Bengali Hate Speech Detection from Social Media using Ensemble Machine Learning Approach

Sadia Tarin[1*], Farzina Akther[2†], Pranta Paul[3†], Tanvinur Rahman Siam[4†]

[1*]Computer Science and Engineering Department, Port City International University, S Khulshi Road, Chittagong, 100190, Bangladesh.
[2]Computer Science and Engineering Department, Port City International University, S Khulshi Road, Chittagong, 100190, Bangladesh.
[3]Computer Science and Engineering Department, Port City International University, S Khulshi Road, Chittagong, 100190, Bangladesh.
[4]Computer Science and Engineering Department, Chittagong University of Engineering & Technology, Kaptai Highway, Chittagong, 4349, Bangladesh.

*Corresponding author(s). E-mail(s): sadiatarinpciu21@gmail.com;
Contributing authors: farzina.cse@gmail.com; prantapaul.cse.01@gmail.com; tanvinursiam@gmail.com;
†These authors contributed equally to this work.

Abstract

The increasing prevalence of hate speech in Bengali across social media is a growing concern for the government and platform providers. Timely detection and removal of such content are essential to preventing cyber violence and real-world conflicts. However, the informal nature of online communication, with variations in spelling and grammar, makes identification challenging.

This study proposes an ensemble-based machine learning model for detecting hate speech in Bengali. A diverse dataset was collected from various online sources, followed by comprehensive preprocessing and classification into three tasks: (i) binary classification (Hate Speech vs. Not Hate), (ii) multi-label classification (categorizing different types of hate speech), and (iii) target identification.

We explored machine learning algorithms alongside deep learning models and the ensemble approach. In our proposed approach, we applied bagging with Decision Tree classifiers to create an ensemble model. Then, we built a stacking ensemble model, integrating Random Forest, SVM, Logistic Regression, and the bagging ensemble classifiers. It achieved 91.49% accuracy with an F1-score of 91.49% on the imbalanced dataset, while on the balanced dataset, accuracy improved to 94.37% with an F1-score of 94.37%.

Keywords: Hate speech detection, Natural Language Processing, Machine Learning, Ensemble Learning, Bi-LSTM, CNN

1 Introduction

Social media has significantly transformed digital communication, providing a platform for users to voice their opinions and ideas freely. While this has enhanced connectivity and collaboration, it has also contributed to the rise of hate speech, cyberbullying, and online harassment. These issues have severe consequences for individuals, communities, and organizations. Cyberbullying, in particular, is a serious concern, involving the use of digital platforms to intimidate, harass, or harm others. Leading to adverse consequences for the victim, including emotional distress and health problems [1] [2]. Hate speech involves the use of offensive or derogatory language directed at individuals or groups based on attributes such as race, ethnicity, gender, sexual orientation, nationality, religion, or health status. It fosters discrimination and hostility, often leading to social division and psychological harm [3] [4]. The widespread occurrence of cyber hate, characterized by religious and racial discourse, holds the capacity to intensify tensions across diverse social media platforms. Furthermore, it poses the risk of fostering terrorist acts and other forms of antisocial behaviors[5].

The rise of Bengali hate speech on social media poses challenges due to linguistic complexities and class imbalance in detection models. Existing methods struggle with accuracy in classification and target identification. This study proposes an ensemble-based machine learning approach to enhance detection, aiding in safer online communication. Here we have worked on binary classification, multi-label classification (categorizing different types of hate speech) and target identification. The anonymity of these platforms can encourage individuals to express hate speech that they might refrain from using in direct, face-to-face interactions [6]. With nearly 205 million native Bengali speakers worldwide approximately 3.05% of the global population Bangla has a significant online presence [7] [8]. In Bangladesh, approximately 81.7% of the population has internet access, with more than 30 million people actively using social media, predominantly through mobile devices[9]. Notably, around 42 million Facebook users engage in Bengali, making up 1.9% of the platform's global user base[10]. With the high level of user engagement, it is essential to monitor and regulate Bengali hate speech to maintain a safer and more inclusive digital space. However, due to the linguistic complexity and informal variations of Bengali, manual detection of hate speech is impractical.

