataset, but we did not remove celebrity names. To undertake abusive language identification research, however, we needed to monitor data on sexual orientation, gender, and philosophical ideas. Annotators reviewed only anonymous entries and have pledged to remain anonymous to the original contributor. If annotators were uneasy, they were offered the option of stopping to annotate. Researchers who agree to adhere to ethical standards will be granted access to the material for research purposes (see Fig. 7).

Table 1
Number of comments in the Train, Dev and Test datasets.

Languages	tam	tam-eng
train	2,240	5,948
dev	560	1,488
test	700	1,859
Total	**3,500**	**9,295**

Table 2
Dataset statistics for Tamil binary abusive comment detection.

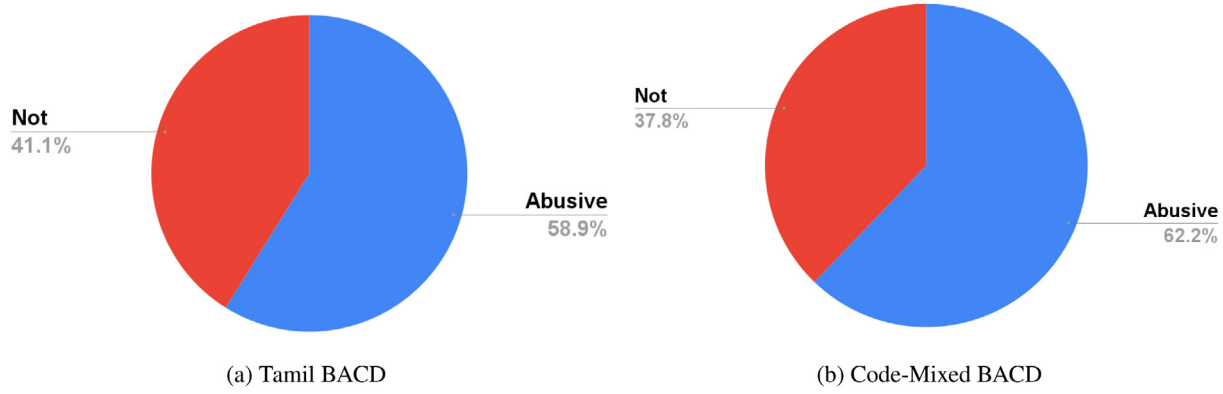
Labels	Train	Dev	Test	Total
Abusive	1,298	346	417	2,061
Not	942	214	283	1,439
Total	**2,240**	**560**	**700**	**3,500**

Table 3
Dataset statistics for code-mixed binary abusive comment detection.

Labels	Train	Dev	Test	Total
Abusive	3,720	919	1,143	5,782
Not	2,228	569	716	3,513
Total	**5,948**	**1,488**	**1,859**	**9,295**

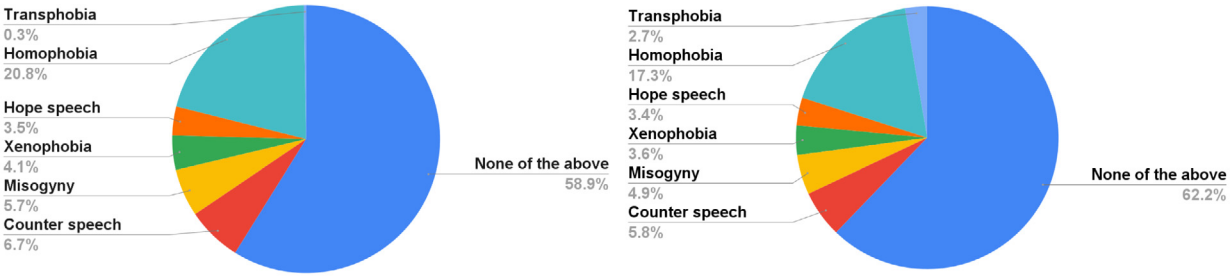
3.2. Data statistics

The preliminary results of a detailed statistical analysis of the data are revealed in this section. Table 1 presents the total number of comments in the Train, Dev, and Test datasets, and it also includes



(a) Tamil BACD (b) Code-Mixed BACD

Fig. 6. Dataset split pie charts for binary data.



(a) Tamil FGACD (b) Code-Mixed FGACD

Fig. 7. Dataset split pie charts for fine-grained data.

Table 4
Dataset statistics for Tamil fine-grained abusive comment detection.

Labels	Train	Dev	Test	Total
None of the above	1,298	346	417	2,061
Counter speech	149	36	48	233
Misogyny	125	29	47	201
Xenophobia	95	24	26	145
Hope speech	86	11	25	122
Homophobia	481	112	135	728
Transphobia	6	2	2	10
Total	**2,240**	**560**	**700**	**3,500**

Table 5
Dataset statistics for code-mixed fine-grained abusive comment detection.

Labels	Train	Dev	Test	Total
None of the above	3,720	919	1,143	5,782
Counter speech	348	95	95	538
Misogyny	297	70	88	455
Xenophobia	213	53	70	336
Hope speech	211	50	58	319
Homophobia	1,002	261	349	1,612
Transphobia	157	40	56	253
Total	**5,948**	**1,488**	**1,859**	**9,295**

the 3500 and 9295 general statements in the Tamil and Tamil–English languages. Tables 2 and 3 explain the dataset statistic for Tamil and code-mixed binary abusive comment detection (BACD) with several labels like abusive and not abusive in the Train, Test, and Dev datasets. The dataset statistics for Tamil and code-mixed fine-grained abusive comment detection (FGACD) as well as the total number of labels in the Train, Test, and Dev datasets are detailed in Tables 4 and 5. The labels none of the above, counter speech, misogyny, xenophobia, hope speech, homophobia, and transphobia are shown in those tables. Abusive comments' dataset was used to detect abusive YouTube comments in regional languages. It comprises several YouTube video comments and performs a comment-level annotation.

3.3. Content analysis

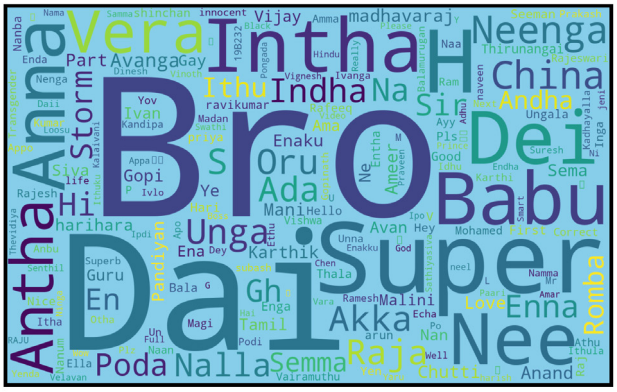
Figs. 8(a) and 8(b) are the wordclouds for the Tamil and Tamil–English datasets. These wordclouds depict the most frequent words used in these datasets. *Bro, Dai, Super, Nee, Intha, Anna, Antha, Babu, Dei* are most used words in the codemixed dataset and *deay, echcha, anand, indha, aiyaa, eanda, ean, naan, theavidiyaa* are transcript words of tamil language that are frequently used words in the Tamil dataset.

In the case of the codemixed dataset, all of the depicted frequent words are not abusive because words like "Bro, Dai, Intha" are beginner works usually used by the user for commenting or replying to other comments. In the case of the Tamil dataset, abusing words such as "echcha" which denotes *a person who does selfish/perfidy/cheating on the persons or friends* and "theavidiyaa" means prostitute, used to abuse others.

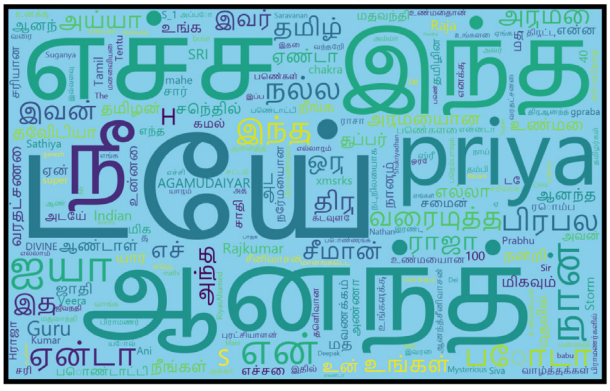
In the comment shown in Fig. 6, "echa" is nothing but "echcha" word is used to abuse a person, and all other words contain stopwords and common words used to frame a sentence. So, the comment may have many non-abusive words, but one single abusive will change the whole comment to an abusive one.

4. Methodology

In order to establish baselines on the introduced datasets we propose the following two tasks:



(a) Word-level visualization for Tamil-English dataset



(b) Word-level visualization for Tamil dataset

Fig. 8. Word-cloud of the proposed dataset.

- Binary abusive comment detection: BACD is a comment-level method in which each comment is evaluated as abusive or non-abusive on the basis of its content as follows: if a text contains abusive speech directed toward at least one of the groups mentioned above, then it is abusive speech (i.e., it receives an "abusive" label); otherwise, it is labeled as non-abusive.
- Fine-grained abusive comment detection: FGACD is a comment-level method in which each comment is evaluated for homophobia, transphobia, misogyny, xenophobia, counter speech, and hope speech.

4.1. Feature selection

We use the following feature selection methods to select features from the texts which in turn are for training the machine learning classifiers:

- **BoW:** All text is regarded as an unordered collection of word properties by the BoW approach, with each word being considered as a distinct feature (Zhang et al., 2010).
- **N-gram** N-gram models are a frequently used technique for selecting features in automated abusive speech detection and related applications (i.e., hateful, aggressive, toxic, racist, sexist) (Mossie and Wang, 2020). N-gram models anticipate the existence of n-1 words prior to the word being processed. We generated n-gram models based on words in this study, with n varied between 2 and 3.
- **TFIDF:** The TFIDF methodology is a frequently used method for calculating the weight of words. The IDF weighting algorithm indicates the importance of a phrase inside a text and prioritizes the most important phrases. The goal of this study was to assess the influence of lexical features on accuracy through the evaluation of various feature selection strategies and TFIDF weighting combinations (Aizawa, 2003).
- **FastText:** FastText is a framework that generates vector representations (embeddings) for character n-grams. Therefore, FastText can be used to generate embeddings for unseen words by leveraging the representations for the constituent character n-grams in the test phase (Joulin et al., 2017).

4.2. Classical machine learning (ML) models

We examined the classification performance of five well-known machine learning-based algorithms NB, SVM, DT, RF, and LR for text classification. This section provides a quick overview of each of the classifiers discussed:

The NB classifier is based on conditional probability and has a low variance but a significant bias. The Bayes theorem is used to construct the NB classifier, which assumes conditionally independent features. The simplicity, robustness, and interpretability of the NB algorithm are its strengths. NB is commonly used in text classification due to its simplicity and processing capacity (Kim et al., 2006; Abooraig et al., 2018).

