Social Network Analysis and Mining (2024) 14:70
https://doi.org/10.1007/s13278-024-01223-y

REVIEW PAPER

A survey of hate speech detection in Indian languages

Arpan Nandi[1] · Kamal Sarkar[1] · Arjun Mallick[1] · Arkadeep De[1]

Received: 30 April 2023 / Revised: 14 January 2024 / Accepted: 7 February 2024

Abstract
With the enormous increase in accessibility of high-speed internet, the number of social media users is increasing rapidly. Due to a lack of proper regulations and ethics, social media platforms are often contaminated by posts and comments containing abusive language and offensive remarks toward individuals, groups, races, religions, and communities. A single remark often triggers a huge chain of reactions with similar abusiveness, or even more. To prevent such occurrences, there is a need for automated systems that can detect abusive texts and hate speeches and remove them immediately. However, most existing research works are limited only to globally popular languages like English. Since India is a nation of many diverse languages and multiple religions, nowadays abusive posts and remarks in Indian languages (monolingual or code-mixed form) are not infrequent on social media platforms. Although resources such as hate speech lexicon and annotated datasets are limited for Indian languages, most research works on hate speech detection in such languages used traditional machine learning and deep learning methods for this task. However, multilingualism and code-mixing make hate speech detection in Indian languages more challenging. Given these facts, this paper mainly focuses on reviewing the latest impactful research works on hate speech detection in Indian languages. In this paper, we have analyzed and compared the latest research works on hate speech detection in Indian languages in terms of various aspects—datasets used, feature extraction and classification methods applied, and the results achieved.

Keywords Hate speech detection · Abusive comments · Indian languages · Mixed languages · Code-mixed

1 Introduction

Cambridge Dictionary defines hate speech as "public speech that expresses hate or encourages violence toward a person or group based on something such as race, religion, sex, or sexual orientation". Often statements like that do not refer to the literal meaning, but they hint or refer to some sort of abuse or violence. There are cases, where the text contains some slang, but does not intend any hate. Such comments are not classified as hate speech. On the other hand, there could be instances where no abusive language has been directly used, but hate is intended symbolically. In this context, we can say that hate speech is a more vast idea that can be caused by a lot of variables, whereas abusive language is a specific case that can make a text count as hate speech. They are not perfectly synonymous, but they refer to the same idea in this study. We understand that any content expressing hate and abuse should not be encouraged by the popular online platforms, where a lot of people from diverse cultures come together. Such comments are hurtful and often spark reactions. Even the tiniest of remarks are capable of triggering the ugliest scenarios.

With the gift of the internet now accessible to almost everyone, more than ever, we have seen an exponential increase in the number of social media and online content delivery platforms. Simultaneously the participation of common people increased manifold. With such enormous engagement of people and unrestricted freedom of speech, genuine issues started to pop up. Conflict of opinion and ideology is very common in human society, but when scaled

✉ Kamal Sarkar
jukamal2001@yahoo.com

Arpan Nandi
arpannandi12@gmail.com

Arjun Mallick
arjunmallick99@gmail.com

Arkadeep De
arkadeepde142@gmail.com

[1] Department of Computer Science and Engineering, Jadavpur University, Raja S C Mallick Road, Kolkata, West Bengal 700032, India

Published online: 26 March 2024

into communication between large groups they often turn into a chain of verbal abuse and offensive remarks. This is exactly what started to happen soon, in almost all platforms. As undisciplined and irresponsible users could not be restricted from using the platforms, they came up with a solution where the abusive texts could be automatically detected using machine learning, and those would be removed immediately or would not be allowed to post. The natural language processing frameworks improved over the years and thus the platforms improved their algorithms as well. But most of these research works were focused on popular global languages, like English (Del Vigna12 et al. 2017; Mathew et al. 2020).

As most of these platforms are meant for casual interaction, a major fraction of the communication is found to be in local languages. Indian languages are often written in corresponding scripts, other scripts, or often in code-mixed forms. Grammar rules, syntax, semantics, and usage vary a lot from one language to another language, and hence research that succeeded for one language did not align with that of other languages. Moreover, many languages do not have sufficient annotated datasets to train deep models. Collecting data for such tasks itself is a laborious and time-consuming task. Other than that, there are regional accents associated with most local languages, which affect the way they are written in texts. Due to such reasons, the success of hate speech detection in Indian languages is not like that of the English language.

This study, therefore, aims to bring together all the innovative and impactful research works in the domain of hate speech detection, in the context of various Indian languages that have been done in the past several years. The major contributions of this study are as follows:

- Gathering and thoroughly studying all the high-impactful research papers that aim to detect hate speech and abusive language in Indian languages like Hindi, Bengali, Tamil, Telugu, Marathi, Malayalam, etc.
- Surveying detailed information about the available datasets used in various existing studies.
- Studying and analyzing the preprocessing, machine learning, and deep learning methods used by each of past research works, and classifying the existing methods.
- Comparing the results obtained in the published research papers, drawing certain conclusions, and finding possibilities for further research.

The overall paper is organized as follows: Section 2 presents how our survey differs from the existing surveys on hate speech detection in Indian languages. Section 3 describes the procedure of collecting research papers used for carrying out this survey work and highlights the types (journal or conference) of research papers, publishers, and year-wise counts. Section 4 provides a detailed description of all the datasets for Indian language hate speech detection that falls within the scope of this study. Section 5 discusses the major approaches undertaken in the existing studies for hate speech detection from Indian language texts. This section also elaborates on data preprocessing, traditional machine learning, deep learning, and ensemble models for hate speech detection. Section 8 provides an overall comparison among the existing models for hate speech detection in various Indian languages. Section 9 criticizes the current research trends on hate speech detection in Indian languages. Section 10 concludes and highlights future research directions.

2 Related works

Numerous surveys have been done in this domain of hate speech detection, but most of them are concentrated around hate speech detection in global languages like English, and a few have focused on hate speech detection in other European languages (Schmidt and Wiegand 2017; Naseem et al. 2021; Alrehili 2019). These surveys have thoroughly analyzed a large number of existing works, the obstacles, and issues, and helped new researchers to better formulate their targets. But we find that the number of surveys significantly decreases when we come to the context of Indian languages, due to a lack of maturity in the approaches for hate speech detection in Indian languages. However, there are a few survey papers that match the context of this study. A generalized review on hate speech detection (Poletto et al. 2021) identifies that the datasets of hate speech detection in Hindi and Hindi–English code-mixed data are very good examples of informal communication in local languages and that they are considerably different from the English datasets.

The Works in Dowlagar and Mamidi (2021) have studied how neural networks have rapidly evolved to detect hate speech in code-mixed multilingual data. There are a considerable number of research papers that have included many languages in their domain of study. Dhanya and Balakrishnan (2021) did consider major Asian languages and tried to figure out which is the best approach for hate speech detection task. They also tried to analyze the relationship between classification accuracy in this context and other parameters like the quality and size of vocabulary and datasets. Some research provides a fresh point of view, like joint modeling of emotional and abusive language detection. Sentiment analysis and abusive language detection are two different problems but they have a lot in common, and in a study, Rahman et al. (2022), the authors decided to jointly model them and they used a Bengali dataset for this task.

The main distinctive features of our survey are as follows:

- Our survey of the latest research works on hate speech detection in Indian languages is more systematic and organized
- Our survey has been conducted on three different aspects of hate speech detection in Indian languages—Machine learning and deep learning-based approaches, availability of datasets in Indian languages, and comparison of the results reported in the research literature.

3 Collecting research papers for the survey

For our study, we collected a large volume of research papers which have been published between 2017 and 2022. We searched for papers that contained certain keywords and had objectives that aligned with our purpose. While collecting them, the emphasized phrases were "detection of hate speech", "abusive language", "offensive texts", "aggression", and "abuse". We also used certain domain-specific keywords—"misogyny", "homophobia", "Islamophobia", etc., which helped us find some works like (Chakravarthi 2022; Khan and Phillips 2021; Barnwal et al. 2022) that were focused on certain domains of hate speech.

After initially surfing through the papers, we downloaded a collection of around 70 research papers related to hate speech detection in Indian languages, from which we filtered out 30 research works and considered them for the survey. For filtering them, we carefully went through parameters like the impact factor of the journal, the number of citations, the quality and detailing of the presentation, the novelty of the approach or objective, and the performance achieved. A summary chart has been provided in Table 1, which shows the year of publication, type of publication (journal or conference), and the total count of considered papers published in a certain year. A glance at Table 1 reveals that the number of studies in this domain, in terms of impact as well as volume, has generously increased in the last 5 years, with its importance increasing rapidly.

From all the downloaded papers which are related to the scope of this study, we have classified the papers into the following areas.

- The available datasets.
- The major approaches used for hate speech detection in Indian languages.
- The metrics and measures used for evaluating the hate detection models.
- And finally an overall comparison of all the papers in terms of results reported in the papers.

Table 1 A summary of all the papers that were considered for this study, in the span of the year 2017–2022

Year	Publisher	Type	Number of studies	Total count
2023	Springer	Journal	3	3
2022	Springer	Journal	5	19
	Elsevier	Journal	4	
	IEEE	Conference	6	
	ACL	Conference	3	
	Arxiv	Journal	1	
2021	Springer	Journal	2	8
	ACM	Conference	1	
	IEEE	Conference	2	
	ACL	Conference	1	
	Arxiv	Journal	1	
	SemEval	Conference	1	
2020	ACM transactions	Journal	1	12
	ACM	Conference	1	
	Elsevier	Journal	2	
	ACL	Conference	2	
	IEEE	Conference	1	
	FIRE	Conference	1	
		Workshop	3	
	Arxive	Conference	1	
2019	FIRE	Conference	1	4
	ACL	Conference	1	
	IEEE	Conference	1	
	IEEE	Conference	1	
2018	ACL	Conference	1	4
	IEEE	Conference	1	
	TRAC	Conference	1	
	Arxiv	Journal	1	
2017	IEEE	Conference	1	1

4 Datasets for hate speech detection in Indian languages

Datasets are important materials used for training, validating, and testing machine learning and deep learning models. Although the publicly available datasets for hate speech detection in English are abundant, the amount of publicly available datasets for hate speech detection in Indian languages is still limited.

India is a multilingual country. This has made the People of India able to communicate in several languages. For an Indian language, dialects of the language vary with geography and culture. Many of these are regularly used in social media and other online platforms, but they are mostly low-resource languages. Low-resource languages do not have an adequate amount of organized data capable of training machine learning or deep learning models. So while working

on hate speech detection tasks in such low-resource languages, many researchers prefer to collect texts from online platforms and build their own datasets.

There are many papers (Eshan and Hasan 2017; Ishmam and Sharmin 2019; Islam et al. 2022), which have considered crawling into platforms like Twitter, Facebook, and YouTube for collecting context-specific posts and comments. The initial collection pool is usually huge as there are a lot of impurities and most of them are discarded to leave out the useful data. The impurities like URLs, irrelevant texts, and characters are usually removed. After that, the remaining corpus is refined and organized into usable data. For supervised learning, accurate labeling of the training data is very important as this directly impacts the results. So, the cleaned data are now labeled. The labeling of data has been done manually in almost all the papers that we studied. It is either done by the researchers themselves or through some public survey. In the case of a public survey, common people are asked to label data. This method is used when the target is to develop a large dataset, but available manpower is limited. A small subset of the unlabeled data is rolled out as forms in public forums, asking common people to label them as they think. In such cases, the final label for a particular text can be decided by voting.

In Table 2, we have presented detailed information regarding the datasets that are related to hate speech and abusive language detection in Indian languages. Not all the datasets mentioned in this table are available for public use by other researchers. The main reason mentioned by the researchers is to preserve certain privacy terms of the various online platforms from where they collected the data. Some specific studies have solely focused on building datasets for hate speech detection and making them available for other researchers.

5 Major approaches to hate speech detection

After a literature survey, we observed that major approaches to abusive language and hate speech detection in Indian languages used traditional machine learning algorithms and deep learning algorithms. Therefore, we have classified the major approaches into two types (1) traditional machine learning-based approaches, and (2) deep learning-based approaches. In this section, we will discuss the major approaches to hate speech detection in Indian languages.