Due to the morphological richness and informal variations in Bengali, manually detecting hate speech remains a challenging task that requires an automated solution. The primary objectives are:

- Utilizing Natural Language Processing (NLP) for data preprocessing.
- Implementing various Machine Learning (ML) classifiers and Deep Learning (DL) algorithms, such as Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory (Bi-LSTM).
- Performing three classification tasks: one for binary classification, another for categorizing various hate speech types in a multi-label format, and a third for identifying the targets of hate speech.
- Assessing the performance of machine learning (ML), deep learning (DL), and ensemble learning models using metrics such as accuracy, precision, recall, and F1-score.

This research introduces a dataset comprising 50,281 Bengali text samples, sourced from the [BD-SHS] available on Kaggle, and addresses three key sub-tasks: classifying hate speech, categorizing hate speech types, and identifying targets of hate speech. It compares multiple models and proposes an ensemble classifier that achieves state-of-the-art performance. The results are intended to help social media platforms in mitigating hate speech, ensuring a safer digital space for Bengali-speaking users.

2 Related Work

Several studies have been conducted on hate speech detection using machine learning, deep learning and ensemble learning techniques.

Romim et al.[11] introduce a BD-SHS dataset of 50,281 Bengali comments collected from online social media and streaming platforms using the Facepager tools. The dataset is annotated three-level hierarchical approach. The author introduced informal word embedding. The informal word embedding trained on noisy, informal texts demonstrated relatively improved performance. The authors utilized both SVM and Bi-LSTM Deep Learning based methods with a hate speech detection dataset consisting of 50,281 samples, achieving the highest F1-Score of 91.0% for classifying hate speech. The data was divided, with 75% allocated for training the model, 15% for validating the model, and the remaining 15% for testing the model's performance. The data's noisy nature is characterized by spelling mistakes and dialect variations. The classifier performed poorly in accurately identifying HS targets. The model's performance could be further improved by refining the annotation schemes and addressing conflicts within the annotations to enhance its accuracy.

Nugroho et al.[12] utilized a dataset consisting of 14,509 samples for Hate Speech Identification was employed, sourced from data.world. The study specifically focused on a Twitter dataset dedicated to hate speech and offensive content identification. The authors implemented machine learning and deep learning based Random Forest, AdaBoost, and Neural Network models in their analysis. Utilizing the entire set of 14,509 samples, they achieved the highest accuracy of 0.722 in the classification of hate speech and offensive language. Notably, the detection of hate speech and offensive words using Random Forest exhibited superior levels of accuracy and precision compared to the AdaBoost method and Neural Network. The author explored a small dataset. The scope for improvement includes addressing dataset size and exploring new methods for audio-based hate speech detection to enhance overall accuracy.

Karim et al.[13] introduced the Bengali Hate Speech Dataset, which is publicly accessible. This Hate Speech (HS) dataset categorized hate speech into four types. The authors presented the Bengali Hate Speech Dataset and developed a Bengali word embedding model named BengFastText. They implemented deep learning based Multichannel Convolutional-LSTM network (MConv-LSTM), incorporating both CNN and LSTM networks. The authors trained various machine learning baseline models, including SVM, KNN, LR, NB, DT, RF, and GBT. By utilizing the Multichannel Convolutional-LSTM network, they achieved the highest F1-Score of 90.45% for hate speech detection across different types. The study faces challenges with misspellings and language variations. Future improvements could include using MOWE embeddings, refining network architectures, and better distinguishing between hate speech, abuse, and cyberbullying.

Abro et al.[14] implemented eight distinct classifiers for the detection of hate speech. The study involved the collection of a publicly available dataset containing hate speech tweets. Various preprocessing techniques were applied to this dataset, and a split of 80-20 was employed for the preprocessed data. The authors executed three different feature engineering techniques, namely n-gram with TFIDF, Word2vec, and Doc2vec. Among the eight classifiers, the SVM classifier exhibited the best performance. In their machine learning model, LR, NB, RF, SVM, KNN, DT, AdaBoost, and MLP were all utilized with a hate speech detection dataset comprising 14,509 tweet samples. The highest values for recall (0.79), precision (0.77), accuracy (0.79), and F-measure (0.77) were achieved by SVM using TFIDF features representation with bigram features for classifying hate speech. The study encounters difficulties with inefficiencies in real-time prediction and difficulties in assessing the severity of hate speech. Enhancements can focus on integrating lexicon-based approaches and expanding the dataset to improve classification accuracy and performance.