In addition to categorizing linear and non-linear data, the SVM classifier is a good classifier for extracting abusive statements due to its ability to distinguish between the two types of data (Abooraig et al., 2018; Pamungkas et al., 2020b). SVM generates an n-dimensional hyperplane that divides a dataset into two groups, which is then used to categorize the data using a classification algorithm. The SVM classifier, with its kernel functions, has the advantages of being robust to noise, posing a low risk of overfitting, and being capable of solving challenging classification tasks. It is frequently used in the study of large textual collections. The key disadvantages are the requirement of framing the issue as a two-class classification, the longer training duration, and the fact that it is a black box model, which makes assessment difficult since it is difficult to evaluate (Liu et al., 2010) at fine grained level. We use the SVM algorithm with RBF kernel.

In order to operate from the top down, the DT technique makes use of an if–then conditional expression structure. There are leaf nodes in its structure, which reflect the classes that contain the maximum number of training samples, which were used to discover this node. The simplicity, interpretability, and ability to deal with features in a clear manner are some of the benefits of the DT (Navada et al., 2011; Hmeidi et al., 2015).

A number of alternative DTs are generated by the RF algorithm, and the final judgments are determined after evaluating each of the numerous trees. The ability of RF to deliver great classification results even when working with large volumes of data is a significant advantage. When there are enough trees in the RF method, it is less probable that an over-adaptation problem would develop in classification (Xu et al., 2012; Shah et al., 2020).

When it comes to difficulties associated with the requirement of abusive text categorization, the text classification approach known as LR is often employed. It is more straightforward to interpret the text categories learned from the training set when using LR rather than other classifiers, which is an advantage over other approaches. However, it has a disadvantage in that it uses more data than various classifiers in order to produce more classification results than the other two classifiers (Xu et al., 2012; Shah et al., 2020).

4.3. Deep learning models

This study employed two distinct forms of deep learning architectures: those based on recurrent neural networks (RNNs) and those based on transformers particularly based on the BERT architecture (Devlin et al., 2018). LSTMs which are an improvement on RNNs have been proven to be a great architecture for natural language processing (NLP)

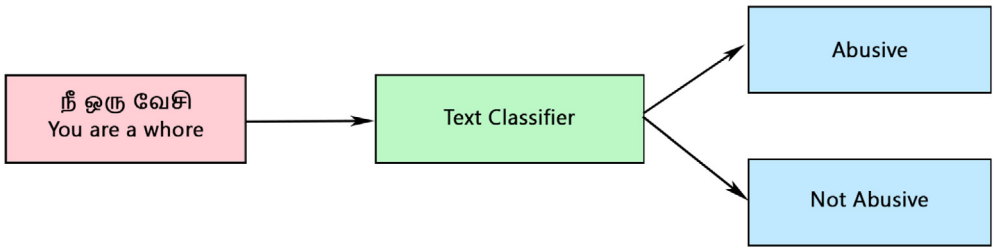


Fig. 9. Binary abusive comment detection task.

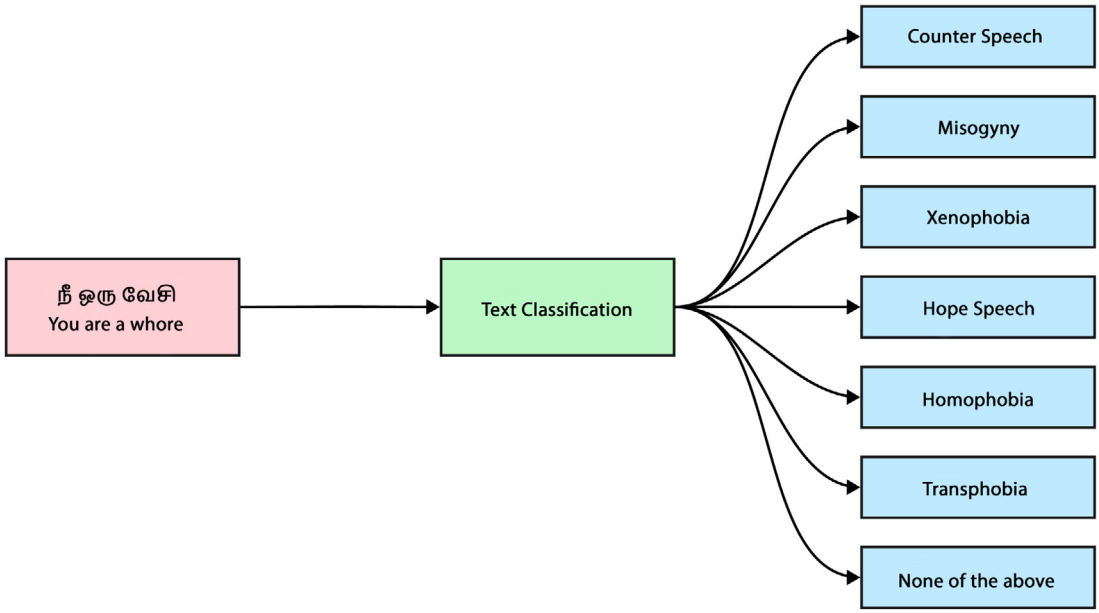


Fig. 10. Fine-grained abusive comment detection task.

Table 6
Hyperparameters for machine learning (ML) models.

Model	Parameters
LR	optimizer='lbfgs', penalty='l2'
SVM	max_iter=2000,kernel='rbf', tol=0.0001
NB	alpha=1.0,class_prior=None,fit_prior=None
RF	criterion='gini', n_estimators=100
DT	min_samples_split=2

tasks. We adopted Bi-LSTM (Hochreiter and Schmidhuber, 1997) and Bi-LSTM with attention (Yang et al., 2019). We encoded the text into TFIDF vectors and fed it to an embedding layer of 300 dimension, followed by two LSTM layers and a final fully connected layer for prediction. In the Bi-LSTM with attention model, the vectors were passed into the attention layer followed by the LSTM layers, which were then passed into a final fully connected layer for predictions. BERT is a transformer-based architecture that has sparked considerable attention in the field of natural language processing due to its exceptional performance on the majority of existing benchmarks. We carried out experiments with BERT based multilingual architectures namely, MURIL (Khanuja et al., 2021), XLM (Conneau and Lample, 2019) and Multilingual BERT (Devlin et al., 2018). (See Tables 13–18.)

5. Settings for the experiment

We used cross-validation with five folds to train all the models, including the ML and deep learning models, for BACD and FGACD. The experiments were carried out using the Python packages scikit-learn (Pedregosa et al., 2011), SciPy (Virtanen et al., 2020), and PyTorch (Paszke et al., 2019), developed by Facebook Artificial Intelligence Research. For training transformer models, we used the transformers (Wolf et al., 2020) package by Huggingface. For machine learning models, we encoded the text to its respective feature extraction methods. We tried three different feature extraction methods such as TFIDF n-gram vectors, the BoW method, and pre-trained FastText vectors. We used Facebook's original library of FastText for implementation.[4] We used the FastText pre-trained Tamil model for Tamil BACD and FGACD and the English pre-trained model for code-mixed BACD and FGACD. Following that, we used these vectors to train our model. Then, we performed classification using ML models such as NB, LR, SVM, RF, and DTs on individual features. The hyperparameters used in the ML models are provided in Table 6 (see Figs. 9 and 10).

The models were trained for ten epochs with a learning rate of 1e-3; we used the Stanza (Qi et al., 2020) Tamil model for tokenizing Tamil texts and Spacy's[5] English model for tokenizing code-mixed texts. For the deep learning models, we used the same feature extraction methods, but this time, we left out FastText since it already gave lower results for ML models. The transformer models were trained for five epochs with 2.9e−5 as the learning rate and linear scheduler with 100 warm-up steps. We used a 0.01 weight decay to avoid overfitting. For BiLSTM and BiLSTM with attention, we trained the models for 10 epochs with

[4] https://github.com/facebookresearch/FastText/
[5] https://github.com/explosion/spaCy

Table 7
Accuracy of ML models on Codemixed BACD and Tamil BACD.

Model	Accuracy-Tamil	Accuracy-Codemixed
NB-TFIDF	0.638 ± 0.003	0.778 ± 0.008
NB-BoW	0.709 ± 0.004	0.787 ± 0.014
NB-FT	0.380 ± 0.029	0.447 ± 0.016
SVM-TFIDF	**0.719 ± 0.009**	0.785 ± 0.016
SVM-BoW	0.710 ± 0.007	0.770 ± 0.010
SVM-FT	0.617 ± 0.001	0.748 ± 0.005
LR-TFIDF	0.688 ± 0.012	0.793 ± 0.014
LR-BoW	0.714 ± 0.013	**0.792 ± 0.011**
LR-FT	0.617 ± 0.001	0.689 ± 0.002
RF-TFIDF	0.702 ± 0.004	0.793 ± 0.017
RF-BoW	0.705 ± 0.007	0.790 ± 0.016
RF-FT	0.626 ± 0.006	0.784 ± 0.012
DT-TFIDF	0.638 ± 0.027	0.745 ± 0.012
DT-BoW	0.663 ± 0.014	0.731 ± 0.010
DT-FT	0.440 ± 0.029	0.668 ± 0.008

a learning rate of 1e-3 and a 0.01 weight decay. Since the dataset was imbalanced, we used focal loss (Lin et al., 2018) for FGACD and cross entropy for BACD .

6. Results and discussion

We computed the weighted precision, recall, and the F1 (F1-ave) and macro-averaged F1 values (F1-macro). F1-macro is computed as an unweighted mean F1 across classes, which treats the classes as balanced and results in a greater penalty for minority class mistakes, especially those associated with a negative attitude. Weighted-F1 is distinguished from F1-macro in that it uses the distribution of classes as weights in calculating the mean, whereas each class is represented proportionately.

The results of the ML models on BACD and FGACD are provided in the Tables 7–12. For Tamil comments in case of the BACD task we observed that SVM and Random Forest classifier significantly outperformed the Bidirectional LSTM and Bidirectional LSTM + attention on the F1 metrics. However, the performance of the MURIL-base model is comparable to that of SVM and Random Forest. As mentioned previously, MURIL is based on the transformer architecture and attention blocks are the fundamental building blocks of the model architecture. In case of code-mixed BACD we observed that SVM is the best performing model among the traditional machine learning models on macro and weighted F1 metrics. However, the MURIL-large model significantly outperforms all the other machine learning and deep learning models on this task. These results indicate that the shared multilingual representation space of MURIL is particularly amenable to code-mixed text. This is not surprising as code-mixed text can be considered to be a specific case of transliterated multilingualism and thus good results achieved by multilingual pre-training can be explained. Similarly, for Tamil BACD task we observed that results achieved by the MURIL-base model are comparable to that of the best performing classical machine learning models (difference of around 1%). Thus, overall the MURIL architecture pretrained multilingually is particularly suitable for this task. It is however pertinent to note that selection of languages during pretraining is important for downstream tasks; to exemplify MURIL which pretrained on Indian languages outperforms MBERT.

Similarly, for the Tamil FGACD task the LSTM models and the Transformer models were outperformed by the Logistic Regression model. For the code-mixed FGACD task Naive Bayes classifier is the best performing model. For the FGACD task we observed a degraded performance across all deep learning models, this can be explained by the large number of parameters in the model and relatively low number of samples per class which in turn hinders the ability of the model to discriminate between different classes. Thus, in general these results indicate that traditional machine learning models outperform deep learning models significantly on this task.