5.1 Traditional machine learning-based approaches to hate speech detection in Indian languages

Among traditional machine learning (ML) algorithms, the Multinomial Naive Bayes (MNB) classifier, which was widely used for the text classification task, is also used for hate speech detection in Indian languages (Akhter et al. 2018; Rani et al. 2020; Subramanian et al. 2022). Other popular machine learning algorithms used for hate speech detection in Indian languages are Support Vector Machines(SVM) Eshan and Hasan (2017), the k-nearest neighbors (KNN) search algorithm (Rani et al. 2020), and Random Forest (Ishmam and Sharmin 2019). These ML algorithms take input in the form of vectors and predict the class it should belong to. To make the input text suitable to feed to an ML algorithm, it needs to be converted to a feature vector which is a vector of feature values where the features are manually engineered.

A generic framework for machine learning-based hate speech detection models used in the above-mentioned research papers is shown in Fig. 1. In general, the machine learning-based approaches involve several steps which are: (1) preprocessing, (2) feature extraction, and (3) classification of texts into hate or non-hate.

5.1.1 Preprocessing

Social media data are usually unstructured and noisy and they may contain spelling and grammatical errors. Therefore, it is preprocessed before feature extraction. The analysis of this unstructured data to get insights about the opinion and the sentiment of the general crowd is known as sentiment analysis (Zhang and Liu 2012). The preprocessing step also reduces the dimensionality of input data by removing useless words that have no less power to discriminate between hate speech and non-hate speech. Such words are called stop words (e.g., prepositions, articles, punctuation, and special characters). The preprocessing step consists of several smaller steps as follows:

- Tokenization: In this case, the text is broken into smaller elements called tokens (e.g., text into words);
- Stop word removal: After tokenization, stop words are removed.
- Stemming and Lemmatization: To deal with the data sparseness problem, the words are converted into base forms using the stemming or the lemmatization method. The difference between stemming and lemmatizing is that stemming often reduces words to forms that may be meaningless. For example, stemming drops the 'ing' from some action words and produces words that are not found in the dictionary. The stemming process produces 'runn' from 'running', 'ris' from 'rising', 'mov' from 'moving', etc. On the other hand, the lemmatization process can reduce a given word to a dictionary word, for example, using lemmatization, we obtain 'run' from 'running' and 'move' from 'moving'.

Table 2 Datasets created and used in all the research works that have been studied

Dataset name/paper of first use	Link	Open or closed?	Dataset description
A Dataset of Hindi–English Code-Mixed Social Media Text for Hate Speech Detection	https://github.com/punyajoy/HateSpeech-Hindi-English-Code-Mixed-Social-Media-Text	Open	Total: 4575 Hindi–English code-mixed tweets. Hate: 1661, Non-Hate: 2914
The Bengali hate speech dataset	link is unavailable. The detail of this dataset can be found in Karim et al. (2021)	Closed	Total:8087 texts collected from facebook and YouTube. Personal: 3537, Geopolitical: 2364, Religious: 1211, Political: 999
Hate Speech Detection in the Bengali Language: A Dataset and Its Baseline Evaluation	https://link.springer.com/chapter/10.1007/978-981-16-0586-4_37	Open	Total: 30k Bengali comments from YouTube and Facebook. Hate: 10k, Non-Hate: 20k
BD-SHS: A Benchmark Dataset for Learning to Detect Online Bangla Hate Speech in Different Social Contexts	https://github.com/naurosromim/hate-speech-dataset-for-Bengali-social-media	Open	Total: 50,281 Bengali comments. Hate:24,156, Non-Hate:26,125
HEOT(Hindi–English Offensive Tweet)	link is unavailable. The detail of this dataset can be found in Mathur et al. (2018)	Closed	Total: 3679 Hindi–English code-mixed tweets. Non-offensive: 1414, Abusive: 1942, Hate-inducing: 323
HASOC-Dravidian-CodeMix	https://sites.google.com/view/dravidian-codemix-fire2020/overview	Open	Malayalam–English code-mixed dataset: Total 5k posts of code-mixed comments/posts in Malayalam–English (offensive class: 2465, Not-class: 2535). Tamil–English code-mixed dataset: Tamil–English 5k posts (offensive class: 2455, Not-class: 2485)
Devanagari Hindi Offensive Tweets (DHOT) data corpus	link is unavailable. The details of this dataset can be found in Jha et al. (2020)	Closed	Total: 2000 Hindi–English code-mixed tweets. Abusive: 500, Non-abusive: 1500
HASOC 2019 (hindi)	https://hasocfire.github.io/hasoc/2019/dataset.html	Open	Total: 4665 Hindi–English code-mixed tweets. Offensive: 2469, Non-offensive: 2196
L3Cube-MahaHate (marathi)	https://github.com/l3cube-pune/MarathiNLP/tree/main/L3Cube-MahaHate	Open	Total: 25,000 Marathi tweets. Four classes are Hate, Offensive, Profane, and Non-Hate, each class contains 6250 tweets
Aggression-annotated Corpus of Hindi–English Code-mixed Data	link is unavailable. The detail of this dataset can be found in Kumar et al. (2018)	Closed	Dataset-1: total 18k Hindi–English code-mixed tweets with class distribution—Overtly aggressive: 1080, Covertly aggressive: 7938, Non-aggressive: 8982. Dataset-2: total 21k Facebook comments with class distribution—Overtly aggressive: 5775, Covertly aggressive: 6279, Non-aggressive: 8946
Tamil–English code-switched, sentiment-annotated corpus	https://github.com/bharathichezhiyan/TamilMixSentiment	Open	Total: 15,744 Tamil–English code-mixed comments collected from YouTube with class distribution- Positive: 10,559, Negative: 2037, Mixed feelings: 1801, Neutral: 850, Other Language: 497
Hostility Detection Dataset in Hindi	link is unavailable. The detail of this dataset can be found in Das et al. (2022)	Closed	Total: 8192 posts. Hostile: 3834 and Non-hostile: 4358
TRAC-2 dataset	https://sites.google.com/view/trac2/shared-task	Open	Total: 20k annotated data collected from social media for each language: Bangla, Hindi, and English. Class distribution in each dataset is: Overtly aggressive: 3,954, Covertly aggressive: 3,366, and Non-aggressive: 12,127

Table 2 (continued)

Dataset name/paper of first use	Link	Open or closed?	Dataset description
Developing a Multilingual Annotated Corpus of Misogyny and Aggression	link is unavailable. The detail of this dataset can be found in Bhattacharya et al. (2020)	Closed	Total: 12.5k YouTube comments in Hindi, English, and Bengali and code-mixed with labels for misogyny(positive: 2092, negative: 9981) and aggression(covert: 2300, overt: 2200 and non-aggressive: 7500)

Depending upon the data format, the preprocessing step may also involve other operations like removing repeated characters in the noisy social media text. For example, the English word "good" from "goood", the Bengali word "khub" (very) from Khuuuub, etc.

5.1.2 Features and ML models for hate speech detection

The most common features that are used for hate speech detection using traditional ML models are word n-grams and character n-grams (Eshan and Hasan 2017; Akhter et al. 2018; Sarker et al. 2022; Bohra et al. 2018). When n-gram features are considered, n is varied from 1 to some limit. When n is set to 1, only 1-gram (unigram) features are considered. Thus, if *n* is varied from 1 to 3, unigrams, bigrams, and trigrams features are taken into consideration. For word n-gram features, varying *n* up to 3 is useful, but the n-grams larger than trigrams are not shown effective in hate speech detection. For the character n-gram features, *n* can be varied to the limit larger than that is used for the word n-gram features, but it is not useful to take small n-grams consisting of one or two characters Sarkar (2018).

When the hand-crafted features are used, the input text is represented using the bag of words model, where an input text is considered a bag of words. However, in this method, an input text is converted into a higher dimensional vector where each component of the vector corresponds to the TF*IDF weight of a vocabulary word occurring in the input text. The TF*IDF weight of a word is calculated by the product of term frequency and inverse document frequency where the term frequency (TF) is the number of times a word occurs in the input text and inverse document frequency (IDF) is calculated as log (N/DF), N= number of texts in the training corpus and DF is called document frequency which is the number of input texts containing the word at least once. Many prior studies on hate speech detection in Indian languages that used traditional ML algorithms have used term frequency (Rani et al. 2020), and TF-IDF (Islam et al. 2022) as their primary feature extraction method. When n-gram features are used, the input text is represented as a bag of n-grams where each n-gram is a term and the TF-IDF vectorization method mentioned above is used for the input text representation.

The other features that have been considered for hate speech detection in Indian languages are emoticons, word count, character count, punctuation density, vowel density, unique word count, and capitalization information (Bohra et al. 2018). The count or density of some specific symbols based on context can also be considered, for example, the number of question marks or exclamation marks or a particular word. When these features are considered, they are usually combined with the n-gram features. Ishmam and Sharmin (2019) used hashtags, URLs, comment length,

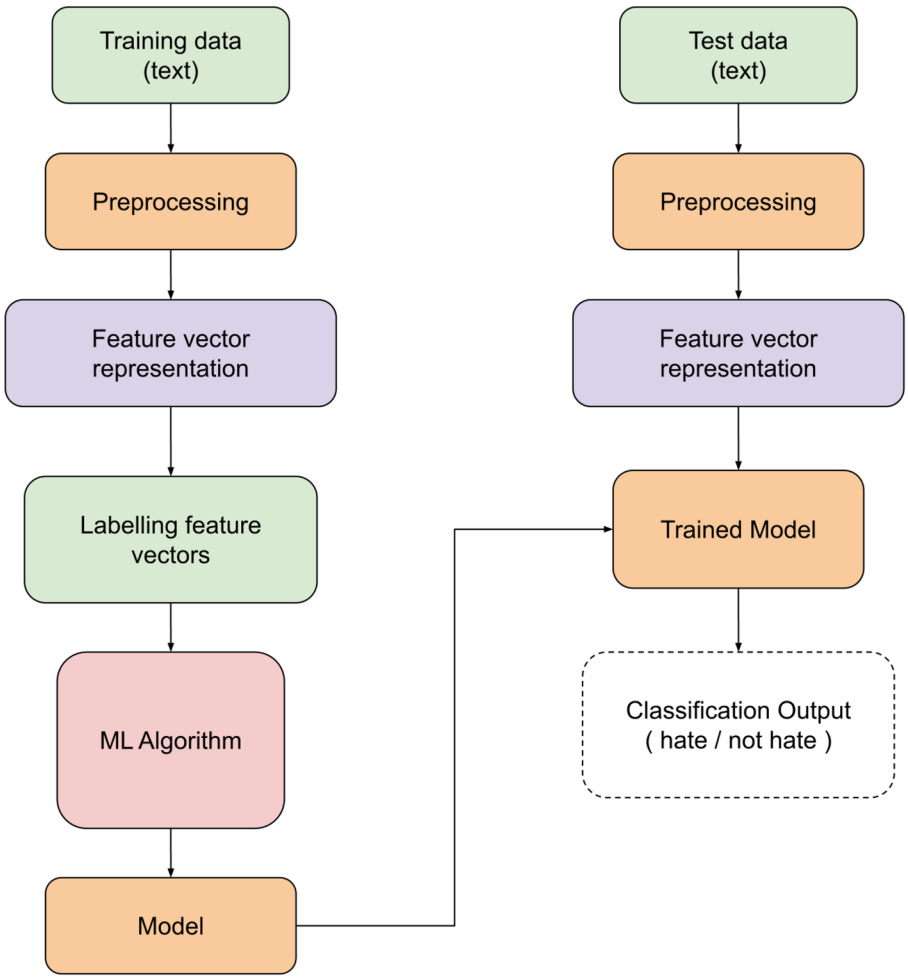

Fig. 1 A general framework for the traditional machine learning models used for hate speech detection

word length, and average syllables as the additional features with the n-gram features for Bengali hate speech detection.

As we can see from the generic machine learning framework for hate speech detection shown in Fig. 1, feature extraction is done using either a set of hand-crafted features or the abstract features generated using an unsupervised pre-trained model. After feature representation, each input text is converted to a numeric feature vector which is fed into an ML model that learns to classify input text as hate speech or not. In some works, an ensemble of ML models has also been used for hate speech detection (Sarker et al. 2022).