Roy et al.[25] proposed a Deep Convolutional Neural Network (DCNN) model utilizing GloVe embeddings for hate speech detection on Twitter. The model achieved a precision of 0.97, recall of 0.88, and an F1-score of 0.92, outperforming existing approaches. The dataset, sourced from Kaggle.com, consisted of 31,962 English tweets. Machine learning models were also employed with TF-IDF feature extraction. The study encountered challenges with dataset imbalance and limited hate speech detection in tweets. Improvements could focus on including multimedia content, expanding language diversity, and enhancing model performance by increasing the dataset size.

Islam et al.[15] employed a diverse set of machine learning algorithms. The study specifically concentrated on pure Bengali data extracted from social media pages (e.g., Facebook, YouTube), groups, and comment sections of news portals. This data was categorized into two classes. The authors trained various machine learning models, including LR, NB, RF, SVM, and KNN, for the detection of hate speech. The highest accuracy they achieved was 67% using the Random Forest (RF) algorithm. The model currently classifies speech into only two categories and needs improvements for more accurate hate speech detection and enhanced recognition performance.

Das et al.[16] proposed an encoder–decoder-based model for Bengali comment classification, utilizing a dataset of 7,425 comments categorized into seven classes, including hate speech and aggressive content. Preprocessing involved tokenization, stopword removal, stemming, and emoticon handling. Feature extraction was performed using TF-IDF and word embeddings. The study applied various machine learning algorithms, achieving 74% accuracy with GRU and LSTM decoders, while an attention-based decoder outperformed previous models with 77% accuracy. The study struggles with dataset diversity and the limited ability to classify mixed-language and photo-based comments. Improvements could focus on enhancing the model's ability to handle diverse speech types, increasing dataset variety, and automating the classification process for better scalability and accuracy.

Romim et al. [17] established HS-BAN, a binary class hate speech (HS) dataset comprising 50,000 comments gathered from Facebook and YouTube. The authors utilized stratified sampling to partition the dataset into training (80%) and test (20%) sets, ensuring equal representation of each category, distinguishing hate and not-hate comments, in both subsets. Furthermore, within the training set, a further division was made into a training subset (80%) and a development subset (20%) for feature selection and hyper-parameter tuning during model development. In this study, an F1-score of 86.78% was achieved for the Bi-LSTM model utilizing FastText informal word embedding. The study contends with non-traditional slang and noisy social media text. Improvements can focus on expanding the dataset with mixed-language and photo comments and enhancing the model's adaptability and interpretability.

Emon et al. [18] introduced various Machine Learning (ML) and Deep Learning (DL) algorithms aimed at identifying diverse forms of abusive content in the Bengali language. The authors proposed Bengali stemming rules derived from specific grammatical principles, incorporating rules aligned with the grammatical structure of the Bengali language. The comments collected for this study exclusively contained the Bengali language and were categorized into seven classes, including slang, religious hatred, personal attack, politically violated, antifeminism, positive, and neutral. The authors applied preprocessing to the dataset and employed machine learning and deep learning based models such as LinearSVC, LR, MNB, ANN, RNN, and RF with LSTM. The deep learning-based algorithm, RNN, achieved the highest accuracy of 82.20%. The study encounters difficulties in accurately detecting abusive content due to language complexities and limited linguistic resources. To improve, the model could be extended by applying additional deep learning algorithms and integrating Bengali spelling correction techniques.

Mnassri et al.[19] explored deep learning models for hate speech classification, leveraging transformer-based models such as BERT and ensemble techniques, including soft voting, hard voting, maximum value, and stacking. The study utilized three publicly available Twitter datasets (Davidson, HatEval2019, OLID) and combined them into a unified DHO dataset for multi-label classification. The authors trained several models, including BERT-MLP, BERT-CNN, and BERT-LSTM, achieving the highest F1-score of 97% on the Davidson dataset and 92% on the DHO dataset, demonstrating the effectiveness of ensemble learning. The study faced issues with dataset imbalance, leading to overfitting. High computational requirements for models like BERT limited the use of complex architectures. The authors plan to enhance the model by tackling data imbalance and exploring K-BERT and advanced ensemble methods for better performance and reduced bias.