Furthermore, we observed that overall frequency based TF-IDF input representations outperformed BoW as well as FastText. This is a significant finding and indicates that performance gains achieved by using sophisticated embedding methods such as FastText need to be evaluated on a case to case basis. Also, the lower F1 scores observed in the experiments on code-mixed text show that learning effective representations for text classification is more challenging for code-mixed text. This is expected as code mixing entails linguistic variations in text structure which are difficult to capture using pattern recognition and rule based systems. Training the models with more data to accurately represent these linguistic variations is a possible solution to these challenges.

6.1. Statistical test

To find whether the results of the models were merely coincidental due to randomness and if they were statistically significant, we applied a t-test on macro precision, macro recall, and the macro F1 score. Specifically, we used a paired t-test with one tail, type 1, and an alpha of 0.5. It examined the differences between the population means of two sets; assuming each set was in the derivative range of normal distribution, the t-test could be defined as

$$t - test = \frac{\bar{x_1} - \bar{x_2}}{\sqrt{s^2(\frac{1}{n_1} + \frac{1}{n_2})}} \quad (1)$$

Table 8
Results of traditional machine learning models in Tamil BACD. M-Precision, M-Recall, and M-F1-Score denote Macro averaging of all classes of Precision, Recall, and F1-score without considering support values. W-Precision, W-Recall, and W-F1-Score denote the Weighted averaging of all classes of Precision, Recall, and F1-score with considering support values.

Model	Features	M-Precision	M-Recall	M-F1-Score	W-Precision	W-Recall	W-F1-Score
NB	TFIDF	**0.793 ± 0.015**	0.664 ± 0.011	0.681 ± 0.014	**0.785 ± 0.012**	0.777 ± 0.008	0.747 ± 0.010
	BoW	0.754 ± 0.018	**0.732 ± 0.017**	0.741 ± 0.017	0.781 ± 0.015	0.787 ± 0.014	0.782 ± 0.015
	FastText	0.589 ± 0.028	0.567 ± 0.019	0.441 ± 0.016	0.681 ± 0.035	0.447 ± 0.016	0.420 ± 0.018
SVM	TFIDF	0.753 ± 0.020	0.723 ± 0.021	0.352 ± 0.034	0.778 ± 0.017	0.785 ± 0.016	0.778 ± 0.017
	BoW	0.753 ± 0.020	0.714 ± 0.020	**0.785 ± 0.016**	0.764 ± 0.014	0.770 ± 0.012	0.765 ± 0.014
	FastText	0.769 ± 0.007	0.618 ± 0.009	0.612 ± 0.013	0.759 ± 0.005	0.748 ± 0.005	0.699 ± 0.009
LR	TFIDF	0.784 ± 0.017	0.705 ± 0.022	0.724 ± 0.024	0.790 ± 0.015	**0.793 ± 0.014**	0.776 ± 0.018
	BoW	0.766 ± 0.014	0.722 ± 0.015	0.736 ± 0.015	0.784 ± 0.012	0.792 ± 0.011	0.782 ± 0.012
	FastText	0.444 ± 0.200	0.499 ± 0.001	0.408 ± 0.002	0.689 ± 0.002	0.617 ± 0.001	0.563 ± 0.002
RF	TFIDF	0.775 ± 0.028	0.714 ± 0.023	0.731 ± 0.022	0.783 ± 0.024	0.789 ± 0.019	**0.779 ± 0.022**
	BOW	0.414 ± 0.068	0.263 ± 0.014	0.735 ± 0.020	0.616 ± 0.017	0.707 ± 0.005	0.780 ± 0.015
	FastText	0.792 ± 0.029	0.670 ± 0.020	0.695 ± 0.020	0.788 ± 0.021	0.783 ± 0.011	0.757 ± 0.025
DT	TFIDF	0.693 ± 0.009	0.696 ± 0.016	0.694 ± 0.012	0.781 ± 0.015	0.739 ± 0.014	0.741 ± 0.011
	BoW	0.685 ± 0.016	0.691 ± 0.014	0.684 ± 0.017	0.738 ± 0.012	0.732 ± 0.020	0.741 ± 0.014
	FastText	0.615 ± 0.005	0.618 ± 0.012	0.609 ± 0.005	0.616 ± 0.005	0.662 ± 0.011	0.661 ± 0.007

Table 9
Results of traditional machine learning models in code-mixed BACD. M-Precision, M-Recall, and M-F1-Score denote Macro averaging of all classes of Precision, Recall, and F1-score without considering support values. W-Precision, W-Recall, and W-F1-Score denote the Weighted averaging of all classes of Precision, Recall, and F1-score with considering support values.

Model	Features	M-Precision	M-Recall	M-F1-Score	W-Precision	W-Recall	W-F1-Score
NB	TFIDF	0.205 ± 0.004	0.160 ± 0.002	0.196 ± 0.026	0.532 ± 0.005	0.638 ± 0.003	0.426 ± 0.023
	BoW	0.403 ± 0.052	0.275 ± 0.015	0.288 ± 0.027	0.636 ± 0.016	0.709 ± 0.004	0.646 ± 0.006
	FastText	0.212 ± 0.025	0.253 ± 0.031	0.196 ± 0.026	0.556 ± 0.020	0.380 ± 0.020	0.424 ± 0.023
SVM	TFIDF	**0.494 ± 0.057**	**0.322 ± 0.027**	0.352 ± 0.034	**0.664 ± 0.019**	**0.719 ± 0.009**	**0.668 ± 0.013**
	BoW	0.378 ± 0.068	0.319 ± 0.029	**0.354 ± 0.035**	0.657 ± 0.011	0.710 ± 0.007	0.660 ± 0.011
	FastText	0.088 ± 0.000	0.142 ± 0.000	0.109 ± 0.0001	0.380 ± 0.001	0.617 ± 0.001	0.470 ± 0.001
LR	TFIDF	0.322 ± 0.050	0.220 ± 0.012	0.225 ± 0.017	0.592 ± 0.014	0.688 ± 0.012	0.601 ± 0.014
	BoW	0.515 ± 0.081	0.142 ± 0.000	0.343 ± 0.032	0.380 ± 0.001	0.710 ± 0.013	0.656 ± 0.016
	FastText	0.088 ± 0.000	0.143 ± 0.000	0.109 ± 0.000	0.408 ± 0.084	0.617 ± 0.001	0.470 ± 0.001
RF	TFIDF	0.404 ± 0.068	0.268 ± 0.025	0.286 ± 0.017	0.615 ± 0.014	0.707 ± 0.006	0.629 ± 0.009
	BoW	0.414 ± 0.068	0.263 ± 0.014	0.280 ± 0.027	0.616 ± 0.017	0.707 ± 0.005	0.628 ± 0.015
	FastText	0.182 ± 0.054	0.152 ± 0.004	0.125 ± 0.005	0.484 ± 0.049	0.621 ± 0.007	0.499 ± 0.004
DT	TFIDF	0.365 ± 0.002	0.301 ± 0.016	0.330 ± 0.018	0.636 ± 0.016	0.648 ± 0.010	0.626 ± 0.012
	BoW	0.387 ± 0.022	0.321 ± 0.030	0.342 ± 0.031	0.615 ± 0.015	0.660 ± 0.018	0.626 ± 0.016
	FastText	0.171 ± 0.018	0.175 ± 0.025	0.185 ± 0.307	0.175 ± 0.021	0.436 ± 0.033	0.470 ± 0.006

Table 10
Results of traditional machine learning (ML) models in Tamil FGACD. M-Precision, M-Recall, and M-F1-Score denote Macro averaging of all classes of Precision, Recall, and F1-score without considering support values. W-Precision, W-Recall, and W-F1-Score denote the Weighted averaging of all classes of Precision, Recall, and F1-score with considering support values.

Model	Features	M-Precision	M-Recall	M-F1-Score	W-Precision	W-Recall	W-F1-Score
NB	TFIDF	0.191 ± 0.006	0.184 ± 0.006	0.171 ± 0.007	0.521 ± 0.011	0.641 ± 0.009	0.543 ± 0.013
	BoW	0.359 ± 0.050	0.246 ± 0.018	0.255 ± 0.023	0.600 ± 0.026	0.661 ± 0.017	0.605 ± 0.021
	FastText	0.261 ± 0.020	**0.361 ± 0.043**	0.237 ± 0.026	0.564 ± 0.009	0.284 ± 0.013	0.310 ± 0.016
SVM	TFIDF	0.404 ± 0.048	0.293 ± 0.022	0.319 ± 0.012	0.615 ± 0.018	0.655 ± 0.011	0.621 ± 0.012
	BoW	0.378 ± 0.068	0.321 ± 0.039	**0.341 ± 0.048**	0.604 ± 0.023	0.632 ± 0.018	0.614 ± 0.020
	FastText	0.193 ± 0.007	0.168 ± 0.002	0.151 ± 0.003	0.515 ± 0.011	0.619 ± 0.002	0.508 ± 0.004
LR	TFIDF	0.408 ± 0.036	0.241 ± 0.003	0.257 ± 0.004	0.614 ± 0.009	**0.665 ± 0.004**	0.602 ± 0.003
	BoW	0.399 ± 0.063	0.306 ± 0.026	0.332 ± 0.036	**0.625 ± 0.030**	**0.665 ± 0.020**	**0.632 ± 0.021**
	FastText	0.126 ± 0.057	0.143 ± 0.0005	0.106 ± 0.001	0.408 ± 0.084	0.588 ± 0.001	0.436 ± 0.002
RF	TFIDF	**0.448 ± 0.095**	0.226 ± 0.010	0.179 ± 0.003	0.615 ± 0.039	0.665 ± 0.009	0.592 ± 0.011
	BoW	0.446 ± 0.088	0.238 ± 0.018	0.251 ± 0.022	0.630 ± 0.020	0.661 ± 0.015	0.594 ± 0.016
	FastText	0.215 ± 0.062	0.186 ± 0.001	0.179 ± 0.003	0.546 ± 0.034	0.644 ± 0.005	0.546 ± 0.006
DT	TFIDF	0.302 ± 0.043	0.277 ± 0.019	0.304 ± 0.055	0.600 ± 0.026	0.569 ± 0.021	0.559 ± 0.017
	BoW	0.318 ± 0.049	0.290 ± 0.022	0.308 ± 0.053	0.546 ± 0.013	0.574 ± 0.017	0.560 ± 0.012
	FastText	0.206 ± 0.019	0.194 ± 0.002	0.200 ± 0.010	0.211 ± 0.011	0.444 ± 0.022	0.457 ± 0.009

Table 11
Results of traditional machine learning models in code-mixed FGACD. M-Precision, M-Recall, and M-F1-Score denote Macro averaging of all classes of Precision, Recall, and F1-score without considering support values. W-Precision, W-Recall, and W-F1-Score denote the Weighted averaging of all classes of Precision, Recall, and F1-score with considering support values.