The most commonly used machine learning models for hate speech detection tasks are linear regression(LR) (Sarker et al. 2022; Islam et al. 2022), Multinomial Naive Bayes (MNB) (Subramanian et al. 2022; Rani et al. 2020; Islam et al. 2022; Sarker et al. 2022), k-nearest neighbors (KNN) (Sarker et al. 2022; Jemima et al. 2022; Islam et al. 2022), support vector machine (SVM) (Sreelakshmi et al. 2020; Eshan and Hasan 2017; Akhter et al. 2018; Remon et al. 2022), decision trees (DT) (Eshan and Hasan 2017; Akhter et al. 2018; Rani et al. 2020), etc. Previous studies (Eshan and Hasan 2017; Akhter et al. 2018) have shown that SVM provides the best results among the single ML algorithms that have been used for hate speech detection.

The ensemble models based on decision trees, like random forest (RF) classifier (Sarker et al. 2022; Ishmam and Sharmin 2019; Islam et al. 2022), gradient boosting (Kamble and Joshi 2018), etc., often produce better results than the single ML algorithm. Anusha and Shashirekha (2020) presented an ensemble method that combines Random forest, Gradient boost, and XGboost classier through voting for hate speech detection in three languages, English, German, and Hindi.

In Table 3, we present a summary of research works on hate speech detection in Indian languages using traditional ML approaches. In this table, we have shown the language domain, the feature extraction methods used, and the ML algorithm used. We have used the following short names for the ML algorithms shown in this table. SVM: Support Vector Machines, DT: Decision Tree, RF: Random Forest, LR: Logistic Regression, MNB: Multinomial Naive Bayes, KNN: K-nearest neighbor, and SVM-RBF: Support Vector Machines with Radial Basis Function.

Table 3 A summary of traditional ML-based approaches used for hate speech detection in Indian languages

Paper title	Year/author	Publisher	Language focus	Feature extraction method	Machine learning Algorithms Used
Social media bullying detection using machine learning on Bangla text	Akhter et al. (2018)	IEEE	Bengali	Trigram language model, Linguistic features	SVM, DT
Hateful speech detection in public Facebook pages in Bengali language	Ishmam and Sharmin (2019)	IEEE	Bengali	5 types of frequency-based features	RF
Hate Speech Detection Using Machine Learning In Bengali Languages	Islam et al. (2022)	IEEE	Bengali	Count vectorizer, TF-IDF	LR, MNB, SVM, KNN, RF
Bengali Hate Speech Detection in Public Facebook PAGES	Remon et al. (2022)	IEEE	Bengali	Fast Text embedding	SVM
An application of Machine Learning to Detect Abusive Bengali Text	Eshan and Hasan (2017)	IEEE	Bengali	Trigram LM, TF-IDF	SVM, DT
Detection of Hate Speech Text in Hindi–English Code-mixed Data	Sreelakshmi et al. (2020)	Elsevier	Hindi–English code mixed	Facebook's pre-trained embedding and Fast Text	SVM-RBF
A Comparative Study of Different State-of-the-Art Hate Speech Detection Methods for Hindi–English Code-Mixed Data	Rani et al. 2020	ACL	Hindi–English Code mixed	Term Frequency	SVM, MNB, KNN, DT
Detecting Offensive Tamil Texts Using Machine Learning And Multilingual Transformer Models	Subramanian et al. (2022)	IEEE	Tamil, Malayalam	BERT-embeddings	MNB, SVM, LR, KNN
A Machine Learning Approach to Classify Anti-social Bengali Comments on Social Media	Sarker et al. (2022)	IEEE	Bengali	unigrams, bigrams, trigrams	LR, RF, MNB and SVM
L-Boost: Identifying Offensive Texts From Social Media Post in Bengali	Mridha et al. (2021)	IEEE	Bengali and Banglish	BERT-embeddings	AdaBoost

LR: Logistic Regression, SVM: Support Vector Machine, MNB: Multinomial Naive Bayes, BNB: Bernoulli Naive Bayes, GNB: Gaussian Naive Bayes, DT: Decision Tree, KNN: k-nearest neighbors, RF: Random Forest

5.2 Deep learning-based approaches

With the increased availability of data, computation power, and unprecedented success of deep learning models in various applications, like English languages, deep learning (DL) models have also become state-of-the-art models for various natural language recognition tasks in Indian languages such as sentiment analysis (Chakravarthi et al. 2022; Meetei et al. 2021), emotion recognition (Kumar et al. 2023), emoji prediction (Himabindu et al. 2022), and hate speech detection. It is generally observed that the DL models largely outperformed traditional machine learning models in the domain of hate speech detection. In this section, the major deep learning-based approaches used for hate speech detection in Indian languages are discussed.

The deep learning-based approaches to hate speech detection have been classified into two types, (1) Word embeddings-based approaches and (2) transfer learning-based approaches.

5.2.1 Word embeddings-based approaches

When the dataset is small, the manually crafted features with traditional machine learning algorithms may not produce an acceptable performance. In this case, the deep learning-based unsupervised pre-trained embedding models like the Word2Vec model (Mikolov et al. 2013) are useful in extracting high-level abstract features from a text. Word vectors extracted from such models, when fed to traditional ML classifiers often produce better results than hand-crafted features. After the dataset is preprocessed as required, the processed dataset is passed through some embedding model. The embedding model transforms the words or characters into corresponding real-valued vectors. Many existing works on hate speech detection used different types of embedding techniques. The papers Remon et al. (2022), Jha et al. (2020) and Sreelakshmi et al. (2020) used the fastText embedding (Grave et al. 2018a, b). Ishmam and Sharmin (2019) used Word2Vec embedding features and Gated Recurrent Unit (GRU) Neural network for hate speech detection in the Bengali language from Facebook pages. This model performed better than the traditional ML algorithms. Mathur et al. (2018) uses a pre-trained embedding model with CNN for detecting offensive tweets in Hindi–English code-mixed language.

Joshi et al. (2021) passed fastText word embedding to various deep learning models such as multichannel CNN, BiLSTM, and a combination of CNN and BiLSTM for hostility detection in the Hindi language.

Kamble and Joshi (2018) suggested domain-specific word embedding to use in the traditional deep models like multichannel CNN, LSTM, and BiLSTM for hate speech detection in English–Hindi code-mixed tweets. They reported in the paper that multichannel 1D CNN performed the best among other deep models.

Sarker et al. (2022) used Gated Recurrent Unit (GRU) for classifying online social media comments into social or anti-social. They compared the GRU-based system with traditional machine learning algorithms like Random Forest (RF), Multinomial Naive Bayes (MNB), etc., and reported that the performance of GRU was worse than MNB because the dataset was limited.

Remon et al. (2022) used FastText word embedding in CNN and LSTM for hate speech detection, but they observed that SVM with RBF kernel performed better than CNN or LSTM.

Mundra and Mittal (2023) combined word embedding and character embedding to obtain hybrid embedding-based feature representation which is fed to a BiLSTM + attention network for developing a hate speech detection model that can identify aggression in Hindi–English code-mixed text. In another work, Mundra and Mittal (2022) also used the hybrid embedding-based feature representation, but this work fuses the outputs of BiLSTM and 1D CNN via the attention mechanism and feeds it to the dense layer for classification.

Our literature survey on hate speech detection in Indian languages reveals that most researchers prefer to use deep learning models like CNN, LSTM, BiLSTM, GRU CNN+LSTM, CNN+BiLSTM along with static word embeddings like Word2Vec or fastText embeddings. The possible reasons for the better accuracy achieved by such deep learning models are transfer learning via word embeddings and the more expressive representation of the sequential input.

In Fig. 2, we have presented a generic architecture for a static word embedding-based deep model for hate speech detection in Indian languages. This model has several steps (1) Input processing, (2) Word embedding, (3) using deep learning models like LSTM, BiLSTM, or CNN for an

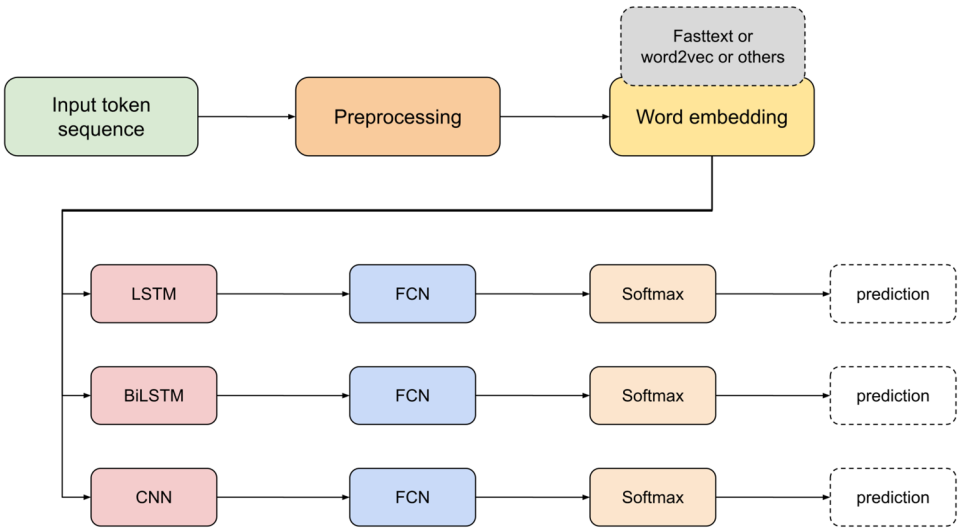

Fig. 2 A General framework for static word embedding-based deep learning models for hate speech detection

effective contextual embedding of the input sequence, (4) Using the fully connected network (FCN) for extracting higher-level abstract features, and (5) using Softmax layer for producing probability distribution over output classes, hate, non-hate, or others.

When the embedding is applied, usually minimal preprocessing is done because the entire corpus is provided to the word embedding model. Sometimes, stop words and noisy characters are removed before submitting the corpus to the embedding model. Most deep learning models that use embeddings include CNN, LSTM, BiLSTM, and their variants. Various types of word embedding such as word embedding, character embedding, and/or subword-based embedding are used. Although the transformer model has a deep learning architecture that is very different from CNN or LSTM, it has also a character n-gram-based embedding layer.

In Fig. 2, we have shown a CNN-LSTM model that uses a multichannel CNN model for extracting features(similar to n-grams) that are fed to the LSTM units. In this case, a multichannel CNN model extracts sequence features similar to n-grams, whereas LSTM learns sequence order. A multichannel CNN model differs from the traditional 1D CNN which has a word embedding layer, one-dimensional convolutional layer, dropout layer, max pooling, and flatten layer. The 1D multichannel convolutional neural network (1D multichannel CNN) is a variation of the basic 1D CNN model with varied sizes of kernels. This allows to processing of a document in different granularity using different n-grams at once, such as unigrams, bigrams, trigrams, and 4-grams. In the multichannel CNN version, several channels are defined for distinct n-grams. For example, if N kernels are used and the input document contains k words where each word is represented as an embedding vector, and the window size and padding input are adjusted in such a way that the output has the same length as the original input, the 1D multichannel CNN produces a feature map k x N. Using 1D max pooling operation with a pool size of 2 along the word dimension, it is reduced to (k/2) x N. This is now fed to an LSTM layer with K/2 units. In the figure, what we have shown as an LSTM layer is nothing but a single LSTM layer with multiple LSTM units.

Instead of directly using a recurrent network and using the embedded text for these models, we can use an additional feature extractor that could extract some more meaningful and contextual features from the embedded text. Then those trained features could be used to train the recurrent models. As a feature extractor, we can use a one-dimensional convolution layer followed by pooling. Dutta et al. (2021) developed a CNN-LSTM hybrid model for hate speech detection for multilingual, Hindi, Meitei, and Bengali datasets. In Vashistha and Zubiaga (2020) a similar hybrid model was also applied for hate speech detection in Hindi tweets. In Fig. 3, a hybrid CNN-LSTM model architecture for Indian language hate speech detection is shown. In this model, CNN employs multiple filters for feature extraction using local contexts of words and LSTM combines the local features for capturing the temporal order in the input sequence. Thus, a better representation of the input sequence is obtained which is then passed to a fully connected layer followed by a softmax layer.