Mridha et al.[20] introduced the L-BOOST model, which integrates the LSTM model with the AdaBoost-BERT model. The dataset used in this study consisted of 16,800 posts, comments, and memes collected from diverse Bengali websites, blogs, and various social media platforms. To enhance accuracy in the data crawling process and prioritize privacy, the authors removed all permalinks, dates, times, and user details. For feature extraction, the authors employed TF-IDF, Word2vec, fastText, and BERT embeddings. The L-Boost's model achieved the highest accuracy of 95.11%. The model faces challenges with data imbalances, regional variations, and overfitting.

These studies highlight the effectiveness of various models and datasets in hate speech detection, underscoring the need for further advancements in Bengali language processing.

3 Methodology

3.1 Overall Research Methodology

Bengali hate speech detection involves multiple stages, as illustrated in Fig.1. The dataset, collected from Kaggle, undergoes preprocessing before model training. The preprocessing steps include removing HTML tags, URLs, punctuation, special characters, stop words, and extra whitespace. These steps are chosen to eliminate noise from the text and improve the model's ability to learn meaningful patterns.

For feature extraction, we employ Term Frequency-Inverse Document Frequency (TF-IDF) for traditional machine learning models, as it effectively represents textual data by emphasizing important words while reducing noise. On the other hand, deep learning models leverage word embeddings, which capture contextual relationships between words and enhance semantic understanding. To ensure a robust evaluation, the dataset is split into training and testing subsets using three different ratios: 60:40, 70:30, and 80:20. These splits are chosen to analyze model performance under varying data distributions and to assess the effect of training data size on model accuracy.

Both machine learning and deep learning approaches are implemented. Machine learning models are trained individually and then combined using bagging and stacking ensemble techniques. Bagging is applied to Decision Tree classifiers to reduce variance and prevent overfitting, while stacking combines three machine learning models with the bagging classifiers to leverage the strengths of multiple models. The combination of these ensemble techniques is chosen to enhance predictive accuracy and generalization capabilities.

The performance of all models is evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score, to provide a comprehensive assessment of their effectiveness.

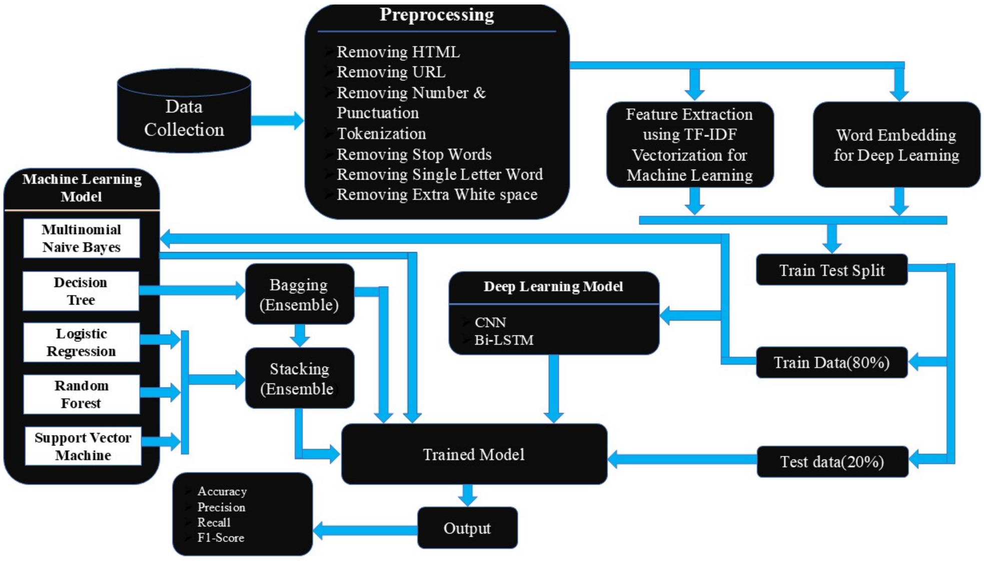


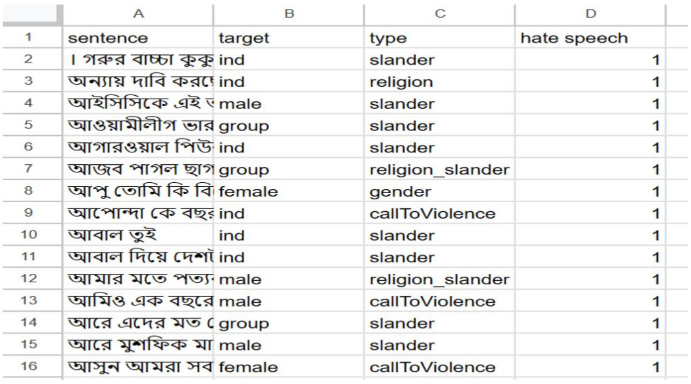
Fig. 1 Proposed Methodology for Effective Implementation