Model	Features	M-Precision	M-Recall	M-F1-Score	W-Precision	W-Recall	W-F1-Score
NB	TFIDF	0.243 ± 0.058	0.182 ± 0.002	0.173 ± 0.004	0.566 ± 0.022	0.668 ± 0.002	0.564 ± 0.004
	BoW	0.623 ± 0.092	0.309 ± 0.014	0.336 ± 0.016	0.722 ± 0.006	0.697 ± 0.020	**0.722 ± 0.006**
	FastText	0.212 ± 0.025	0.253 ± 0.031	0.196 ± 0.026	0.556 ± 0.020	0.380 ± 0.029	0.424 ± 0.023
SVM	TFIDF	0.631 ± 0.021	0.425 ± 0.008	**0.481 ± 0.012**	0.722 ± 0.004	**0.743 ± 0.003**	0.718 ± 0.002
	BoW	0.583 ± 0.013	**0.418 ± 0.012**	0.471 ± 0.013	0.714 ± 0.004	0.738 ± 0.003	0.714 ± 0.005
	FastText	0.088 ± 0.000	0.142 ± 0.000	0.109 ± 0.000	0.380 ± 0.001	0.617 ± 0.001	0.470 ± 0.001
LR	TFIDF	0.662 ± 0.079	0.308 ± 0.004	0.349 ± 0.007	0.710 ± 0.017	0.724 ± 0.006	0.669 ± 0.006
	BoW	0.653 ± 0.026	0.392 ± 0.017	0.456 ± 0.020	0.725 ± 0.009	0.742 ± 0.006	0.709 ± 0.009
	FastText	0.088 ± 0.000	0.142 ± 0.000	0.109 ± 0.000	0.386 ± 0.000	0.621 ± 0.000	0.476 ± 0.000
RF	TFIDF	0.703 ± 0.048	0.291 ± 0.007	0.319 ± 0.010	0.728 ± 0.017	0.719 ± 0.002	0.653 ± 0.004
	BoW	**0.790 ± 0.069**	0.288 ± 0.005	0.311 ± 0.010	**0.734 ± 0.006**	0.715 ± 0.004	0.648 ± 0.004
	FastText	0.419 ± 0.051	0.148 ± 0.002	0.164 ± 0.006	0.594 ± 0.013	0.641 ± 0.002	0.527 ± 0.003
DT	TFIDF	0.429 ± 0.016	0.354 ± 0.013	0.371 ± 0.013	0.697 ± 0.020	0.668 ± 0.007	0.65 ± 0.009
	BoW	0.458 ± 0.010	0.362 ± 0.011	0.393 ± 0.006	0.638 ± 0.006	0.686 ± 0.003	0.652 ± 0.107
	FastText	0.194 ± 0.006	0.200 ± 0.010	0.196 ± 0.007	0.192 ± 0.003	0.470 ± 0.005	0.479 ± 0.004

where x_1 is the observed mean of first sample, x_2 is the observed mean of second sample, n_1 and n_2 are the sizes of the first and second samples, respectively, and S is the standard error between the two groups.

We computed the t-test by recording class-wise average metrics (macro precision, weighted precision, macro recall, weighted recall, macro F1 score, and weighted F1 score) for each fold of all the models and then compared each model with the best-performing model. The

Table 12
Accuracy of ML models on Tamil FGACD and Codemixed FGACD.

Model	Features	Accuracy-Codemixed	Accuracy-Tamil
NB	TFIDF	0.668 ± 0.002	0.643 ± 0.009
	BoW	0.670 ± 0.007	0.661 ± 0.017
	FastText	0.382 ± 0.008	0.284 ± 0.013
SVM	TFIDF	**0.738 ± 0.003**	0.655 ± 0.011
	BoW	**0.738 ± 0.003**	0.632 ± 0.020
	FastText	0.622 ± 0.000	0.632 ± 0.020
LR	TFIDF	0.724 ± 0.006	**0.665 ± 0.004**
	BoW	0.743 ± 0.006	**0.665 ± 0.020**
	FastText	0.621 ± 0.000	0.588 ± 0.001
RF	TFIDF	0.718 ± 0.003	0.662 ± 0.108
	BoW	0.717 ± 0.003	**0.665 ± 0.012**
	FastText	0.717 ± 0.003	0.643 ± 0.005
DT	TFIDF	0.671 ± 0.008	0.571 ± 0.020
	BoW	0.671 ± 0.008	0.570 ± 0.012
	FastText	0.465 ± 0.012	0.448 ± 0.021

Table 13
Accuracy of deep learning models on Tamil BACD and Codemixed BACD.

Model	Accuracy-Tamil	Accuracy-Code-mixed
BiLSTM	0.709 ± 0.027	0.723 ± 0.003
BiLSTM + Attention	0.574 ± 0.032	0.615 ± 0.031
MuRIL-BASE	**0.787 ± 0.023**	0.736 ± 0.042
MuRIL-LARGE	0.765 ± 0.065	**0.846 ± 0.013**
XLM-BASE	0.735 ± 0.058	0.729 ± 0.029
XLM-LARGE	0.701 ± 0.001	0.715 ± 0.000
MBERT	0.780 ± 0.046	0.768 ± 0.052

Table 14
Accuracy of deep learning models on Tamil FGACD and Codemixed FGACD.

Model	Accuracy-Tamil	Accuracy-Code-mixed
BiLSTM	0.571 ± 0.026	0.622 ± 0.000
BiLSTM + Attention	0.478 ± 0.026	0.513 ± 0.033
MuRIL-BASE	0.588 ± 0.000	0.649 ± 0.013
MuRIL-LARGE	0.588 ± 0.000	0.631 ± 0.000
XLM-BASE	0.583 ± 0.048	0.506 ± 0.236
XLM-LARGE	0.513 ± 0.151	0.213 ± 0.208
MBERT	0.588 ± 0.000	0.561 ± 0.258

t-test results are provided in Tables 19–22. From the statistical tests, we observed that there was a significant difference between models on code-mixed and Tamil FGACD, but the same could not be said for Tamil and code-mixed BACD since it contains only two categories. This can be explained by the existence of a larger number of classification categories in case of FGACD

6.2. Error analysis

As a part of the model evaluation process, errors were first discovered automatically against ground truth data, prompting additional analysis of the model's flaws. In order to understand and explain the predictions generated by the models built as a part of this study, we performed SHAP analysis on codemixed BACD and Tamil BACD using on the Naive Bayes and Random forest classifiers respectively. SHAP analysis reveals the connections that come together to produce the model's output. The SHAP analysis visualizations are depicted in Fig. 11, Fig. 12, Fig. 13 and Fig. 14 respectively. The SHAP analysis visualizations for BACD CodeMix depicts that deep learning model was able to learn better when compared to the best performing Machine Learning model as it identifies the features i.e. the subtexts that push the prediction toward non-abusive class better. For the text, "ante peddi yedukkure ponnai vacci panam sambatikkuraange", the r Naive Bayes model's prediction is influenced by the subtext "ponnai" which means a girl although in the context of the text it shouldn't be influencing to predict the text as abusive. This is not the case for MURIL, thus conforming to our result that MURIL is a better model for this task. In case of BACD Tamil, the SHAP analysis visualizations also shows a similar results as seen with BACD CodeMix. For the text, " Antha Tamilan oru sathiveri" (transcribed for ease of writing) which roughly translates to "That Tamilian is a casteist", we can see that Random Forrest is influenced by the sub-text "Tami" (transcribed for ease of writing), which refers to one being a person who belongs to the land of Tamil language, toward classifying the text as Abusive whereas in any angle that word does not indicate hatred or abuse whereas the MURIL model doesn't get influenced by it.

We discovered statements that made use of cultural or societal themes to make personal attacks on specific persons (communities) and genders. It was found that sufficient contextual data is available for the model to resolve the meaning of these posts. This was observed in politically charged messages contained in Tamil and code-mixed data. For example, "Adhellam thalaya la vetti iruka matanga, Neenga ovara buildup la vidadheenga ..., Unna madhiriye yevanachum muslim thalaivar vandhu unna madhiri oola viturupan avlodhan..nadadhu irukum..".(English Translation: They won't have cut his head, some Muslim leader like you would have lied) was classified as "none of the above" by all the models. However, the original label was "Xenophobia". In our opinion, these posts lacked contextual information or the worldly understanding necessary to qualify them as abusive content.

Sentences containing profane terms are not always hate-inducing. On the other hand, our model may not have comprehended the ramifications of such profanity when the text contained obscene language but did not instill animosity. For instance, "Saadhi tha perusunu pesura naaingala Ellam seruppa kalati adikanum" (English translation: All the dogs that talk that caste is essential should be beaten up with sandals) was classified as transphobic due to the occurrence of the word "dog", which is offensive to users. But the real-world label for this text is "none of the above". The aforementioned errors occurred even when we used a machine learning models like TF-IDF+SVM, BoW+NB and also for MBERT which a transformer model capable of modeling context. Due to the nature of code-mixing, thess models failed at capturing the relevant context for the aforementioned sample.

Due to the nature of code-mixing and since we extracted the comments from social media, many misspelled words were in the texts across the dataset. Additionally, we trained our pre-trained models on native scripts, and code-mixing was not prevalent in the training data of the models that were used to prepare these, so the models could not learn any contextual information from our datasets. Sarcasm and

Table 15
Results of deep learning models on Tamil BACD. M-Precision, M-Recall, and M-F1-Score denote Macro averaging of all classes of Precision, Recall, and F1-score without considering support values. W-Precision, W-Recall, and W-F1-Score denote the Weighted averaging of all classes of Precision, Recall, and F1-score with considering support values.

Model	M-Prec	M-Recall	M-F1	W-Prec	W-Recall	W-F1
BiLSTM	0.652 ± 0.092	0.571 ± 0.039	0.559 ± 0.058	0.680 ± 0.063	0.709 ± 0.027	0.656 ± 0.041
BiLSTM+Attention	0.502 ± 0.023	0.501 ± 0.022	0.499 ± 0.221	0.573 ± 0.019	0.574 ± 0.032	0.572 ± 0.023
MURIL-BASE	0.764 ± 0.037	**0.778 ± 0.002**	**0.762 ± 0.020**	**0.808 ± 0.016**	**0.787 ± 0.028**	**0.791 ± 0.023**
MURIL-LARGE	0.624 ± 0.232	0.646 ± 0.123	0.620 ± 0.176	0.680 ± 0.169	0.765 ± 0.065	0.708 ± 0.120
XLM BASE	0.519 ± 0.214	0.595 ± 0.011	0.545 ± 0.170	0.604 ± 0.158	0.735 ± 0.058	0.655 ± 0.115
XLM LARGE	0.350 ± 0.000	0.500 ± 0.000	0.412 ± 0.000	0.492 ± 0.001	0.701 ± 0.011	0.579 ± 0.001
MBERT	**0.792 ± 0.034**	0.694 ± 0.098	0.687 ± 0.130	0.795 ± 0.022	0.780 ± 0.046	0.750 ± 0.084

Table 16
Results of deep learning models on code-mixed BACD. M-Precision, M-Recall, and M-F1-Score denote Macro averaging of all classes of Precision, Recall, and F1-score without considering support values. W-Precision, W-Recall, and W-F1-Score denote the Weighted averaging of all classes of Precision, Recall, and F1-score with considering support values.