5.2.2 Neural language model-based approaches

In recent years, neural language model-based approaches have become very successful in many natural language process tasks. The main reason for the success of this kind of approach is that the underlying language model is trained on a huge amount of text and when a language model is connected to a deep neural network, the obtained model can be fine-tuned using some amount of labeled data for achieving better performance in the domain under consideration. Thus, the use of a language model in the text classification process alleviates the data scarcity problem. This is also called transfer learning (Pan and Yang 2010) because the knowledge captured by a large neural language model trained on a large corpus can be transferred to the model used for text classification in various domains.

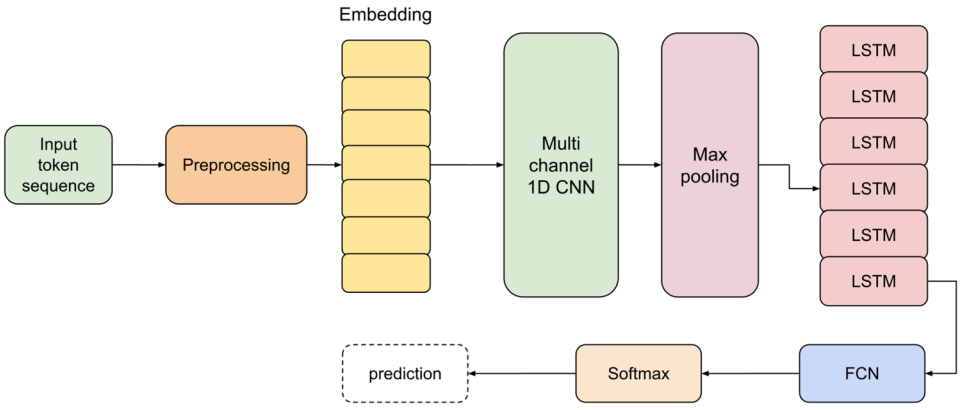

Fig. 3 A hybrid CNN-LSTM model for hate speech detection

Most of the recently used language models are based on transformers (Vaswani et al. 2017). The transformer is a neural network model that uses self-attention mechanisms for producing contextualized embeddings of the words in an input sequence or the contextualized embedding of the entire input sequence of words. The most commonly used language model that uses an encoder mechanism for language modeling is the BERT (Bidirectional Encoder Representations from Transformers). Biradar and Saumya (2022) used various transformers like IndicBERT, mBERT, ULMFIT for hate speech detection in Hindi–English code-mixed texts. They applied a machine translation system for translating the input texts to a common Devanagari script before applying the BERT model. Joshi et al. (2021) compared the performance of indicBERT and mBERT for hostility detection in Hindi.

Patil et al. (2022) used various BERT models for hate speech detection in the Marathi language. Another work on hate speech detection in Marathi language was done by Zampieri et al. (2022). In this work, the authors introduced a Marathi Offensive Language Dataset called MOLD 2.0. They reported the baseline results on the MOLD 2.0 dataset through experimentation using the support vector classifier (SVC), some deep learning models (CNN and BiLSTM), and some transformer models (mBERT, XLM-R, and IndicBERT).

Our survey reveals that the BERT models are used in hate speech detection in two different ways (1) freeze mode and (2) fine-tuned mode. In the freeze mode, the weights of the BERT model are not changed, but the weights of the connections from BERT to the output softmax layer are only trained in the supervised mode for developing the system. On the other hand, in the system that uses a fine-tuned BERT, the weights of the BERT model are allowed to be fine-tuned when the network is trained in the supervised mode using the hate speech training data. The generic framework for the BERT-based hate speech detection model is given in Fig. 4.

Sharma et al. (2022) presented a hate speech detection model for the English–Hindi code-mixed languages. They used language identification, mapping from Roman to Devanagari language, and a multilingual BERT model called MuRIL for hate speech detection. Bharathi and Varsha (2022) compared several transformer models for hate speech detection for the Tamil language. They trained three transformer models—BERT, mBERT, and XLNET and their results revealed that BERT and mBERT models showed very close F1 scores, and both models performed better than XLNET.

Ensemble deep learning is often used to improve hate detection accuracy (Zimmerman et al. 2018). When multiple trained weak learners are available and they are complementary to each other, there is scope for improving the performance by combining those learners. The most common ensemble techniques are majority voting, model averaging, and stacking (Karim et al. 2021; Roy et al. 2022). Roy et al. (2018) used an ensemble architecture for aggressive language identification, where convolutional neural networks and support vector machines are combined using a softmax classifier. They tested this model on the English and Hindi datasets. Very recently, an ensemble model combining three deep learning models for hate speech detection in Dravidian languages has been presented in Roy et al. (2022). In this work, authors have considered multiple variants of BERT models and combined them with the deep learning models—CNN, and/or DNN for developing multiple ensemble deep learning models.

Figure 5 presents a workflow of how deep ensemble approaches are used in hate speech classification. Table 4 presents most of the studies that have considered deep learning-based approaches to hate speech detection in Indian languages. It portrays the approaches taken by individual

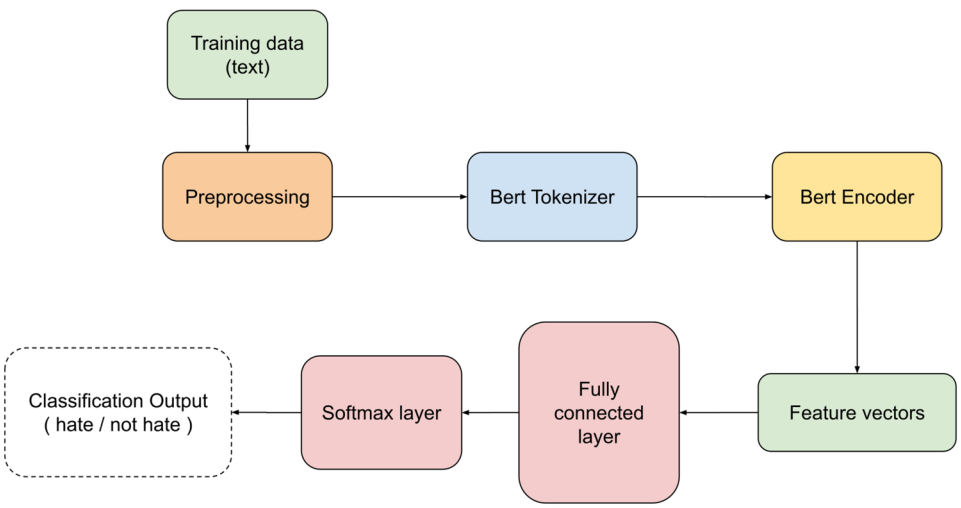

Fig. 4 General framework for all approaches where BERT is fine-tuned and used as classifier

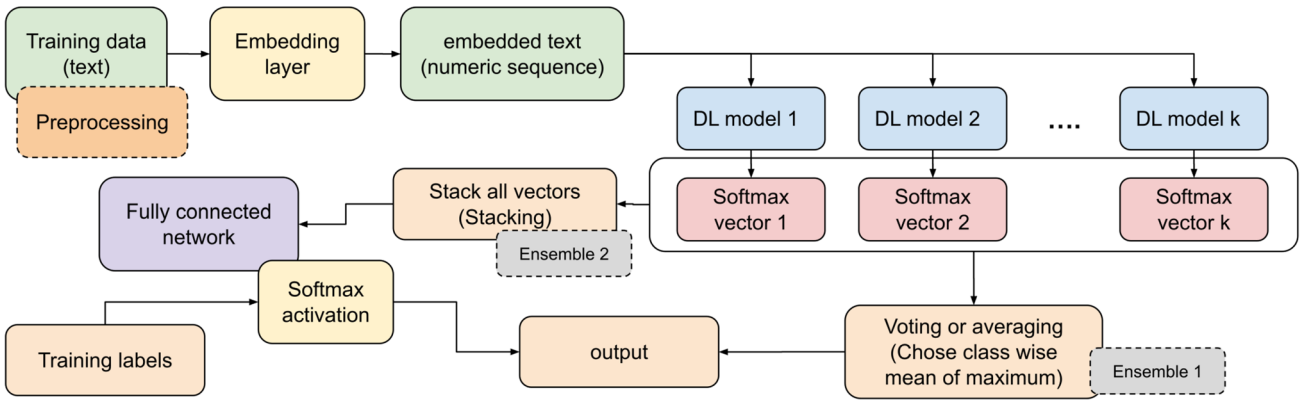

Fig. 5 General framework for all deep ensemble approaches used for hate speech detection

studies, the language of their dataset, and the embedding methods used.

6 Dataset annotation

Annotation is the primary process in the development of hate speech datasets. The texts that are submitted to human annotators must be sorted into which class they belong to complete the Annotation task. The annotation procedure can be carried out in numerous ways. There is no universally accepted best practice.

Some researchers use a limited number of professionals (Guest et al. 2021) or non-experts (Mandl et al. 2019) while others rely on crowd workers (Pavlopoulos et al. 2021). Since the labeling process is highly subjective, annotating data for hate speech detection is a highly challenging task because systematic bias occurs because of varying degrees of knowledge about societal concerns or even language variations (Sap et al. 2019). Bias can also result from demographic characteristics (Al Kuwatly et al. 2020). Users of data collection occasionally might think that certain tweets have been incorrectly labeled.

Since opinions about particular tweets vary, multiple people need to work on the annotation process. The common way for testing annotation quality is that some things are annotated at least twice, and metrics for inter-judge agreement are used to measure the agreement. However, when there is low agreement, it is difficult to say whether this is because the annotators do not have a common understanding or because there are a lot of questionable examples in the collection. Prior to beginning the annotation, it is unclear what amount of questionable cases are present in the data. Therefore, the annotation's quality cannot be assured, not even by the inter-judge agreement.

Our survey on Indian language hate speech detection reveals that a limited number of people with varying degrees of knowledge were employed for hate speech data annotation tasks. We found that most researchers collected data from Facebook, Twitter, YouTube, and other social media. For collecting tweets or comments, a predefined set of keywords was used. For example, the phrases "Loksabha election", "Loksabha election 2019 of India" were used for collecting election-related tweets from Twitter.

We noticed that two primary schemes were used for labeling data. The first is a binary scheme, which uses two values—usually yes or no—to indicate whether a particular phenomenon is present or absent. For example, the hate class is referred to as yes class, and the not-hate class is referred to as "no". This is also termed as "coarse-grained" classification. The second annotation scheme is the non-binary scheme, where more than two labels are used to label the data. This includes different shades for a given phenomenon, such as overtly aggressive, covertly aggressive, and not aggressive (Bhattacharya et al. 2020).

Recently, a few contests have offered datasets from India in several languages establishing significant benchmarks and resources for these languages. Among these, the notable shared tasks are the HASOC shared tasks, the TRAC shared task and a shared task on Dravidian languages organized in conjunction with the Dravidian LangTech workshop 2021. The HASOC shared task is conducted yearly starting from 2019 and the TRAC shared task was conducted in 2018 and 2020.