3.2 Data Collection

This study uses the BD-SHS dataset[11], which is publicly available on Kaggle. It comprises 50,281 Bengali comments gathered from social media and streaming platforms through the Facepager tool. To ensure data diversity, duplicate and highly similar comments were removed using the Jaccard Index with a cutoff of 0.8. The dataset is annotated at three levels: identifying hate speech in a binary manner, determining the target, and categorizing the type of hate speech. The Jaccard Index method was applied to eliminate redundancy and maintain a diverse vocabulary.

3.3 Dataset Overview

- **Hate Speech Identification:** Binary classification (Hate Speech, Not Hate Speech) with 50,281 samples.
  Hate Speech Categorization: Multi-label classification (Call to Violence, Gender, Religion, Slander) with 24,137 samples.
  Hate Speech Target Identification: Multi-label classification (Male, Female, Individual, Group) with 24,137 samples.



	A	B	C	D
1	sentence	target	type	hate speech
2	। গরুর বাচ্চা কুকু	ind	slander	1
3	অন্যায় দাবি করছে	ind	religion	1
4	আইসিসিকে এই ত	male	slander	1
5	আওয়ামীলীগ ভার	group	slander	1
6	আগারওয়াল পিউ	ind	slander	1
7	আজব পাগল ছাগ	group	religion_slander	1
8	আপু তোমি কি বি	female	gender	1
9	আপোন্দা কে বছর	ind	callToViolence	1
10	আবাল তুই	ind	slander	1
11	আবাল দিয়ে দেশে	ind	slander	1
12	আমার মতে পত্য	male	religion_slander	1
13	আমিও এক বছরে	male	callToViolence	1
14	আরে এদের মত (	group	slander	1
15	আরে মুশফিক মা	male	slander	1
16	আসুন আমরা সব	female	callToViolence	1

Fig. 2 Sample of the Collected Dataset

3.4 Data Preprocessing

Data preprocessing plays a vital role in maintaining quality and enhancing model performance. The key preprocessing steps include:

Removing HTML Tags: As the dataset originated from web scraping, it included unnecessary HTML elements. A regular expression-based approach was used to remove any HTML elements from the text, ensuring that only meaningful content remained.

Removing URLs: Social media comments often contain hyperlinks, which do not contribute to the classification task. These URLs were identified and removed using a regex-based filtering technique.

Removing Numbers, Punctuation, and Special Characters: To standardize the text, all numerical values, punctuation marks (e.g., commas, semicolons, hyphens, question marks, exclamation marks), and special characters were removed. Additionally, distorted or noisy characters that do not comply with Unicode encoding removed, as they added little value to the overall semantic context [24].

Tokenization: The Bengali text was tokenized into individual words to facilitate further processing, such as stopword removal and feature extraction.

Removing Stop Words and Single-letter Words Filtering: Common stopwords in Bengali were eliminated as they do not add value to the classification task.

Additionally, single-letter words were removed since they either lacked semantic importance or were errors introduced during data entry.
Removing Extra White Space: Some comments contained unnecessary spaces, which were removed to ensure uniform text representation. The text was also converted to lowercase for consistency.
Splitting Data: The dataset was partitioned into training and testing subsets to assess model performance. Several partition ratios were explored, including 60:40, 70:30, and 80:20. A stratified sampling technique was used to maintain the class distribution across both sets.
Multi-Label Binarization: For multi-label classification tasks, a MultiLabelBinarizer was used to convert the target labels (e.g., types of hate speech) into binary format. This allows the model to classify comments into multiple categories simultaneously.

3.5 Random Over-Sampling

Random over-sampling is a technique employed to tackle class imbalance in hate speech detection by augmenting the minority class. To address class imbalance, random over-sampling was utilized to replicate instances from the minority classes until all classes had equal representation [21]. In this study, it was specifically applied to balance the hate speech and non-hate speech classes, ensuring more equal representation for better model performance.