Model	M-Prec	M-Recall	M-F1	W-Prec	W-Recall	W-F1
BiLSTM	0.784 ± 0.093	0.528 ± 0.012	0.479 ± 0.033	0.760 ± 0.051	0.723 ± 0.008	0.632 ± 0.015
BiLSTM+Attention	0.516 ± 0.024	0.514 ± 0.020	0.512 ± 0.021	0.604 ± 0.018	0.615 ± 0.034	0.607 ± 0.023
MURIL-BASE	0.444 ± 0.174	0.548 ± 0.097	0.485 ± 0.138	0.571 ± 0.121	0.736 ± 0.042	0.639 ± 0.086
MURIL-LARGE	**0.811 ± 0.023**	**0.800 ± 0.005**	**0.805 ± 0.012**	**0.842 ± 0.012**	**0.843 ± 0.013**	**0.842 ± 0.011**
XLM BASE	0.436 ± 0.158	0.538 ± 0.077	0.475 ± 0.117	0.564 ± 0.106	0.729 ± 0.029	0.632 ± 0.071
XLM LARGE	0.357 ± 0.000	0.500 ± 0.000	0.416 ± 0.000	0.511 ± 0.000	0.715 ± 0.000	0.596 ± 0.000
MBERT	0.603 ± 0.203	0.654 ± 0.130	0.621 ± 0.169	0.689 ± 0.147	0.768 ± 0.052	0.720 ± 0.104

Table 17
Results of deep learning models on the Tamil FGACD. M-Precision, M-Recall, and M-F1-Score denote Macro averaging of all classes of Precision, Recall, and F1-score without considering support values. W-Precision, W-Recall, and W-F1-Score denote the Weighted averaging of all classes of Precision, Recall, and F1-score with considering support values.

Model	M-Prec	M-Recall	M-F1	W-Prec	W-Recall	W-F1
BiLSTM	**0.177 ± 0.036**	0.169 ± 0.010	**0.154 ± 0.017**	**0.469 ± 0.019**	0.571 ± 0.026	**0.493 ± 0.019**
BiLSTM + Attention	0.175 ± 0.029	0.154 ± 0.007	0.143 ± 0.011	0.433 ± 0.016	0.478 ± 0.026	0.441 ± 0.014
MuRIL-BASE	0.084 ± 0.000	0.142 ± 0.000	0.105 ± 0.000	0.346 ± 0.000	0.588 ± 0.000	0.435 ± 0.000
MuRIL-LARGE	0.084 ± 0.000	0.142 ± 0.000	0.105 ± 0.000	0.346 ± 0.000	**0.588 ± 0.000**	0.435 ± 0.000
XLM-BASE	0.132 ± 0.039	**0.184 ± 0.034**	0.147 ± 0.035	0.453 ± 0.088	0.583 ± 0.048	0.490 ± 0.054
XLM-LARGE	0.075 ± 0.017	0.145 ± 0.004	0.097 ± 0.017	0.267 ± 0.118	0.513 ± 0.156	0.364 ± 0.144
MBERT	0.084 ± 0.000	0.142 ± 0.000	0.10 ± 0.000	0.346 ± 0.000	0.588 ± 0.000	0.435 ± 0.000

Table 18
Results of deep learning models on code-mixed FGACD. M-Precision, M-Recall, and M-F1-Score denote Macro averaging of all classes of Precision, Recall, and F1-score without considering support values. W-Precision, W-Recall, and W-F1-Score denote the Weighted averaging of all classes of Precision, Recall, and F1-score with considering support values.

Model	M-Prec	M-Recall	M-F1	W-Prec	W-Recall	W-F1
BiLSTM	0.145 ± 0.069	0.143 ± 0.000	0.109 ± 0.000	0.456 ± 0.084	0.622 ± 0.000	0.477 ± 0.000
BiLSTM + Attention	0.172 ± 0.018	0.173 ± 0.007	0.169 ± 0.009	0.470 ± 0.011	0.513 ± 0.035	0.488 ± 0.021
MuRIL-BASE	0.150 ± 0.309	0.206 ± 0.031	0.171 ± 0.031	**0.519 ± 0.065**	**0.649 ± 0.013**	**0.571 ± 0.047**
MuRIL-LARGE	0.090 ± 0.000	0.142 ± 0.000	0.110 ± 0.000	0.398 ± 0.000	0.631 ± 0.000	0.488 ± 0.000
XLM-BASE	0.118 ± 0.069	0.182 ± 0.053	0.129 ± 0.073	0.392 ± 0.210	0.506 ± 0.236	0.428 ± 0.219
XLM-LARGE	0.057 ± 0.022	0.142 ± 0.003	0.050 ± 0.032	0.197 ± 0.142	0.213 ± 0.208	0.130 ± 0.173
MBERT	**0.275 ± 0.171**	**0.270 ± 0.087**	**0.240 ± 0.131**	0.498 ± 0.256	0.561 ± 0.258	0.514 ± 0.259

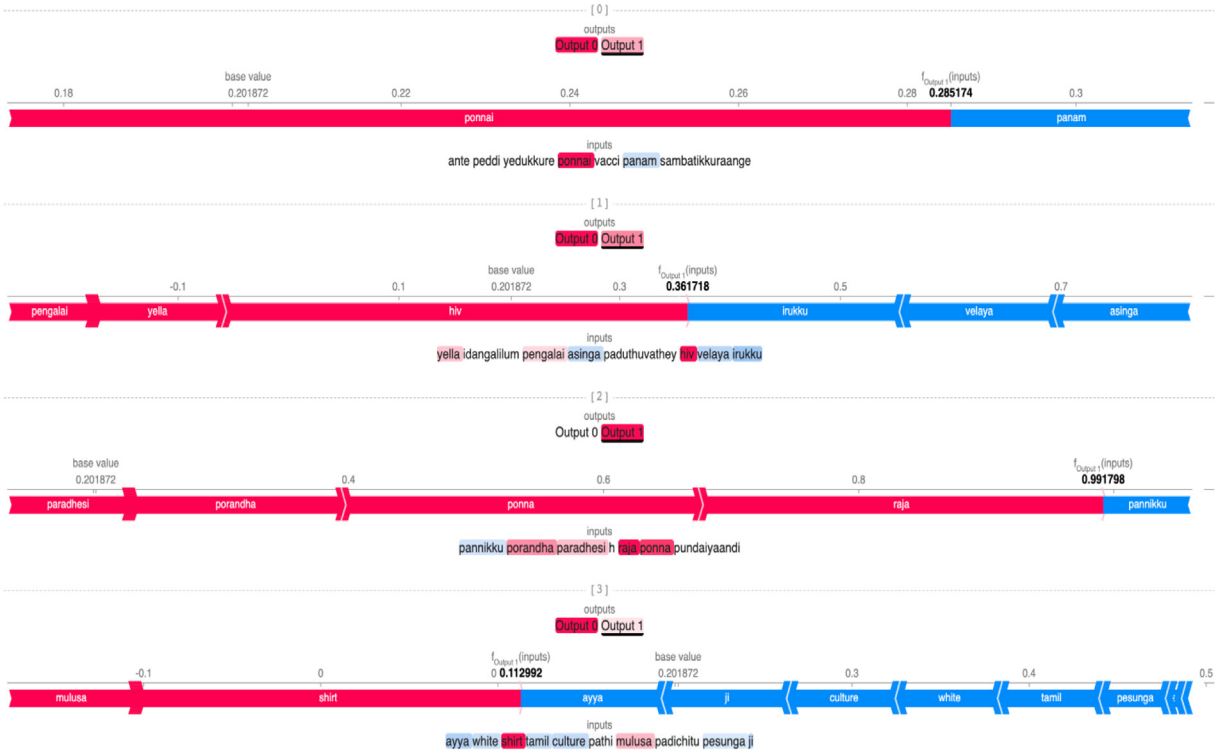


Fig. 11. SHAP analysis visualization about randomly sampled data from test dataset of Code-Mixed BACD, for Naive Bayes classifier.

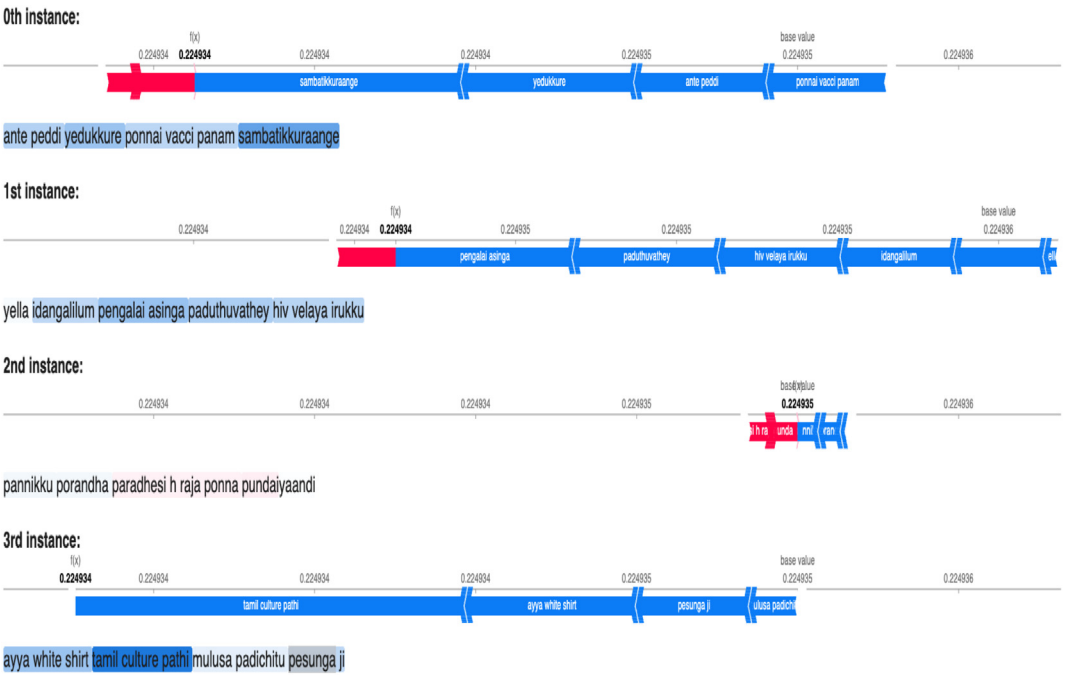


Fig. 12. SHAP analysis visualization about randomly sampled data from test dataset of Code-Mixed BACD, for best deep learning model i.e. MURIL.