Together with the TRAC workshop, two iterations of the TRAC shared task on aggression identification were conducted. In TRAC 2018 (Kumar et al. 2018) at COLING, participants were provided with training and test sets containing Facebook comments, and another test set containing tweets in Hindi and English. The task was to classify posts as aggressive, covertly aggressive, and non-aggressive. Participants in TRAC 2020 (Kumar et al. 2020) at LREC received datasets with YouTube comments in Bengali, English, and Hindi. There were two subtasks: subtask B had two classes,

Table 4 Deep learning-based approaches used for hate speech detection in Indian languages

Title	Year/author	Publisher	Language focus	Deep learning model	Feature extraction/Embedding
Fighting hate speech from bilingual Hinglish speaker's perspective, a transformer-and translation-based approach	Biradar and Saumya (2022)	Springer	Bilingual Hinglish	Deep ANN	Transformer-based Interpreter and Feature extraction model on Deep Neural Network (TIF-DNN)
Detecting Offensive Tweets in Hindi–English Code-Switched Language	Mathur et al. (2018)	ACL	Hindi–English	Pre-trained CNN with initial layers freezed and last layers tuned	Transliteration of Hinglish tweets into English, then word2vec
Offensive language detection in Tamil YouTube comments by adapters and cross-domain knowledge transfer	Subramanian et al. (2022)	Elsevier	Tamil	Adding some extra connected layers to the freezed pre-trained encoders	Pre-trained mBERT, MuRIL (Base and Large), and XLM-RoBERTa (Base and Large)
L3Cube-MahaHate: A Tweet-based Marathi Hate Speech Detection Dataset and BERT models	Patil et al. (2022)	arxiv	Marathi	Monolingual and Multilingual BERT (mBERT)	BERT tokenizer and embeddings
Does aggression lead to hate? Detecting and reasoning offensive traits in Hinglish code-mixed texts	Sengupta et al. (2022)	Elsevier	Hindi–English Code mixed	Custom transformer-based network trained from scratch	Word piece tokenizer
Ceasing hate with MoH: Hate Speech Detection in Hindi–English code-switched language	Sharma et al. (2022)	Elsevier	Hindi–English Code mixed	Fined tuned Multilingual BERT, and MuRIL	BERT tokenizer and embeddings
Transformer-based approach for detection of abusive comment for Tamil language	Bharathi and Varsha (2022)	ACL	Tamil–English code mixed	Pre-trained BERT, XLNet and mBERT	BERT tokenizer and embeddings
Hate speech and offensive language detection in Dravidian languages using deep ensemble framework	Roy et al. (2022)	Elsevier	Dravidian languages	Ensemble and voting	Several deep models including BERT, LSTM, CNN
HateCheckHIn: Evaluating Hindi Hate Speech Detection Models	Das et al. (2022)	ACL	Hindi	Fined tuned Multilingual BERT	BERT tokenizer and embeddings
Aggression and Misogyny Detection using BERT: A Multi-Task Approach	Samghabadi et al. (2020)	ACL	English, Hindi, Bengali	Multilingual BERT base model	BERT base multilingual tokenizer
A Comparative Study of Different State-of-the-Art Hate Speech Detection Methods for Hindi–English Code-Mixed Data	Rani et al. (2020)	ACL	Hindi–English Code mixed	Character level 1D CNN	Character level encoding—one hot and char2vec
Hate Speech Detection from Code-mixed Hindi–English Tweets Using Deep Learning Models	Kamble and Joshi (2018)	arxiv	Hindi–English code mixed	1D CNN, LSTM, BiLSTM	Domain-specific word2vec from gensim
CMHE-AN: Code-mixed hybrid embedding-based attention network for aggression identification in Hindi–English code-mixed text	Mundra and Mittal (2023)	Springer	Hindi–English code mixed	BiLSTM + Attention	Hybrid embedding: a combination of word embedding and character embedding

Table 4 (continued)

Title	Year/author	Publisher	Language focus	Deep learning model	Feature extraction/Embedding
FA-Net: fused attention-based network for Hindi English code-mixed offensive text classification	Mundra and Mittal (2022)	Springer	Hindi–English code mixed	Fusion of BiLSTM and 1D CNN via the attention mechanism	Hybrid embedding-based feature representation obtained by combining word embedding and character embedding
Aggressive and Offensive Language Identification in Hindi, Bangla, and English: A Comparative Study	Kumar et al. (2021)	springer	Hindi Bangla, English	BERT and other multilingual transformers	BERT tokenizer and embeddings
Evaluation of Deep Learning Models for Hostility Detection in Hindi Text	Joshi et al. (2021)	IEEE	Hindi	BERT, CNN, LSTM, mBERT, IndicBERT	LSTM, BERT-embeddings used
DeepHateExplainer: Explainable Hate Speech Detection in Under-resourced Bengali Language	Karim et al. (2021)	IEEE	Bengali	ensemble voting on bert results	Feature representation using multiple pre-trained BERT models
Predicting the type and target of offensive social media posts in Marathi	Zampieri et al. (2022)	Springer	Marathi	SVC, CNN, BiLSTM, mBERT, XLM-R, and IndicBERT	Word unigram features for SVC, Word Embedding features for CNN and LSTM, BERT-embeddings for mBERT, and IndicBERT

one of which sought to identify gendered violence in posts directed against women, while subtask A had three classes from TRAC 2018.

The most well-known set of contests involving Indian languages is the HASOC shared task, which stands for "hate speech and offensive content identification" in Indo-European Languages (Mandl et al. 2020, 2019). It was started at the Forum for Information Retrieval (FIRE) in 2019. While datasets in English, German, and Hindi were available to participants in HASOC 2019, datasets in Tamil and Malayalam were also included in HASOC 2020. HASOC events are in progress and other languages like Marathi will probably be added in the subsequent years. Each HASOC event defined three tasks, a coarse-grained binary classification task, and two fine-rained (multi-class) classification tasks. For example, In HASOC 2019, there were three subtasks—(1) subtask1 was to classify hate speech (HOF) and non-offensive content, (2) subtask3 was to identify the type of hate(Hate, Offensive, and Profane) if the post is HOF, and (3) subtask3 was to decide the target of the post. Datasets were tagged by the organizers before distribution to the participants.

The shared task at Dravidian LangTech (Chakravarthi et al. 2021) used the code-mixed dataset of comments and posts in three Dravidian Languages, namely "Tamil–English", "Malayalam–English", and "Kannada–English" collected from social media. The task was to identify offensive languages in these data.

The more significant concern about the hate speech data annotation is that there were cases of erroneous annotation. For example, the participants raised this issue for the annotation of the TRAC datasets. In this case, for the subjective phenomenon "aggression", different annotators judged the same comment differently and some of the annotations did appear quite questionable. Therefore, they require additional investigation and validation.

7 Performance metrics

Detecting hate speech is a classification problem, and the metrics that are used to measure the performance of the approaches put forward by all the studies are generic classification metrics. Some studies have treated the problem as a binary classification (Islam et al. 2022; Jha et al. 2020) and some have treated this as a multi-class classification (Kumaresan et al. 2021; Patil et al. 2022) problem. We have found that accuracy and F1 score are the two mostly used metrics. In some studies along with accuracy and F1 score, authors have also used precision and recall to quantify their performance. The mathematical definitions of these metrics are provided below.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$precision = \frac{TP}{TP + FP} \tag{2}$$

$$recall = \frac{TP}{TP + FN} \tag{3}$$

$$f1\ score = \frac{2 \times precision \times recall}{precision + recall} \tag{4}$$

In the equations of the metrics given above, the notations TP, TN, FP, and FN denote the number of true positives, true negatives, false positives, and false negatives, respectively.

Other than the above-mentioned metrics, some researchers have introduced more fine-grained evaluation measures. Das et al. (2022) presented an evaluation framework that evaluates the BERT-based hate speech model using multiple functionalities. The main drawback of this method is that it requires a lot of human effort in preparing ground truth. However, we observe that the most common metrics used by the researchers are accuracy, precision, recall, and F-measure. For the multi-class datasets (> 2 classes), the macro average and the weighted average F-measures are commonly used.

8 Performance comparison of existing works

In this section, we compare the works presented in the papers reviewed by us. We observe that the data used for training and testing the various approaches differ largely. There are datasets of different languages, different scripts, different sizes, different objectives, and different content. And due to this fact, the performances of the existing methods for hate speech detection in Indian languages cannot be compared to each other fairly. The value of accuracy and F1 is not enough to justify the performance of a model. Hence, we should refrain from comparing their performances directly.

However, we observe that the earliest studies used feature extractors like n-grams (unigram, bigrams, trigrams, etc.) (Eshan and Hasan 2017; Akhter et al. 2018) and shallow machine learning models for the hate speech classification task. Some recent works used basic embeddings and deep learning models like CNN (Convolutional Neural Networks) (Mathur et al. 2018; Kamble and Joshi 2018) and LSTM (Long Short Term Memory). When better embeddings like fastText (Sreelakshmi et al. 2020) came into the picture, and the hate speech detection performance improved. Deep learning models started to dominate when advanced language models like BERT were used generously (Samghabadi et al. 2020). The transformer-based BERT model was a revolutionary approach, and it was pre-trained on an enormous amount of data and made available for public use. But in the initial days, the BERT was trained mainly in popular Western languages. They were usually used as feature extractors though they could be fine-tuned with the domain-specific data. In recent studies on hate speech detection in Indian languages (Sharma et al. 2022; Das et al. 2022; Patil et al. 2022), the multilingual BERT models have been used, and the multilingual BERT-based models have shown outperforming the traditional word embedding-based deep models. Multilingual BERT like mBERT is a transformer-based model which is trained in more than 100 different languages including some Indian languages. Comparing the studies over the years, we observe that the availability of a large volume of data, and computation power have made it possible to develop better pre-trained models that enabled the researchers to design better hate speech detection models in Indian languages. In Table 5, language-wise hate speech detection studies have been presented along with the best metrics achieved by the models developed by the researchers from time to time, we have also shown in the table, the dataset sizes used by the researchers. This is to provide adequate reference to the readers and refrain from directly comparing them. As we can see from this table, the datasets used by most researchers have sizes of less than 10k. For a few cases where the sizes of the datasets are a little bit larger (>30K). However transformer-based models like BERT and its variants performed the best for coarse-grained classification (hate or non-hate classes) tasks. Several researchers have combined BERT with LSTM and/or CNN to achieve better performance. Such deep learning models have shown above 90

9 Criticism, challenges, and suggestions

Our survey reveals that hate speech detection in Indian languages is still at the nascent stage even though many researchers have recently applied transformer-based language models for hate speech representation and detection. Particularly, most hate speech datasets in Indian languages are closed and not publicly available for comparing the existing results with the results obtained by the newly developed systems. This has created an obstacle to developing, testing and benchmarking the Indian language hate speech detection systems. For the public datasets, we find that the size is a big problem. We can see by analyzing the existing works that insufficient data has been used for training the models. In some cases, an enormous volume of texts was scraped, but after filtering out the unusable ones, only a few thousand remained for use in system development. To deal with the

Table 5 Comparison of the important research papers on hate speech detection in Indian languages, in terms of methods used, nature of datasets used, the language used, challenges faced, and the best results reported in the papers

Language	Author/year	Approach	Dataset description	Challenges faced	Best results
Bengali	Akhter et al. (2018)	Trigram language model, Linguistic and user information features. Models used: SVM, DT	2400 social media comments containing 10% bullying texts	Two classes were considered: bullying and not-bullying. No language-specific issues were addressed. This paper makes a general statement regarding the impact of linguistic differences and socio-emotional behavior of the users	Acc: 97%
	Ishmam and Sharmin (2019)	Features: n-gram, text quality, indicator count, word count, frequency of subwords. The model performed best: GRU	5126 comments (90% training, 10% testing)	Most existing works used two or three classes where this work considered Six classes: hate speech, communal attack, inciteful, religious hatred, political comments, and religious comments. Language-specific resources and knowledge were only applied at the preprocessing step for stemming and stop word removal. The possible reason for obtaining low accuracy is that it uses a large number of classes	Acc: 70%
	Islam et al. (2022)	Count vectorizer and TF-IDF features are used. ML algorithms used: LR, MNB, SVM, KNN, RF. Best model: RF with TF-IDF features	3006 comments are used in the experiments. Train-test split is not mentioned in the paper	Except preprocessing and data cleaning, no other step included any language-specific resources or knowledge. Two classes were considered: abusive and non-abusive. Data was mostly collected from Facebook. Dataset used in the experiments is imbalanced with distribution: non-abusive- 56.25% and abusive−43.75%	Acc: 67%

Table 5 (continued)