3.6 Feature Extraction

To effectively represent textual data and enhance classification performance, we employ multiple feature extraction techniques:
TF-IDF Vectorization: TF-IDF (Term Frequency-Inverse Document Frequency) vectorization is a technique used to convert text data into numerical representations by evaluating the importance of each word in the document relative to the entire dataset. It assigns higher weights to terms that appear frequently in a document but less frequently across all documents, highlighting significant words for machine learning models.

Character N-grams with TF-IDF: Combining TF-IDF with character n-grams involves applying TF-IDF to the representations of text based on character n-grams. This allows the model to capture information about the character sequences in addition to word-level information.

Word Unigram with TF-IDF: The Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer is applied to individual words in the document, capturing the importance of each word in relation to the entire corpus. This method helps in emphasizing terms that are unique to specific documents while reducing the weight of common words.

Word-Character Combined Feature: To combine both word-level and character-level features, a feature union of TF-IDF with word unigram and character n-grams (up to 5-grams) is performed. This allows the model to capture both the meaning expressed by individual words and the patterns embedded in character sequences, providing a more comprehensive representation of the text.

Word Embeddings: For word representation, we used pretrained Multilingual FastText (MFT) embeddings [22], which map words into a high-dimensional vector space. These embeddings are available for download from the official fastText repository.

3.7 Model Training and Evaluation

Machine learning and deep learning models are trained separately, and ensemble methods such as bagging and stacking are implemented to enhance performance. To address dataset imbalance, random oversampling is applied. The models are evaluated based on accuracy, precision, recall, and F1-score.

This approach provides a systematic framework for Bengali hate speech detection by combining machine learning and deep learning models with efficient preprocessing, feature extraction, and ensemble learning strategies.

3.7.1 Machine Learning Models

We employ different machine learning models for classification, each optimized with specific hyperparameters:

Random Forest (RF)

Random Forest (RF) is an ensemble classifier that combines several decision trees to enhance its robustness and accuracy. We trained $n\_estimators = 100$ for binary and $n\_estimators = 200$ for multilabel classification to balance accuracy and efficiency. The parameter $random\_state = 42$ ensures reproducibility, while $max\_depth = None$ allows trees to grow fully for better pattern recognition. The $min\_samples\_split = 2$ prevents unnecessary splits, and $min\_samples\_leaf = 1$ ensures flexibility in leaf nodes. These parameters were fine-tuned using GridSearchCV to achieve optimal performance.

Support Vector Machine (SVM)

A margin-based classifier that identifies the best hyperplane for separating classes. The parameter $C = 1.0$ manages the balance between the margin size and the risk of misclassification, ensuring a balance between complexity and generalization. The hinge loss function was selected for its suitability in linear classification, while $max\_iter =$

500 ensures sufficient iterations for convergence. The $random\_state = 42$ guarantees reproducibility, and all parameters were fine-tuned using GridSearchCV for optimal performance.

Logistic Regression (LR)

A commonly used classifier in text classification tasks, utilizing the sigmoid function for binary decision-making. We set $C = 10.0$ to control the regularization strength, striking a balance between bias and variance to avoid overfitting. The parameter $max\_iter = 500$ was selected to allow sufficient iterations for convergence. The $random\_state = 42$ ensures reproducibility, and the hyperparameters were optimized using GridSearchCV to achieve the best performance while avoiding overfitting.

Naïve Bayes (NB)

A probabilistic model derived from Bayes' theoremS, suitable for text classification tasks. We employed a multinomial Naïve Bayes classifier with $\alpha = 0.5$ to smooth probabilities and handle zero probabilities effectively, improving model performance. GridSearchCV was used to fine-tune this parameter and optimize the model for the best results.

Decision Tree (DT)

A rule-based classifier that splits data based on feature importance. We optimized the tree using $random\_state = 42$ for reproducibility, $criterion =' gini'$ to measure impurity, and $max\_depth = None$ to allow the tree to grow fully and capture complex patterns. The parameter $min\_samples\_split = 5$ was used to prevent overfitting by requiring more samples to split, while $min\_samples\_leaf = 1$ ensured flexibility in leaf nodes. These parameters were selected through GridSearchCV to enhance model performance.

3.7.2 Ensemble Learning

To enhance classification performance, we employ ensemble learning techniques:

Bagging

Bootstrap aggregation (bagging) generates multiple versions of the classifier using different data subsets, improving stability and reducing overfitting. A Decision Tree-based bagging classifier is trained with $n\_estimators = 10$ to balance the number of classifiers and computational efficiency. The model was optimized through GridSearchCV to adjust the parameters and obtain the best possible performance.