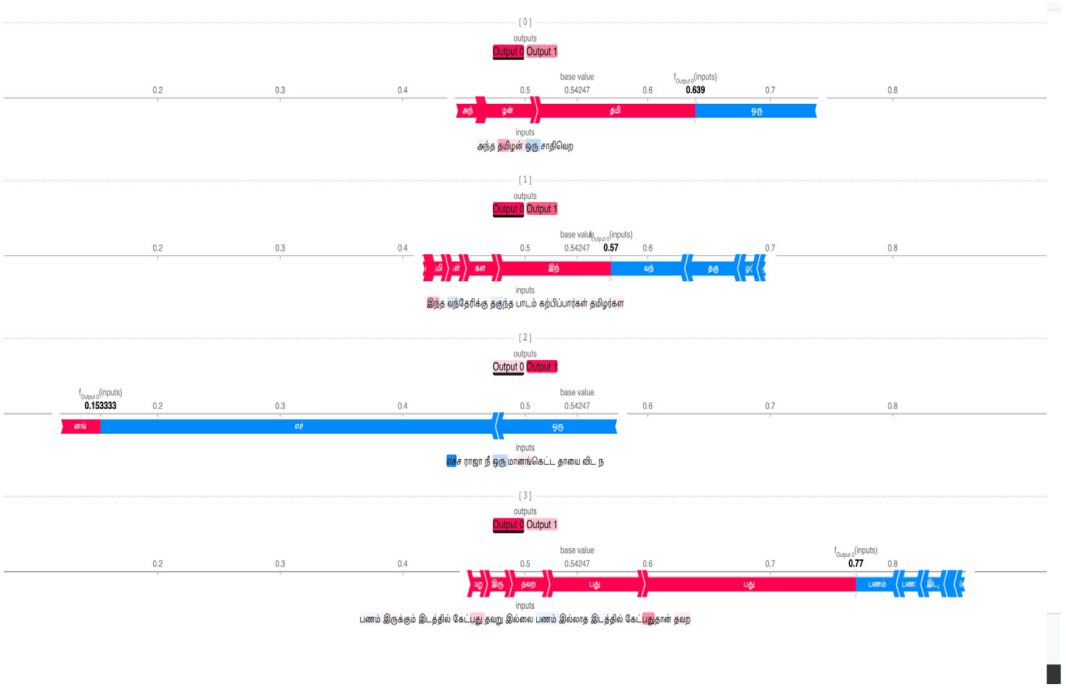


Fig. 13. SHAP analysis visualization about randomly sampled data from test dataset of Tamil BACD, for Random Forest Classifier.

irony are prevalent across many comments in the dataset since sarcasm plays an essential role in abuse detection. Our model was not able to generalize on those domains. To comprehend such types of discourse, classifiers must thoroughly understand the context of the language.

7. Limitation and ethical consideration

Although we have made significant theoretical and empirical improvements, our work has a number of limitations that should be taken into consideration when extending the results to the detection of abusive comments directed toward other groups or individuals and written in other languages. Beginning with the dataset annotation phase, we noticed that inter-annotator agreement was low throughout our inquiry, from the dataset annotation phase to error analysis. That FGACD is still a developing concept with no clearly defined bounds, particularly in instances requiring innovative and uncommon terminology, is most likely to be held responsible. It is important to note that the proposed model currently recognizes fine-grained abusive comments using a simple strategy; however, handling more complicated examples

Fig. 14. SHAP analysis visualization about randomly sampled data from test dataset of Tamil BACD, for best deep learning model i.e. MURIL.

Table 19
The table illustrates the *t*-statistics for Precision, Recall and F1 Score for the models trained on the Tamil BACD task. We used paired t-test at a significance level of 0.05 to ascertain the statistical significance of the predictions generated by the models.

Model	Precision	Recall	F1 Score
RF -TFIDF	9.49E−06	1.45E−07	3.09E−07
RF -BoW	3.80E−07	5.18E−07	5.07E−07
RF - FT	1.32E−05	4.18E−08	2.31E−07
DT -TFIDF	2.70E−06	6.84E−07	1.13E−06
DT -BoW	1.19E−06	5.89E−07	8.65E−07
DT - FT	1.40E−06	5.91E−07	8.55E−07
SVC -TFIDF	9.27E−07	2.81E−06	1.73E−06
SVC -BoW	4.70E−07	2.95E−07	9.72E−08
SVC - FT	2.34E−05	1.98E−07	6.66E−07
LR -TFIDF	1.29E−07	3.13E−06	3.34E−06
LR -BoW	6.50E−07	6.51E−07	2.04E−07
LR - FT	1.28E−04	8.46E−10	1.54E−09
NB -TFIDF	1.91E−05	1.52E−07	4.47E−07
NB -BoW	8.83E−07	7.50E−06	5.87E−06
NB - FT	5.84E−07	1.48E−07	5.51E−07
BILSTM	8.67E−05	1.39E−07	4.14E−04
BiLSTM+Attention	2.79E−06	4.89E−07	1.22E−06
MURIL BASE	4.12E−02	3.82E−02	3.35E−02
XLM R	**4.38E−02**	2.85E−02	**3.71E−02**
XLM R LARGE	1.27E−08	8.46E−10	4.48E−10
MBERT	1.42E−04	**1.94E−01**	5.33E−03

Table 20
The table illustrates the *t*-statistics for Precision, Recall and F1 Score for the models trained on the codemixed BACD task. We used paired t-test at a significance level of 0.05 to ascertain the statistical significance of the predictions generated by the models.

Model	Precision	Recall	F1 Score
RF -TFIDF	3.28E−01	2.83E−01	1.74E−01
RF -BoW	3.06E−01	2.82E−01	1.80E−01
RF - FT	1.99E−01	2.73E−01	2.00E−01
DT -TFIDF	1.99E−02	2.17E−01	1.43E−01
DT -BoW	1.10E−02	2.34E−01	1.54E−01
DT - FT	8.86E−02	1.49E−01	1.23E−01
SVM -TFIDF	1.02E−01	2.50E−01	1.23E−01
SVM -BoW	1.12E−01	2.59E−01	1.52E−01
SVM - FT	1.17E−01	2.76E−01	2.07E−01
NB -TFIDF	**4.10E−01**	2.85E−01	1.93E−01
NB -BoW	1.12E−01	2.60E−01	1.47E−01
NB - FT	1.56E−01	1.34E−01	7.76E−03
LR -TFIDF	2.90E−01	**2.83E−01**	1.77E−01
LR -BoW	1.83E−01	2.66E−01	1.61E−01
LR - FT	1.46E−01	2.76E−01	2.07E−01
BILSTM	2.64E−01	2.72E−01	2.02E−01
BiLSTM+Attention	9.80E−02	1.84E−01	1.44E−01
MBERT	1.18E−01	2.50E−01	1.84E−01
MURIL BASE	5.92E−02	2.57E−01	1.77E−01
MURIL LARGE	1.28E−01	2.72E−01	2.00E−01
XLM R	1.37E−01	2.73E−01	2.03E−01
XLM R LARGE	1.49E−01	2.76E−01	**2.07E−01**

of abusive comments expressed as sarcasm or irony is a different challenge that must be tackled in a real-world environment and is, therefore, outside the scope of this research.

Important ethical considerations must also be accounted for in our job. To begin with, algorithms that automatically detect abusive comments should never be used to discredit the authors. These tools should be used in conjunction with expert judgment rather than in place of expert opinion. Abusive comment detection research, according to the researchers, is currently focused on gaining scientific knowledge on the diverse phenomena of abusive speech rather than automatically punishing writers on social media platforms. Neither our research nor its findings are meant to stigmatize the targets of abusive comments; the inclusion of instances of abusive comments in our articles and datasets does not imply that we concur with the conclusions drawn by hateful writers. Given that we are making our dataset publicly available, we believe the most effective strategies for avoiding the risks of stigmatization are as follows: (1) restricting its use to research purposes only and (2) author anonymization.

8. Conclusion and future work

The main contribution of this paper is the creation of a dataset containing 12,795 social media texts tagged with fine-grained abusive speech in the low-resourced Tamil language, which is available to the academic community. We built the dataset containing comments from YouTube and annotated the corpus at the comment level. We introduce two annotation levels, namely BACD and FGACD. For BACD, each

Table 21
The table illustrates the *t*-statistics for Precision, Recall and F1 Score for the models trained on the Tamil FGACD task. We used paired t-test at a significance level of 0.05 to ascertain the statistical significance of the predictions generated by the models.

Model	Precision	Recall	F1 Score
RF -TFIDF	1.57E−01	1.43E−04	3.26E−04
RF -BoW	1.56E−01	1.85E−04	3.66E−04
RF - FT	1.31E−01	8.00E−05	1.20E−04
DT -TFIDF	1.44E−01	1.61E−04	3.36E−04
DT -BoW	1.48E−01	**9.36E−04**	1.29E−03
DT - FT	1.28E−01	3.07E−04	2.60E−04
SVC -TFIDF	1.57E−01	1.44E−04	3.39E−04
SVC -BoW	1.31E−01	5.28E−04	**1.68E−03**
SVC - FT	1.57E−01	8.00E−05	1.20E−04
LR -TFIDF	**1.64E−01**	1.30E−04	3.37E−04
LR -BoW	1.54E−01	1.59E−04	4.75E−04
LR - FT	1.31E−01	8.00E−05	1.20E−04
NB -TFIDF	1.38E−01	1.05E−04	1.78E−04
NB -BoW	1.49E−01	1.43E−04	2.51E−04
NB - FT	1.31E−01	8.00E−05	1.23E−04
BILSTM	1.37E−01	3.98E−05	2.74E−04
BiLSTM+Attention	1.37E−01	1.12E−04	2.01E−04
MURIL BASE	1.32E−01	4.94E−05	5.04E−05
MURIL LARGE	1.31E−01	8.00E−05	1.20E−04
XLM R	1.35E−01	1.56E−04	1.77E−04
XLM R LARGE	1.30E−01	8.86E−05	1.13E−04
MBERT	1.31E−01	8.00E−05	1.20E−04

Table 22
The table illustrates the *t*-statistics for Precision, Recall and F1 Score for the models trained on the codemixed FGACD task. We used paired t-test at a significance level of 0.05 to ascertain the statistical significance of the predictions generated by the models.