Language	Author/year	Approach	Dataset description	Challenges faced	Best results
	Remon et al. (2022)	Fast Text embedding features. Models used: CNN, SVM. Best model: SVM	10,133 Bengali comments	Data is collected from Facebook comment sections. Dataset is imbalanced. 3,737 hate and 6,396 non-hate speech. Two classes are considered: hate and non-hate. Language-specific resources like a stop word list and tools like a stemmer were used for preprocessing only	Acc: 87%
	Eshan and Hasan (2017)	Features: unigrams, bigrams, trigrams. Models used: LR, RF, MNB, and SVM (linear, RBF, polynomial, Sigmoid). Best model: SVM linear, but MNB performs closely to SVM also	Experiments are conducted with varying sizes of datasets: 500, 1000, 1500, 2000, and 2500 comments. 10-fold cross-validation is used	Bengali abusive comments are collected from Facebook. To ensure privacy, only public comments are collected. Since comments are noisy, comments are cleaned. Other than data cleaning, no language-specific features are used	Acc: 81% (approx) on 2500 comments. This result is estimated from the bar graph given in the paper. No table of results is given in the paper
	Karim et al. (2021)	Character n-gram and word unigrams features were used for the various ML models- LR, SVM, KNN, RF, and GBT, and these ML models were compared with various deep learning models like CNN, BiLSTM, Conv-LSTM and BERT variants. A voting-based ensemble of BERT models performed the best	The Bengali dataset consists of 8087 texts labeled with 4 categories with distribution, political(999), Religious(1211), Geopolitical(2364), and personal (3513). 80-20 split was used for creating training and testing sets	It deals with a 4-class classification problem. Language-specific operations are used at the preprocessing step for stemming. Along with the classification output, this work measures the explainability of the models	F1 score: 0.91 (not mentioned if Weighted or macro)
	Sarker et al. (2022)	Unigrams, bigrams, trigrams features were used for the ML Models- LR, RF, MNB, and SVM. The traditional ML models were compared to the GRU model with Word2Vec embedding. Best model: MNB. GRU performed close to MNB	2000 comments, balanced between two classes. 90-10 split was used for creating the training set and the test set	It deals with a 2-class problem. Two classes were anti-social and socially accepted. A language-specific stop word list was used at the preprocessing step	Acc: 80.51% (MNB), F1: 0.8766 (GRU) ((not mentioned if Weighted or macro))

Table 5 (continued)

Language	Author/year	Approach	Dataset description	Challenges faced	Best results
Hindi- English code- mixed	Sreelakshmi et al. (2020)	Features: fastText embedding, Word2Vec embedding and Doc2Vec embedding. Models used: SVM with RBF kernel, and RF. The best model: SVM-RBF with fastText features	10,000 sample texts were equally divided into hate and non-hate	This work deals with a binary hate speech detection task which is a common task in the domain. Since word embedding was used for feature extraction, light preprocessing operations like text cleaning were applied	Acc: 85.81%, F1: 0.8580 (not mentioned if Weighted or macro)
	Rani et al. (2020)	Features: Frequency-based term weighting. Models used: traditional ML algorithms—SVM, MNB, KNN, DT, and deep learning algorithm: character-based CNN. The best model: character-based CNN model which performed equally well on all three datasets	Three datasets were used. Dataset-1 contains 4579 posts/tweets(hate:2290, non-hate:2289), Dataset-2 (HASOC dataset) contains 4665 post/tweets (hate: 2419 non-hate: 2246), Datset-3 (author's dataset) contains 3367 posts/tweets(hate:478, non-hate: 2889). 70-10-20 split for each dataset was done for creating train, validation, and test sets	This work deals with a binary classification problem which is common in the hate speech detection domain. Since character CNN is used, no preprocessing was done. The models were tested on three different datasets to prove generalization capability. Authors reported that the data annotation process was not easy while developing their datasets, particularly, inter-annotator agreement was poor without having specific annotation guidelines	Acc: 86% (average over three datasets), micro F1: 0.74 (average over three datasets)
	Mathur et al. (2018)	Feature representation: word embedding. Models used: Ternary trans-CNN which was pre-trained on tweets in English and then the same model was again retrained after adding only two trainable layers and keeping other layers freezed	Two datasets are used. Dataset-1 contains a total of 14,509 tweets with class distribution- Non-Offensive: 7274, Abusive: 4836, Hate-inducing: 2399. Dataset-2(HEOT) contains a total of 3679 with class distribution- Non-Offensive:1414, Abusive: 1942, Hate-inducing: 323	This work deals with a 3-class hate speech detection in Hindi–English code-mixed data. The Preprocessing step involves common data cleaning operations with the exception that Hinglish words are translated into corresponding English words. Gensim stop word list is used to remove less important words	Acc: 83.90% (for HEOT dataset), Macro F1: 0.714 (for HEOT dataset)

Table 5 (continued)

Language	Author/year	Approach	Dataset description	Challenges faced	Best results
	Sengupta et al. (2022)	Custom-created two-layer transformer model trained with Hinglish dataset was implemented for each task. The texts were tokenized into a series of subwords using a custom WordPiece tokenizer. To create a position-aware representation, the positional encoding and the original subword embeddings were added together. Global average pooling was used to obtain a single vector representation for each text. The statistical features were added to each text representation	The models were tested on multiple datasets: the Aggression dataset contains 30,000 texts with class distribution- covertly aggressive,- 38.08%, overtly aggressive,—31.64%, non-aggressive,—30.32%, the Hate dataset contains 4,578 texts with class distribution- YES—36.26%, NO—63.74%, the Humor dataset contains 2,952 texts with class distribution- YES—59.59%, NO—40.41%, the Sarcasm dataset contains 5,250 texts with class distribution: NO—90.40%, YES—9.60%, and the Stance dataset contains 3,545 texts with class distribution: Favor—27.19%, Against- 18.25%, None—54.56%	This work is different from the traditional transformer-based model in various ways: custom transformer was developed from scratch using Highlish data (Hindi–English code mixed). To handle the labeled data scarcity problem, it used a pseudo-labeling-based framework to transfer knowledge among models	The custom transformer model with statistical features achieved a macro F1 score of 0.927
	Kamble and Joshi (2018)	Three deep learning models, 1D CNN, LSTM, and BiLSTM were used. For these models, Domain-specific word embeddings were created by applying the Gensim Word2Vec model on a large corpus of tweets. Deep learning models were compared with SVM and RF with features such as Character N-Grams, Word N-Grams, Punctuations, Lexicon, and Negations. Best model: 1D CNN	3849 tweets, out of which 1436 are hateful	This work deals with a binary hate speech detection problem. Data size is limited. Domain-specific word embedding was used	Acc: 82.62%, F1: 0.8085 (not mentioned if weighted or macro)

Table 5 (continued)

Language	Author/year	Approach	Dataset description	Challenges faced	Best results
	Mundra and Mittal (2023)	Hybrid embedding: a combination of word embedding and character embedding was used for feature representation and the attention-based BiLSTM model was used. The proposed model performed better than LR, XGBoost, CharCNN, WordCNN and FineTunedBert	The TRAC-2-2020 dataset was used- the training set consists of 3984 YouTube comments (Covertly Aggressive(CAG): 829, Non-Aggressive(NAG): 2245, Overtly Aggressive (OAG): 910), and the test dataset consists of 1200 YouTube comments (Covertly Aggressive(CAG):191, Non-Aggressive(NAG):325, Overtly Aggressive (OAG): 684)	This work deals with multi-class hate speech detection in Hindi–English code-mixed data. Code-mixed hybrid embedding was proposed. The attention mechanism was used	Acc: 75.23%, Weighted F1: 73.34 (not mentioned if weighted or macro)
	Mundra and Mittal (2022)	Hybrid embedding-based feature representation was obtained by combining word embedding and character embedding and the same feature representation was fed to BiLSTM and !D CNN model simultaneously and then their outputs were fused via attention mechanism for better high-level feature extraction. The proposed FA-Net model performed better than LR, XGBoost, CharCNN, WordCNN, and FineTunedBert	TRAC-2-2020 dataset was used. The training set consists of 3984 YouTube comments(Covertly Aggressive(CAG): 829, Non-Aggressive(NAG): 2245, Overtly Aggressive (OAG): 910), and the test dataset consists of 1200 YouTube comments (Covertly Aggressive(CAG):191, Non-Aggressive(NAG):325, Overtly Aggressive (OAG): 684)	This work deals with multi-class hate speech detection in Hindi–English code-mixed data. Code-mixed hybrid embedding was proposed. The model fusion via an attention mechanism was used	ACC: 80.89%, Weighted F1: 81.35 (not mentioned if weighted or macro)
Hindi	Joshi et al. (2021)	CNN, LSTM, BERT, mBERT, IndicBERT models were used. For CNN and LSTM, the pre-trained Hindi fastText embeddings were used. For the coarse-grained task, Multi-CNN with IndicNLP FastText embedding achieved the best weighted F1 score, and for the fine-grained task, the IndicBert model obtained the best weighted F1 score	The dataset contains a total of 8192 posts with two broad categories, hostile and non-hostile. The non-hostile class contains a total of 4358 posts which were divided into train: 3050, validation: 435, and test:873. The non-hostile class was further sub-categorized into the fine-grained classes- Fake, Hate, Offense, and Defame	No preprocessing was used since automatic feature representation was used. The fine-grained classification problem was shown more challenging than the binary classification problem. The fine-grained classification achieved a low F1 score	weighted F1: 0.9667 (coarse-grained), 0.6129 (fine-grained)

Table 5 (continued)

Language	Author/year	Approach	Dataset description	Challenges faced	Best results
Hindi, Bengali	Samghabadi et al. (2020)	A Multilingual BERT base model with an attention layer on its top was used for implementing multi-task learning. This model was evaluated using the datasets released for two subtasks: Subtask A involves the classification of texts into one of three aggression classes—not aggressive, covertly aggressive, and overtly aggressive, and subtask B involves the classification of texts into two misogyny classes—Gendered and Non-Gendered	The Hindi dataset contains 6181 texts which are split into the training set of 3984, the development set of 997 and the test set of 1200. The Bengali dataset contains a total of 5972 posts which are split into the training set of 3826, the development set of 957, and the test set of 1188	Two subtasks were defined, one for aggression classes and another for misogyny classes. These tasks are more difficult than merely classifying texts as hate or non-hate. For subtask A, the performance score showed a low value because this task was more fine-grained and the class overlapping was high	For subtask A, weighted F1: 0.7183 (Hindi), and 0.7369(Bengali). For subtask B, weighted F1: 0.8008(Hindi) and 0.9206(Bengali)
	Kumar et al. (2021)	SVM, BERT and BERT variants—ALBERT and DistilBERT were compared according to their performance on Aggressive and Offensive language identification	The Models were tested on multiple datasets- HASOC and TRAC-2 datasets. For the Hindi HASOC2019 dataset, the training set contains 4665 tagged tweets (hate-2469 and non-hate: 2196) and the test set contains 1318 tweets (hate: 605 and non-hate: 713). The TRAC-2 dataset consists of Hindi and Bengali datasets. The Hindi dataset contains 6181 posts (training:3984, development: 997, and the test: 1200), and the Bengali dataset contains a total of 5972 posts(training:3826, development:957, and test set: 1188)	TRAC-2 dataset is more challenging because TRAC-2 defined two subtasks: one for aggression classes and another for misogyny classes. These tasks are more difficult than merely classifying texts as hate or non-hate. subtask A was more challenging because this task is more fine-grained and the class overlapping is high	F1 score: 0.80 (not mentioned if weighted or macro)

Table 5 (continued)

Language	Author/year	Approach	Dataset description	Challenges faced	Best results
Tamil	Subramanian et al. (2022)	Traditional ML algorithms with TF-IDF features were used. The used ML models are Bernoulli Naïve Bayes, SVM, LR, and KNN. Pre-trained multilingual transformer models that were used in this work are mBERT, MuRIL (Base and Large), and adapter XLM-RoBERTa(Base and Large). The adapters are the additional layers added to the main pre-trained model while freezing the weights of the pre-trained models. The purpose of adapters is the same as that of fine-tuners. The best model suggested by the authors: is Adapter XLM-RoBERTa(Large)	The dataset contains 5877 Tamil texts for training(offensive:1153, not offensive:4724) and 654 Tamil texts for testing	This work deals with a binary classification problem. The preprocessing step that demands some domain-specific knowledge involves the removal of non-Tamil punctuation, characters, words, etc.	Acc: 88.532%
Marathi	Patil et al. (2022)	Deep learning models- CNN+fastText, LSTM+fastText, and monolingual and Multilingual BERTs, namely MahaBERT, IndicBERT, mBERT, and xlm-RoBERTa were used. MahaBERT (Monolingual Marathi BERT) achieved the best results	A total of 25,000 tweets which were split into three parts -Train:21500 tweets(Hate-5375, Offensive-5375, Profan-5375, Not-5375), Test: 2000 tweets(Hate-500, Offensive-500, Profan-500, Not-500), and Validation: 1500 tweets(Hate-375, Offensive-375, Profan-375, Not-375)	This work deals with 2-class(hate and not-hate) and 4-class (Hate, Offensive, Profan, Not) problems. Class distribution in each set is balanced. The traditional monolingual and multilingual BERT variants were used. Monolingual Marathi corpus annotation was done for this work	Acc: 90.9% (for 2-class problem), ACC: 80.3% (for 4-class problem)
	Zampieri et al. (2022)	The authors introduced a Marathi Offensive Language Dataset called MOLD 2.0. They reported the baseline results on the MOLD 2.0 dataset through experimentation using SVC (Support Vector Classifier), CNN, BiLSTM, mBERT, XLM-R, and IndicBERT. The IndicBERT model achieved the best performance on the Marathi offensive language identification dataset	Total dataset has 3,611 tweets. The training set consists of 3101 tweets (2034 are not offensive and the remaining are offensive). The test set has 510 tweets (250 tweets are not offensive and the remaining are offensive)	The main focus of this work is to introduce a Marathi dataset called MOLD 2.0 and report baseline results on this dataset	Macro F1: 0.85