Stacking

Stacking is an ensemble method where base classifiers generate predictions, which a meta-classifier then refines for better accuracy [23]. A meta-learning technique combining multiple classifiers, where base models (RF, LR, and bagging) generate predictions, which are then processed by a final meta-classifier (SVM) for improved decision-making.

3.7.3 Deep Learning Models

We leverage deep learning architectures to capture complex text representations:

Bidirectional Long Short-Term Memory (Bi-LSTM)

Bi-LSTMs are well-suited for capturing long-range dependencies by processing text in both forward and backward directions. In our Bi-LSTM architecture, we built upon the model introduced by Romim et al. [11] which includes a bidirectional LSTM layer with 100 units, an average pooling layer, a fully connected hidden layer containing 16 units, and output layers that employ softmax and sigmoid activation functions for specific tasks. However, in our implementation, we expanded the Bi-LSTM layer to 128 units. This adjustment was made to better capture complex patterns improve model performance and handling complex datasets.

Convolutional Neural Network (CNN)

CNNs extract hierarchical features from text by using a 1D convolutional layer with 128 filters and a kernel size of 8. This is followed by a MaxPooling1D layer (with a pool size of 2) to reduce the dimensionality of the features, and a Dropout layer (rate=0.5) to prevent overfitting. The processed output is then passed through two Dense layers, each with 16 units and `ReLU` activation, before being fed into a final Dense layer. This last layer has 1 unit for binary classification or 4 units for multilabel classification, with `sigmoid` activation. The model is trained with the `Adam` optimizer (`learning_rate=0.001`) over 5 epochs, with a batch size of 16. TThese parameter selections are designed to optimize performance, enhancing the model's accuracy.

This framework adopts a holistic approach to Bengali hate speech detection by incorporating machine learning, deep learning, and ensemble methods, ensuring a balance between accuracy and efficiency.

4 Results and Discussion

This section discusses the evaluation of different machine learning (ML) and deep learning (DL) models for identifying, categorizing, and targeting hate speech. The performance of each model is assessed using three distinct feature representations: F1 (TF-IDF + word unigram), F2 (TF-IDF + character n-grams), and F3 (TF-IDF + word unigram + character n-grams). The models are evaluated on both imbalanced and balanced datasets to ensure a comprehensive assessment.

Both machine learning and ensemble learning models have been applied and trained using these features. Specifically, ensemble learning has been utilized with the F2 feature for hate speech identification and categorization, while the F3 feature has been used for hate speech target identification. The best-performing features for each task have been highlighted in this study, demonstrating their effectiveness in improving model performance. Our stacking ensemble model consistently outperforms conventional classifiers, demonstrating its robustness in hate speech analysis. The evaluation is conducted using key performance metrics, including Accuracy (A), Precision (P), Recall (R), and F1-score (F1).

4.1 Result Analysis for Hate Speech Identification

Table 1 Summary of Hate Speech Identification Performance at an 80:20 Split Ratio

Model + Feature	**Imbalanced Dataset**				**Balanced Dataset**			
	A (%)	P (%)	R (%)	F1 (%)	A (%)	P (%)	R (%)	F1 (%)
RF + F2	89.62	89.79	89.62	89.59	93.02	93.02	93.02	93.02
SVM + F2	91.34	91.38	91.34	91.34	92.55	92.56	92.55	92.55
LR + F2	90.69	90.74	90.69	90.68	93.29	93.29	93.29	93.29
MNB + F3	85.73	85.76	85.73	85.73	86.64	86.88	86.64	86.61
DT + F2	86.35	86.35	86.35	86.35	89.68	89.80	89.68	89.67
Bi-LSTM + MFT	88.94	89.09	88.94	88.92	88.20	88.43	88.20	88.18
CNN + MFT	85.51	85.51	85.51	85.51	89.19	89.25	89.19	89.18
Bagging + F2	89.40	89.40	89.40	89.40	92.73	92.75	92.73	92.73
Stacking + F2	**91.49**	**91.50**	**91.49**	**91.49**	**94.37**	**94.37**	**94.37**	**94.37**

In every tabes, A stands for Accuracy, P stands for Precision, R stands for Recall, and F1 stands for F1-Score. As shown in Table 1 the stacking ensemble model attains the highest performance, achieving an accuracy of 91.49% and an F1-score of 91.49% on the imbalanced dataset, which further improves to 94.37% when using a balanced dataset. Traditional classifiers such as SVM and LR perform competitively, yet the ensemble approach consistently outperforms them. In addition, balancing the dataset significantly enhances the performance of most models, highlighting the adverse impact of class imbalance.