Model	Precision	Recall	F1 Score
RF -TFIDF	8.80E−02	2.13E−02	1.41E−02
RF -BoW	**3.93E−01**	2.04E−02	1.32E−02
RF - FT	1.16E−02	4.14E−03	2.25E−03
DT -TFIDF	2.01E−03	1.60E−02	9.97E−03
DT -BoW	2.70E−03	1.81E−02	1.18E−02
DT - FT	9.40E−05	2.48E−03	1.18E−03
SVC -TFIDF	1.79E−02	3.45E−02	**2.66E−02**
SVC -BoW	9.67E−03	3.17E−02	2.35E−02
SVC - FT	3.59E−03	9.33E−03	1.94E−03
NB -TFIDF	1.99E−02	1.17E−02	3.57E−03
NB -BoW	2.13E−01	2.17E−02	1.12E−02
NB - FT	3.59E−03	9.33E−03	1.94E−03
LR -TFIDF	2.75E−03	1.88E−02	1.21E−02
LR -BoW	1.90E−02	2.86E−02	1.90E−02
LR - FT	5.23E−05	9.33E−03	1.94E−03
BILSTM	1.30E−04	4.53E−03	1.94E−03
BiLSTM+Attention	1.52E−04	3.64E−03	1.73E−03
MBERT	5.90E−04	6.94E−03	2.22E−03
MURIL BASE	2.34E−04	1.08E−02	3.78E−03
MURIL LARGE	8.93E−05	**4.82E−02**	9.18E−04
XLM R	3.57E−06	3.17E−03	8.30E−04
XLM R LARGE	8.18E−07	3.56E−04	1.63E−05

comment was annotated as binary and none of the above. For FGACD we used a broad definition of abusive speech to encompass all expressions of hostility toward a person or group. The annotation divided attitudes toward groups into seven categories: misogyny, homophobia, transphobia, xenophobia, counter speech, hope speech, and none of the above. We conducted tests on abusive comment detection using comment-level BACD and FGACD techniques combined with machine learning and deep learning models. Simple BoW, FastText, and TFIDF trained on our dataset embeddings were used to represent text. NB, LR, SVM, DT, and RF were used as traditional ML models. The LSTM models and pretrained Transformer deep learning models were also used to conduct experiments. Overall on the BACD task we observed that MURIL achieved good performance, indicating the suitability of the multilingual transformers on this task. However, fine grained annotation for FGACD led to relatively lower number of samples per class and the deep learning models which need large datasets for training were outperformed by classical machine learning models on this task. To the best of our knowledge, this was the first study of FGACD directed against multiple minorities in the Tamil language. Our findings supported the following assertions: Abusive comments directed at multiple vulnerable groups should be addressed using a case-based fine-grained approach that incorporates misogynistic, homophobic, transphobic, and xenophobic views as well as counter speech and hope speech, although we achieved better results with binary annotation yet we found the fine-grained annotation to be more representative of the underlying nature of the text(RQ1);

We have experimented with different feature extraction methods along with the machine learning models and found that TFIDF and BoW achieve better results than other feature extractors. Furthermore, for BACD we found the transformer models to be particularly useful and they outperformed all other models for code-mixed text. (RQ2)

As a future work, we intend to improve our dataset and incorporating the following phenomena:

1. Meaningful misspellings and unconventional obscene forms in code-mixed Tamil–English: The text normalization process for code-mixed data has been investigated utilizing a wide variety of supervised and unsupervised approaches. To improve our results, we will look at different techniques to deal with misspellings and other forms of error, which will cause many out of vocabulary problems.
2. Sarcasm: There are a number of abusive comments that are delivered in a sarcastic tone. We will investigate the role that is performed by innovative language techniques particularly sarcasm in abusive speech detection within the context of multilingual settings.

Acronyms

- BACD - Binary Abusive Comment Detection
- BERT - Bidirectional Encoder Representation of Transformers
- BiLSTM - Bidirectional Long Short-Term Memory
- BOW - Bag Of Words
- CNN - Convolutional Neural Networks
- DT - Decision Tree classifier
- FGACD - Fine-Grained Abusive Comment Detection
- LR - Logistic Regression
- MBERT - Multilingual Bidirectional Encoder Representation of Transformers
- ML - Machine Learning
- MURIL - Multilingual Representations for Indian Languages
- NB - Naive Bayes classifier
- OSN - Online Social Networks
- RF - Random Forest classifier
- RQ - Research Question
- SHAP - SHapley Additive exPlanations
- SVM - Support Vector Machine classifier
- TFIDF - Term Frequency-Inverse Document Frequency
- XLM - Cross-lingual Language Model

CRediT authorship contribution statement

Bharathi Raja Chakravarthi: Conceptualization of this study, Data curation, Methodology, Software, Writing – original draft, Writing – review & editing. **Ruba Priyadharshini:** Conceptualization, Data curation, Methodology, Writing – original draft, Writing – review & editing. **Shubanker Banerjee:** Software, Writing – original draft, Writing – review & editing. **Manoj Balaji Jagadeeshan:** Software. **Prasanna Kumar Kumaresan:** Data curation, Writing – review & editing. **Rahul Ponnusamy:** Data curation, Writing-review & editing. **Sean Benhur:** Software. **John Philip McCrae:** Writing – review.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abooraig, R., Al-Zu'bi, S., Kanan, T., Hawashin, B., Al Ayoub, M., Hmeidi, I., 2018. Automatic categorization of Arabic articles based on their political orientation. Digit. Investig. 25, 24–41.

Aizawa, A., 2003. An information-theoretic perspective of TF-IDF measures. Inf. Process. Manage. 39 (1), 45–65.

Arango, A., Pérez, J., Poblete, B., 2019. Hate speech detection is not as easy as you may think: A closer look at model validation. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 45–54.

Barragán-Medero, F., Pérez-Jorge, D., 2020. Combating homophobia, lesbophobia, biphobia and transphobia: A liberating and subversive educational alternative for desires. Heliyon 6 (10), e05225. http://dx.doi.org/10.1016/j.heliyon.2020.e05225, URL: https://www.sciencedirect.com/science/article/pii/S2405844020320685.

Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Rangel Pardo, F.M., Rosso, P., Sanguinetti, M., 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In: Proceedings of the 13th International Workshop on Semantic Evaluation. Association for Computational Linguistics, Minneapolis, Minnesota, USA, pp. 54–63. http://dx.doi.org/10.18653/v1/S19-2007, URL: https://aclanthology.org/S19-2007.

Beddiar, D.R., Jahan, M.S., Oussalah, M., 2021. Data expansion using back translation and paraphrasing for hate speech detection. Online Soc. Netw. Media 24, 100153. http://dx.doi.org/10.1016/j.osnem.2021.100153, URL: https://www.sciencedirect.com/science/article/pii/S2468696421000355.

Bosco, C., Felice, D., Poletto, F., Sanguinetti, M., Maurizio, T., 2018. Overview of the evalita 2018 hate speech detection task. In: EVALITA 2018-Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Vol. 2263. CEUR, pp. 1–9.

Burnap, P., Williams, M.L., 2015. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. Policy Internet 7 (2), 223–242.

Castaño-Pulgarín, S.A., Suárez-Betancur, N., Vega, L.M.T., López, H.M.H., 2021. Internet, social media and online hate speech. Systematic review. Aggress. Violent Behav. 58, 101608. http://dx.doi.org/10.1016/j.avb.2021.101608, URL: https://www.sciencedirect.com/science/article/pii/S1359178921000628.

Chakravarthi, B.R., 2020. HopeEDI: A multilingual hope speech detection dataset for equality, diversity, and inclusion. In: Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media. Association for Computational Linguistics, Barcelona, Spain, pp. 41–53, Online, URL: https://aclanthology.org/2020.peoples-1.5.

Chakravarthi, B.R., 2022a. Hope speech detection in YouTube comments. Soc. Netw. Anal. Min. 12 (1), 75.

Chakravarthi, B.R., 2022b. Multilingual hope speech detection in English and Dravidian languages. Int. J. Data Sci. Anal. 14 (4), 389–406.

Chakravarthi, B.R., M, A.K., McCrae, J.P., Premjith, B., Soman, K., Mandl, T., 2020. Overview of the track on HASOC-offensive language identification-DravidianCodeMix. In: FIRE (Working Notes). pp. 112–120.

Chakravarthi, B.R., Priyadharshini, R., Jose, N., Kumar M, A., Mandl, T., Kumaresan, P.K., Ponnusamy, R., Hariharan, R.L., McCrae, J.P., Sherly, E., 2021. Findings of the shared task on offensive language identification in Tamil, Malayalam, and Kannada. In: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages. Association for Computational Linguistics, Kyiv, pp. 133–145, URL: https://aclanthology.org/2021.dravidianlangtech-1.17.

Conneau, A., Lample, G., 2019. Cross-lingual language model pretraining. Adv. Neural Inf. Process. Syst. 32.

Dai, H., Song, Y., 2019. Neural aspect and opinion term extraction with mined rules as weak supervision. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Florence, Italy, pp. 5268–5277. http://dx.doi.org/10.18653/v1/P19-1520, URL: https://aclanthology.org/P19-1520.

Davidson, T., Warmsley, D., Macy, M., Weber, I., 2017. Automated hate speech detection and the problem of offensive language. In: Proceedings of the International AAAI Conference on Web and Social Media, Vol. 11, No. 1.

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

Fatahillah, N.R., Suryati, P., Haryawan, C., 2017. Implementation of naive Bayes classifier algorithm on social media (Twitter) to the teaching of Indonesian hate speech. In: 2017 International Conference on Sustainable Information Engineering and Technology. SIET, IEEE, pp. 128–131.

Fersini, E., Nozza, D., Rosso, P., 2020. AMI@ EVALITA2020: Automatic misogyny identification. In: EVALITA.

Fortuna, P., Nunes, S., 2018. A survey on automatic detection of hate speech in text. ACM Comput. Surv. 51 (4), 1–30.

Fortuna, P., Soler-Company, J., Wanner, L., 2021. How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets? Inf. Process. Manage. 58 (3), 102524. http://dx.doi.org/10.1016/j.ipm.2021.102524, URL: https://www.sciencedirect.com/science/article/pii/S0306457321000339.

García-Díaz, J.A., Cánovas-García, M., Colomo-Palacios, R., Valencia-García, R., 2021. Detecting misogyny in Spanish tweets. An approach based on linguistics features and word embeddings. Future Gener. Comput. Syst. 114, 506–518. http://dx.doi.org/10.1016/j.future.2020.08.032, URL: https://www.sciencedirect.com/science/article/pii/S0167739X20301928.

Ginting, P.S.B., Irawan, B., Setianingsih, C., 2019. Hate speech detection on twitter using multinomial logistic regression classification method. In: 2019 IEEE International Conference on Internet of Things and Intelligence System. IoTaIS, IEEE, pp. 105–111.

Guest, E., Vidgen, B., Mittos, A., Sastry, N., Tyson, G., Margetts, H., 2021. An expert annotated dataset for the detection of online misogyny. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. Association for Computational Linguistics, pp. 1336–1350, Online, URL: https://aclanthology.org/2021.eacl-main.114.

Haas, J., 2012. Hate speech and stereotypic talk. In: The Handbook of Intergroup Communication. Routledge, pp. 150–162.

Hmeidi, I., Al-Ayyoub, M., Abdulla, N.A., Almodawar, A.A., Abooraig, R., Mahyoub, N.A., 2015. Automatic arabic text categorization: A comprehensive comparative study. J. Inf. Sci. 41 (1), 114–124.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.

İrsoy, O., Cardie, C., 2014. Opinion mining with deep recurrent neural networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. EMNLP, Association for Computational Linguistics, Doha, Qatar, pp. 720–728. http://dx.doi.org/10.3115/v1/D14-1080, URL: https://aclanthology.org/D14-1080.

Joulin, A., Grave, É., Bojanowski, P., Mikolov, T., 2017. Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pp. 427–431.

Karayiğit, H., İnan Acı, Ç., Akdağlı, A., 2021. Detecting abusive instagram comments in turkish using convolutional neural network and machine learning methods. Expert Syst. Appl. 174, 114802. http://dx.doi.org/10.1016/j.eswa.2021.114802, URL: https://www.sciencedirect.com/science/article/pii/S0957417421002438.