Table 5 (continued)

Language	Author/year	Approach	Dataset description	Challenges faced	Best results
Tamil–English code mixed	Bharathi and Varsha (2022)	Pre-trained models—BERT, XLNet and mBERT were used. The Best model is the bert-base-uncased model for both datasets	3500 (Tamil comments with distribution: Train-2240, Development-560, Test-700) and 9295 code-mixed YouTube comments with distribution: Train—5948, Development-1488, Test-1859	This work uses a multi-class dataset with fine-grained classes: Misogyny, Misandry, Homophobia, Transphobia, Xenophobia, Counter Speech, Hope Speech. Some Tamil language-specific operations were only applied at the preprocessing step	weighted F1: 0.96 (code-mixed) and 0.59(Tamil text)
Malayalam and Tamil–English code mixed	Roy et al. (2022)	Ensemble and voting on several deep models including BERT, LSTM, CNN were used	Malayalam dataset- Train: 1953(offensive), 2047(non-offensive), validation set: 478(offensive), 473(on-offensive). Tamil dataset- train: 2020(offensive), 1980 (not offensive), Validation dataset- 475(offensive), 465(not offensive)	This work deals with two class problem	F1: 0.802 for Malayalam and 0.933 for Tamil code mixed (not mentioned if weighted or macro)

data scarcity problem, some studies have used transfer learning (Biradar and Saumya 2022).

We also observe that most existing Indian language hate speech detection systems have been developed based on the methods applied for hate speech detection in English. The linguistic knowledge or semantics of the Indian languages is seldom used by researchers in designing language-specific features effective for Indian language hate speech detection.

Though, over the last several years, many research papers have been published on hate speech detection in Indian languages, there are a lot of challenges that are still major obstacles. Text or speech itself is a very abstract entity and it is very difficult to represent them to make them suitable for processing by any shallow or deep learning models. Hate speech detection in a particular language does not simply boil down to the detection of certain abusive keywords or phrases. Natural languages have very complex semantics and they vary from one language to another language. There can be very offensive text without a single abusive word in it. For example, in every language, we have proverbs whose meanings do not depend on the individual literal meanings of the words in them, rather their meanings are determined based on the situations or the contexts they are used. Moreover, users of social media platforms constantly modify their way of expressing things—using symbols, acronyms, other unrelated words, emojis, etc., which makes hate speech detection a challenging task. So, the detection algorithms have to constantly keep up with the trending vocabulary.

Identifying a text as offensive also depends a lot on external factors other than the text content. The sensitiveness of the context where the text is posted, to whom it has been directed and the sentiment or tonality of the users also impact the detection. For example, a comment on a celebrity tweet happens to be much more sensitive and has to be handled more delicately. Sarcasm makes the problem more difficult because words with certain meanings when said in different tones can mean opposite things.

Although social media is the best place for collecting training data, these data have to be labeled very carefully and manually because the type of data is highly noisy. Therefore, data annotation for hate speech detection is a laborious and time-consuming task. Perfection of the algorithms depends a lot on the quality of labeling. Since most Indian language hate speech datasets are not publicly available, this forces the new researchers to develop a new dataset from scratch. Thus, various datasets are reported in the literature along with the results obtained on these datasets. However, this creates another important obstacle to research on hate speech detection in Indian languages because different researchers used different evaluation metrics. The most common evaluation metrics are accuracy, precision, recall, and F-measure. Accuracy is not the appropriate measure when the dataset is imbalanced. For better evaluation, the F-measure can be of

three types: Macro F1 score, Micro F1 score, and weighted F-measure. We observed during this study that different researchers have used different evaluation metrics. For the cases, where datasets are not public and the type of F1 score is not mentioned in the paper (only F1 score is mentioned), the new researchers will have to face difficulty in comparing their research outcomes with the existing ones.

In the earlier part of this section, we have highlighted some limitations and obstacles to the research on hate speech detection in Indian languages. The first and the important obstacles are the lack of annotated data and the most datasets are not publicly available. To mitigate this limitation, we should force the authors to make datasets online before the publication of their research papers. The crowd workers might be employed to annotate more data. Another approach to mitigating this limitation can be using semi-supervised learning to label a large amount of unlabeled data and scrutinizing manually the data labeled by the semi-supervised learning model with higher confidence. Although the data augmentation approach is a common approach used in the image analysis task, we can think about how this idea can be applied to text example generation (Thomson et al. 2023).

Data imbalance problem is also a crucial problem for hate speech detection because hate speech texts naturally follow a skewed distribution when these are generated on online platforms. We have already discussed in this survey that many researchers used pre-trained models or a combination of pre-trained models to deal with this problem. However, minority oversampling techniques (Chawla et al. 2002) and data augmentation techniques (Thomson et al. 2023) can be used.

The vocabulary of the hate speech texts is substantially different from the traditional natural language vocabulary and it constantly changes its size as users add very uncommon words, symbols, emojis, etc. To mitigate this limitation, we need an alternative approach that can automatically populate the hate speech terms and add to the vocabulary.

Since, hate speech semantics are very difficult to model without any reference to a specific domain or application, an open domain hate speech detection task is very difficult to achieve. To deal with other issues such as sarcasm, we need to have deep semantic analysis which needs to combine the deep learning models with the knowledge-based approaches to morphology, lexical, and semantic analysis.

10 Conclusion and future works

The main purpose of this review work was to present and organize recent research works in hate speech detection for Indian languages. We have gone through research studies on Indian language hate speech detection published in the last five years. In our survey, multiple aspects of hate speech detection including datasets, preprocessing, hand-crafted feature engineering, embedding-based feature representation, and various machine learning, and deep learning models, have been thoroughly covered.

We have classified our survey into three main parts: a survey of Indian language hate speech detection datasets, various machine, and deep learning methods used by the researchers for hate speech detection in Indian languages, and a comparison of the language-wise results reported in the recently published research papers.

We observed that most researchers evaluated their work using their datasets which are not made public. They used methods for hate speech detection that include traditional machine learning and deep learning methods. Our survey reveals that language-specific linguistic or semantic features have not received much attention from the researchers. We also observe that the noisy social media texts and intermixing of multiple languages by social media users represent a significant challenge for hate speech detection in Indian languages.

Among the existing models, the BERT-based model or its variants have been reported by many researchers as a successful hate speech detection model for Indian languages. The possible reason for the success of the BERT-based model is the lack of resources for Indian languages because many Indian languages are still resource-poor languages.

We hope that the new researchers interested in doing research on hate speech detection in any Indian language will quickly get a comprehensive overview of the recent works in the field. Although our main focus was to review hate speech detection, there are some existing research works (Masud and Charaborty 2023) that attempted to assess the power dynamics between the ruling and opposing parties by correlating the reported online trends with actual events. The purpose of this study is to demonstrate how political discourse on social media is influenced by elections. This type of work is interesting, but it differs from hate speech detection. In this case, political attacks are classified as a specific type of offense, apart from identity-based attacks like hate speech. In this survey, we have not covered this type of work.

To improve hate speech detection, we need to investigate the following issues in the future. These issues are broadly related to (1) lack of sufficient annotated data, (2) data imbalance problem, (3)code-mixed and multilingual texts, (4) constantly changing the vocabulary words, and (5) short and highly noisy data, and (5) assessing the difficulty level of hate speech by the experts while annotating data and (6) combining the traditional knowledge-based morphological, lexical and semantic analysis with the deep learning models.

Acknowledgements No funding was received for this work.

Author Contributions Kamal Sarkar was involved in conceptualization. Arpan Nandi, Arjun Mallick, Arkadeep De, and Kamal Sarkar

contributed to methodology. Software: This is a survey paper. So, it is not applicable. Validation: This is a survey paper. So, this is not applicable. Kamal Sarkar, Arpan Nandi, Arjun Mallick, and Arkadeep De were involved in formal analysis. Kamal Sarkar, Arpan Nandi, Arjun Mallick, and Arkadeep De contributed to investigation. Arpan Nandi, Arjun Mallick, Arkadeep De, and Kamal Sarkar were involved in resources. Data Curation: This is a survey paper. So, this is not applicable. Arpan Nandi, Arjun Mallick, Arkadeep De, and Kamal Sarkar contributed to writing—original draft. Kamal Sarkar was involved in writing—review and editing. Arpan Nandi, Arjun Mallick, and Arkadeep De contributed to visualization. Kamal Sarkar was involved in supervision. Kamal Sarkar contributed to project administration.

Data availability This is a survey paper. We have reviewed the hate speech detection datasets used in Indian languages and the necessary links to the datasets are given in the paper.

Declarations

Conflict of interest The authors have no conflicts of interest.

Code availability This is a survey paper. So, it is not applicable.

Ethical approval During the research, no human participants or animals were involved.

Informed Consent: This article does not involve any studies with human participants or animals performed by any of the authors.

Editorial policies for: Springer journals and proceedings: https://www.springer.com/gp/editorial-policies Nature Portfolio journals: https://www.nature.com/nature-research/editorial-policies *Scientific Reports*: https://www.nature.com/srep/journal-policies/editorial-policies BMC journals: https://www.biomedcentral.com/getpublished/editorial-policies

References

Akhter S, et al (2018) Social media bullying detection using machine learning on Bangla text. In: 2018 10th International conference on electrical and computer engineering (ICECE). IEEE, pp 385–388

Alrehili A (2019) Automatic hate speech detection on social media: a brief survey. In: 2019 IEEE/ACS 16th international conference on computer systems and applications (AICCSA). IEEE, pp 1–6

Al Kuwatly H, Wich M, Groh G (2020) Identifying and measuring annotator bias based on annotators' demographic characteristics. In: Proceedings of the 4th Workshop on online abuse and harms, pp 184–190

Anusha M, Shashirekha H (2020) An ensemble model for hate speech and offensive content identification in Indo-European languages. In: FIRE (Working Notes), pp 253–259

Barnwal S, Kumar R, Pamula R (2022) IIT DHANBAD CODECHAMPS at SemEval-2022 task 5: MAMI—multimedia automatic misogyny identification. In: Proceedings of the 16th international workshop on semantic evaluation (SemEval-2022). Association for Computational Linguistics, Seattle, pp 733–735. https://doi.org/10.18653/v1/2022.semeval-1.101

Bharathi B, Varsha J (2022) Ssncse nlp@ tamilnlp-acl2022: transformer based approach for detection of abusive comment for Tamil language. In: Proceedings of the 2nd workshop on speech and language technologies for Dravidian languages, pp 158–164

Bhattacharya S, Singh S, Kumar R, Bansal A, Bhagat A, Dawer Y, Lahiri B, Ojha AK (2020) Developing a multilingual annotated corpus of misogyny and aggression. arXiv preprint arXiv:2003.07428

Biradar S, Saumya S et al (2022) Fighting hate speech from bilingual Hinglish speaker's perspective, a transformer-and translation-based approach. Soc Network Anal Min 12(1):1–10

Bohra A, Vijay D, Singh V, Akhtar SS, Shrivastava M (2018) A dataset of Hindi–English code-mixed social media text for hate speech detection. In: Proceedings of the 2nd workshop on computational modeling of people's opinions, personality, and emotions in social media. Association for Computational Linguistics, New Orleans, Louisiana, pp 36–41. https://doi.org/10.18653/v1/W18-1105

Chakravarthi BR (2022) Hope speech detection in Youtube comments. Soc Network Anal Min 12(1):1–19

Chakravarthi BR, Priyadharshini R, Muralidaran V, Jose N, Suryawanshi S, Sherly E, McCrae JP (2022) Dravidiancodemix: sentiment analysis and offensive language identification dataset for Dravidian languages in code-mixed text. Lang Resour Eval 56(3):765–806

Chakravarthi BR, Priyadharshini R, Jose N, Mandl T, Kumaresan PK, Ponnusamy R, Hariharan R, McCrae JP, Sherly E, et al (2021) Findings of the shared task on offensive language identification in Tamil, Malayalam, and Kannada. In: Proceedings of the 1st workshop on speech and language technologies for Dravidian languages, pp 133–145

Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357

Das M, Saha P, Mathew B, Mukherjee A (2022) Hatecheckhin: Evaluating Hindi hate speech detection models. arXiv preprint arXiv:2205.00328

Del Vigna12 F, Cimino23 A, Dell'Orletta F, Petrocchi M, Tesconi M (2017) Hate me, hate me not: hate speech detection on facebook. In: Proceedings of the 1st Italian conference on cybersecurity (ITASEC17), pp 86–95

Dhanya L, Balakrishnan K (2021) Hate speech detection in Asian languages: A Survey. In: 2021 International conference on communication, control and information sciences (ICCISc) 1:1–5 (IEEE)

Dowlagar S, Mamidi R (2021) A survey of recent neural network models on code-mixed Indian hate speech data. In: Forum for information retrieval evaluation, pp 67–74

Dutta S, Majumder U, Naskar SK (2021) sdutta at comma@ icon: a CNN-LSTM model for hate detection. In: Proceedings of the 18th international conference on natural language processing: shared task on multilingual gender biased and communal language identification, pp 53–57

Eshan SC, Hasan MS (2017) An application of machine learning to detect abusive bengali text. In: 2017 20th International conference of computer and information technology (ICCIT). IEEE, pp 1–6

Grave E, Bojanowski P, Gupta P, Joulin A, Mikolov T (2018a) Learning Word Vectors for 157 Languages. https://doi.org/10.48550/ARXIV.1802.06893

Grave E, Bojanowski P, Gupta P, Joulin A, Mikolov T (2018b) Learning word vectors for 157 languages. arXiv preprint arXiv:1802.06893

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, pp 1336–1350

Himabindu GSSN, Rao R, Sethia D (2022) A self-attention hybrid emoji prediction model for code-mixed language: (Hinglish). Social Network Anal Min 12(1):137

Ishmam AM, Sharmin S (2019) Hateful speech detection in public facebook pages for the Bengali language. In: 2019 18th IEEE international conference on machine learning and applications (ICMLA). IEEE, pp 555–560

Islam M, Hossain MS, Akhter N (2022) Hate speech detection using machine learning in Bengali languages. In: 2022 6th International conference on intelligent computing and control systems (ICICCS). IEEE, pp 1349–1354

Jemima PP, Majumder BR, Ghosh BK, Hoda F (2022) Hate speech detection using machine learning. In: 2022 7th international conference on communication and electronics systems (ICCES). IEEE, pp 1274–1277

Jha VK, Hrudya P, Vinu P, Vijayan V, Prabaharan P (2020) Dhot-repository and classification of offensive tweets in the Hindi language. Procedia Comput Sci 171:2324–2333

Joshi R, Karnavat R, Jirapure K, Joshi R (2021) Evaluation of deep learning models for hostility detection in Hindi text. In: 2021 6th International conference for convergence in technology (I2CT). IEEE, pp 1–5

Kamble S, Joshi A (2018) Hate speech detection from code-mixed Hindi–English tweets using deep learning models. arXiv preprint arXiv:1811.05145

Karim MR, Dey SK, Islam T, Sarker S, Menon MH, Hossain K, Hossain MA, Decker S (2021) Deephateexplainer: explainable hate speech detection in under-resourced Bengali language. In: 2021 IEEE 8th international conference on data science and advanced analytics (DSAA). IEEE, pp 1–10

Khan H, Phillips JL (2021) Language agnostic model: detecting islamophobic content on social media. In: Proceedings of the 2021 ACM southeast conference, pp 229–233

Kumar R, Lahiri B, Ojha AK (2021) Aggressive and offensive language identification in Hindi, Bangla, and English: a comparative study. SN Comput Sci 2(1):1–20

Kumar R, Reganti AN, Bhatia A, Maheshwari T (2018) Aggression-annotated corpus of Hindi–English code-mixed data. arXiv preprint arXiv:1803.09402

Kumar T, Mahrishi M, Sharma G (2023) Emotion recognition in Hindi text using multilingual Bert transformer. Multimed Tools Appl 1–22

Kumar R, Ojha AK, Malmasi S, Zampieri M (2018) Benchmarking aggression identification in social media. In: Proceedings of the 1st workshop on trolling, aggression and cyberbullying (TRAC-2018), pp 1–11

Kumar R, Ojha AK, Malmasi S, Zampieri M (2020) Evaluating aggression identification in social media. In: Proceedings of the 2nd workshop on trolling, aggression and cyberbullying, pp 1–5

Kumaresan PK, Sakuntharaj R, Thavareesan S, Navaneethakrishnan S, Madasamy AK, Chakravarthi BR, McCrae JP (2021) Findings of shared task on offensive language identification in Tamil and Malayalam. In: Forum for information retrieval evaluation, pp 16–18

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 annual meeting of the forum for information retrieval evaluation, pp 14–17

Mandl T, Modha S, Kumar MA, Chakravarthi BR (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, pp 29–32

Masud S, Charaborty T (2023) Political mud slandering and power dynamics during Indian assembly elections. Soc Network Anal Min 13(1):108

Mathew B, Illendula A, Saha P, Sarkar S, Goyal P, Mukherjee A (2020) Hate begets hate: a temporal study of hate speech. Proc ACM Hum–Comput Interaction 4(CSCW2):1–24

Mathur P, Shah R, Sawhney R, Mahata D (2018) Detecting offensive tweets in Hindi–English code-switched language. In: Proceedings of the 6th international workshop on natural language processing for social media, pp 18–26

Meetei LS, Singh TD, Borgohain SK, Bandyopadhyay S (2021) Low resource language specific pre-processing and features for sentiment analysis task. Lang Resour Eval 55(4):947–969

Mikolov T, Chen K, Corrado G, Dean, J (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781

Mridha MF, Wadud MAH, Hamid MA, Monowar MM, Abdullah-Al-Wadud M, Alamri A (2021) L-boost: identifying offensive texts from social media post in Bengali. IEEE Access 9:164681–164699

Mundra S, Mittal N (2022) Fa-net: fused attention-based network for Hindi English code-mixed offensive text classification. Soc Network Anal Min 12(1):100

Mundra S, Mittal N (2023) Cmhe-an: code mixed hybrid embedding based attention network for aggression identification in Hindi English code-mixed text. Multimed Tools Appl 82(8):11337–11364

Naseem U, Razzak I, Eklund PW (2021) A survey of pre-processing techniques to improve short-text quality: a case study on hate speech detection on twitter. Multimed Tools Appl 80(28):35239–35266

Pan SJ, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359

Patil H, Velankar A, Joshi R (2022) L3cube-mahahate: A tweet-based marathi hate speech detection dataset and bert models. In: Proceedings of the 3rd workshop on threat, aggression and cyberbullying (TRAC 2022), pp 1– 9

Pavlopoulos J, Sorensen J, Laugier L, Androutsopoulos I (2021) Semeval-2021 task 5: toxic spans detection. In: Proceedings of the 15th international workshop on semantic evaluation (SemEval-2021), pp 59–69

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

Rahman AI, Akhand Z-E, Noor MAU, Islam J, Mahtab M, Mehedi MHK, Rasel AA, et al (2022) Comparative analysis on joint modeling of emotion and abuse detection in Bangla language. In: International conference on advances in computing and data sciences. Springer, pp 199–209

Rani P, Suryawanshi S, Goswami K, Chakravarthi BR, Fransen T, McCrae JP (2020) A comparative study of different state-of-the-art hate speech detection methods in Hindi–English code-mixed data. In: Proceedings of the 2nd workshop on trolling, aggression and cyberbullying, pp 42–48

Remon NI, Tuli NH, Akash RD(2022) Bengali hate speech detection in public facebook pages. In: 2022 International conference on innovations in science, engineering and technology (ICISET). IEEE, pp 169–173

Roy PK, Bhawal S, Subalalitha CN (2022) Hate speech and offensive language detection in Dravidian languages using deep ensemble framework. Comput Speech Lang 75:101386

Roy A, Kapil P, Basak K, Ekbal A(2018) An ensemble approach for aggression identification in english and hindi text. In: Proceedings of the 1st workshop on trolling, aggression and cyberbullying (TRAC-2018), pp 66–73

Samghabadi NS, Patwa P, Pykl S, Mukherjee P, Das A, Solorio T(2020) Aggression and misogyny detection using bert: a multi-task approach. In: Proceedings of the 2nd workshop on trolling, aggression and cyberbullying, pp 126–131

Sap M, Card D, Gabriel S, Choi Y, Smith NA (2019) The risk of racial bias in hate speech detection. In: Proceedings of the 57th annual meeting of the association for computational linguistics, pp 1668–1678

Sarkar K (2018) Using character n-gram features and multinomial naïve bayes for sentiment polarity detection in Bengali tweets. In: 2018 5th International conference on emerging applications of information technology (EAIT), pp 1–4

Sarker M, Hossain MF, Liza FR, Sakib SN, Al Farooq A (2022) A machine learning approach to classify anti-social Bengali comments on social media. In: 2022 International conference on advancement in electrical and electronic engineering (ICAEEE). IEEE, pp 1–6

Schmidt A, Wiegand M (2017) A survey on hate speech detection using natural language processing. In: Proceedings of the 5th international workshop on natural language processing for social media, pp 1–10

Sengupta A, Bhattacharjee SK, Akhtar MS, Chakraborty T (2022) Does aggression lead to hate? Detecting and reasoning offensive traits in Hinglish code-mixed texts. Neurocomputing 488:598–617

Sharma A, Kabra A, Jain M (2022) Ceasing hate with moh: Hate speech detection in Hindi–English code-switched language. Inf Process Manag 59(1):102760

Sreelakshmi K, Premjith B, Soman K (2020) Detection of hate speech text in Hindi–English code-mixed data. Procedia Comput Sci 171:737–744

Subramanian M, Ponnusamy R, Benhur S, Shanmugavadivel K, Ganesan A, Ravi D, Shanmugasundaram GK, Priyadharshini R, Chakravarthi BR (2022) Offensive language detection in Tamil youtube comments by adapters and cross-domain knowledge transfer. Comput Speech Lang 76:101404

Subramanian M, Adhithiya G, Gowthamkrishnan S, Deepti R (2022) Detecting offensive Tamil texts using machine learning and multilingual transformer models. In: 2022 International conference on smart technologies and systems for next generation computing (ICSTSN). IEEE, pp 1–6

Thomson M, Murfi H, Ardaneswari G (2023) Bert-based hybrid deep learning with text augmentation for sentiment analysis of Indonesian hotel reviews. In: DATA, pp 468–473

Vashistha N, Zubiaga A (2020) Online multilingual hate speech detection: experimenting with Hindi and English social media. Information 12(1):5

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Adv Neural Inf Process Syst 30:1

Zampieri M, Ranasinghe T, Chaudhari M, Gaikwad S, Krishna P, Nene M, Paygude S (2022) Predicting the type and target of offensive social media posts in Marathi. Soc Network Anal Min 12(1):77

Zhang L, Liu B (2012) Sentiment analysis and opinion mining. In: Encyclopedia of machine learning and data mining

Zimmerman S, Kruschwitz U, Fox C (2018) Improving hate speech detection with deep learning ensembles. In: Proceedings of the 11th international conference on language resources and evaluation (LREC 2018)