Kavitha, K., Shetty, A., Abreo, B., D'Souza, A., Kondana, A., 2020. Analysis and classification of user comments on YouTube videos. Procedia Comput. Sci. 177, 593–598. http://dx.doi.org/10.1016/j.procs.2020.10.084, URL: https://www.sciencedirect.com/science/article/pii/S1877050920323553, The 11th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2020) / The 10th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2020) / Affiliated Workshops.

Keum, B.T., Ahn, L.H., 2021. Impact of online racism on psychological distress and alcohol use severity: Testing ethnic-racial socialization and silence about race as moderators. Comput. Hum. Behav. 120, 106773. http://dx.doi.org/10.1016/j.chb.2021.106773, URL: https://www.sciencedirect.com/science/article/pii/S0747563221000960.

Khairy, M., Mahmoud, T.M., Abd-El-Hafeez, T., 2021. Automatic detection of cyberbullying and abusive language in Arabic content on social networks: A survey. Procedia Comput. Sci. 189, 156–166. http://dx.doi.org/10.1016/j.procs.2021.05.080, URL: https://www.sciencedirect.com/science/article/pii/S1877050921011959, AI in Computational Linguistics.

Khanuja, S., Bansal, D., Mehtani, S., Khosla, S., Dey, A., Gopalan, B., Margam, D.K., Aggarwal, P., Nagipogu, R.T., Dave, S., et al., 2021. Muril: Multilingual representations for indian languages. arXiv preprint arXiv:2103.10730.

Kim, S.B., Han, K.S., Rim, H.C., Myaeng, S.H., 2006. Some effective techniques for naive bayes text classification. IEEE Trans. Knowl. Data Eng. 18 (11), 1457–1466.

Kocoń, J., Figas, A., Gruza, M., Puchalska, D., Kajdanowicz, T., Kazienko, P., 2021. Offensive, aggressive, and hate speech analysis: From data-centric to human-centered approach. Inf. Process. Manage. 58 (5), 102643. http://dx.doi.org/10.1016/j.ipm.2021.102643, URL: https://www.sciencedirect.com/science/article/pii/S0306457321001333.

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., Jackel, L., 1989. Handwritten digit recognition with a back-propagation network. Adv. Neural Inf. Process. Syst. 2.

Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2018. Focal loss for dense object detection. arXiv:1708.02002.

Liu, Z., Lv, X., Liu, K., Shi, S., 2010. Study on SVM compared with the other text classification methods. In: 2010 Second International Workshop on Education Technology and Computer Science, Vol. 1. IEEE, pp. 219–222.

Mandl, T., Modha, S., Kumar M, A., Chakravarthi, B.R., 2020. Overview of the HASOC track at FIRE 2020: Hate speech and offensive language identification in Tamil, Malayalam, Hindi, English and German. In: Forum for Information Retrieval Evaluation. In: FIRE 2020, Association for Computing Machinery, New York, NY, USA, pp. 29–32. http://dx.doi.org/10.1145/3441501.3441517.

Mandl, T., Modha, S., Majumder, P., Patel, D., Dave, M., Mandlia, C., Patel, A., 2019. Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in Indo-European languages. In: Proceedings of the 11th Forum for Information Retrieval Evaluation. pp. 14–17.

Märtens, M., Shen, S., Iosup, A., Kuipers, F., 2015. Toxicity detection in multiplayer online games. In: 2015 International Workshop on Network and Systems Support for Games. NetGames, IEEE, pp. 1–6.

Mencarini, L., 2018. The potential of the computational linguistic analysis of social media for population studies. In: Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media. Association for Computational Linguistics, New Orleans, Louisiana, USA, pp. 62–68. http://dx.doi.org/10.18653/v1/W18-1109, URL: https://aclanthology.org/W18-1109.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., Joulin, A., 2018. Advances in pre-training distributed word representations. In: Proceedings of the International Conference on Language Resources and Evaluation. LREC 2018.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems. pp. 3111–3119.

Mossie, Z., Wang, J.H., 2020. Vulnerable community identification using hate speech detection on social media. Inf. Process. Manage. 57 (3), 102087.

Navada, A., Ansari, A.N., Patil, S., Sonkamble, B.A., 2011. Overview of use of decision tree algorithms in machine learning. In: 2011 IEEE Control and System Graduate Research Colloquium. IEEE, pp. 37–42.

Nugroho, K., Noersasongko, E., Fanani, A.Z., Basuki, R.S., et al., 2019. Improving random forest method to detect hatespeech and offensive word. In: 2019 International Conference on Information and Communications Technology. ICOIACT, IEEE, pp. 514–518.

Pamungkas, E.W., Basile, V., Patti, V., 2020a. Do you really want to hurt me? Predicting abusive swearing in social media. In: Proceedings of the 12th Language Resources and Evaluation Conference. European Language Resources Association, Marseille, France, pp. 6237–6246, URL: https://aclanthology.org/2020.lrec-1.765.

Pamungkas, E.W., Basile, V., Patti, V., 2020b. Misogyny detection in Twitter: a multilingual and cross-domain study. Inf. Process. Manage. 57 (6), 102360. http://dx.doi.org/10.1016/j.ipm.2020.102360, URL: https://www.sciencedirect.com/science/article/pii/S0306457320308554.

Pamungkas, E.W., Basile, V., Patti, V., 2021. A joint learning approach with knowledge injection for zero-shot cross-lingual hate speech detection. Inf. Process. Manage. 58 (4), 102544. http://dx.doi.org/10.1016/j.ipm.2021.102544, URL: https://www.sciencedirect.com/science/article/pii/S0306457321000510.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. PyTorch: An imperative style, high-performance deep learning library. arXiv:1912.01703.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al., 2011. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., 2018. Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). Association for Computational Linguistics, New Orleans, Louisiana, pp. 2227–2237. http://dx.doi.org/10.18653/v1/N18-1202, URL: https://aclanthology.org/N18-1202.

Poletto, F., Basile, V., Sanguinetti, M., Bosco, C., Patti, V., 2021. Resources and benchmark corpora for hate speech detection: a systematic review. Lang. Resour. Eval. 55 (2), 477–523.

Poteat, V.P., Rivers, I., 2010. The use of homophobic language across bullying roles during adolescence. J. Appl. Dev. Psychol. 31 (2), 166–172. http://dx.doi.org/10.1016/j.appdev.2009.11.005, URL: https://www.sciencedirect.com/science/article/pii/S0193397309001129.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C.D., 2020. Stanza: A Python natural language processing toolkit for many human languages. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations.

Ritter, A., Cherry, C., Dolan, W.B., 2011. Data-driven response generation in social media. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Edinburgh, Scotland, UK., pp. 583–593, URL: https://aclanthology.org/D11-1054.

Robinson, D., Zhang, Z., Tepper, J., 2018. Hate speech detection on twitter: Feature engineering vs feature selection. In: European Semantic Web Conference. Springer, pp. 46–49.

Sap, M., Card, D., Gabriel, S., Choi, Y., Smith, N.A., 2019. The risk of racial bias in hate speech detection. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Florence, Italy, pp. 1668–1678. http://dx.doi.org/10.18653/v1/P19-1163, URL: https://aclanthology.org/P19-1163.

Schmidt, A., Wiegand, M., 2017. A survey on hate speech detection using natural language processing. In: Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media. Association for Computational Linguistics, Valencia, Spain, pp. 1–10. http://dx.doi.org/10.18653/v1/W17-1101, URL: https://aclanthology.org/W17-1101.

Sevani, N., Soenandi, I.A., Wijaya, J., et al., 2021. Detection of hate speech by employing support vector machine with Word2Vec model. In: 2021 7th International Conference on Electrical, Electronics and Information Engineering. ICEEIE, IEEE, pp. 1–5.

Shah, K., Patel, H., Sanghvi, D., Shah, M., 2020. A comparative analysis of logistic regression, random forest and KNN models for the text classification. Augment. Hum. Res. 5 (1), 1–16.

Tekiroğlu, S.S., Chung, Y.L., Guerini, M., 2020. Generating counter narratives against online hate speech: Data and strategies. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, pp. 1177–1190. http://dx.doi.org/10.18653/v1/2020.acl-main.110, Online, URL: https://aclanthology.org/2020.acl-main.110.

Urbaniak, R., Ptaszyński, M., Tempska, P., Leliwa, G., Brochocki, M., Wroczyński, M., 2022. Personal attacks decrease user activity in social networking platforms. Comput. Hum. Behav. 126, 106972. http://dx.doi.org/10.1016/j.chb.2021.106972, URL: https://www.sciencedirect.com/science/article/pii/S0747563221002958.

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al., 2020. SciPy 1.0: Fundamental algorithms for scientific computing in Python. Nature Methods 17 (3), 261–272.

Waseem, Z., Hovy, D., 2016. Hateful symbols or hateful people? Predictive features for hate speech detection on twitter. In: Proceedings of the NAACL Student Research Workshop. pp. 88–93.

Wiegand, M., Siegel, M., Ruppenhofer, J., 2018. Overview of the germeval 2018 shared task on the identification of offensive language.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., Rush, A., 2020. Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Association for Computational Linguistics, pp. 38–45. http://dx.doi.org/10.18653/v1/2020.emnlp-demos.6, Online.

Xu, B., Guo, X., Ye, Y., Cheng, J., 2012. An improved random forest classifier for text categorization. J. Comput. 7 (12), 2913–2920.

Yang, Y., Cer, D., Ahmad, A., Guo, M., Law, J., Constant, N., Hernandez Abrego, G., Yuan, S., Tar, C., Sung, Y.-h., Strope, B., Kurzweil, R., 2020. Multilingual universal sentence encoder for semantic retrieval. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. Association for Computational Linguistics, pp. 87–94. http://dx.doi.org/10.18653/v1/2020.acl-demos.12, Online, URL: https://aclanthology.org/2020.acl-demos.12.

Yang, Y., Zheng, X., Ji, C., 2019. Disease prediction model based on bilstm and attention mechanism. In: 2019 IEEE International Conference on Bioinformatics and Biomedicine. BIBM, IEEE, pp. 1141–1148.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., Kumar, R., 2019a. Predicting the type and target of offensive posts in social media. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota, pp. 1415–1420. http://dx.doi.org/10.18653/v1/N19-1144, URL: https://aclanthology.org/N19-1144.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., Kumar, R., 2019b. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In: Proceedings of the 13th International Workshop on Semantic Evaluation. Association for Computational Linguistics, Minneapolis, Minnesota, USA, pp. 75–86. http://dx.doi.org/10.18653/v1/S19-2010, URL: https://aclanthology.org/S19-2010.

Zeinert, P., Inie, N., Derczynski, L., 2021. Annotating online Misogyny. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, pp. 3181–3197. http://dx.doi.org/10.18653/v1/2021.acl-long.247, Online, URL: https://aclanthology.org/2021.acl-long.247.

Zhang, Y., Jin, R., Zhou, Z.H., 2010. Understanding bag-of-words model: a statistical framework. Int. J. Mach. Learn. Cybern. 1 (1–4), 43–52.