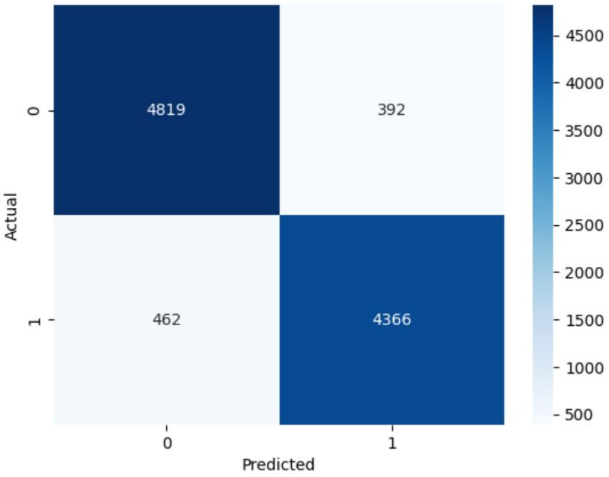


Fig. 3 Confusion Matrices for Hate Speech Identification using the Stacking Ensemble Model on Imbalanced Datasets.

In the confusion matrix for the imbalanced dataset Fig. 3, the model demonstrates a reasonable balance between true positives (4366) and true negatives (4819), with a relatively low misclassification rate despite the dataset imbalance. True positives represent correctly predicted “hate speech”, while true negatives represent correctly predicted “not hate speech”. The model also shows 392 false positives, where “not hate speech“ are misclassified as “hate speech”, and 462 false negatives, where “hate speech” is incorrectly identified as “not hate speech”. Despite the class imbalance, the model maintains good classification performance overall.

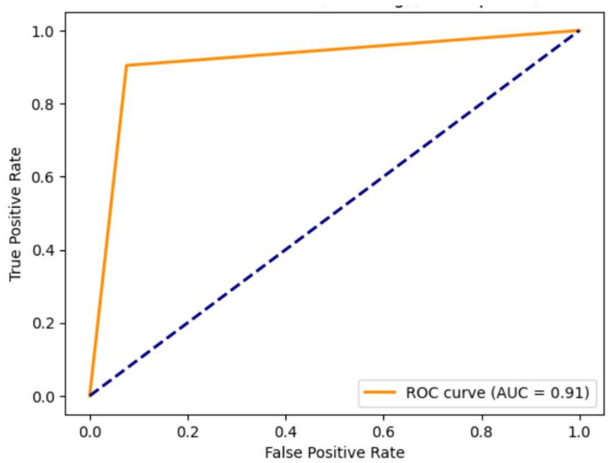


Fig. 4 ROC Curves for Hate Speech Detection Utilizing the Stacking Ensemble Model on Imbalanced Datasets

The ROC curve, as illustrated in Fig. 4, visualizes the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR). A higher Area Under the Curve (AUC) signifies better predictive accuracy for both positive and negative classes. Specifically, the model achieves an AUC of 0.91 on the imbalanced dataset, signifying excellent performance in distinguishing between “hate speech” and “non-hate speech”.

After balancing the dataset Fig. 5, the number of true positives (4923) and true negatives (4913) increases, while false positives (298) and false negatives (289) decrease significantly. This indicates an improvement in classification accuracy. True positives represent instances where the classifier correctly predicted “hate speech” for texts that were indeed “hate speech”, and true negatives represent cases where “not hate speech” was accurately predicted. False positives represent instances where the classifier incorrectly predicted “hate speech” for texts that were “not hate speech”, while false negatives represent cases where “hate speech” was wrongly predicted as “not hate speech.” These improvements suggest that balancing the dataset enhances the model’s performance, reducing misclassification and improving overall classification accuracy.

The ROC curve in Fig. 6 depicts the correlation between the True Positive Rate and the False Positive Rate. With an AUC of 0.94, the model exhibits outstanding performance, reflecting improved accuracy and more effective class separation after dataset balancing.

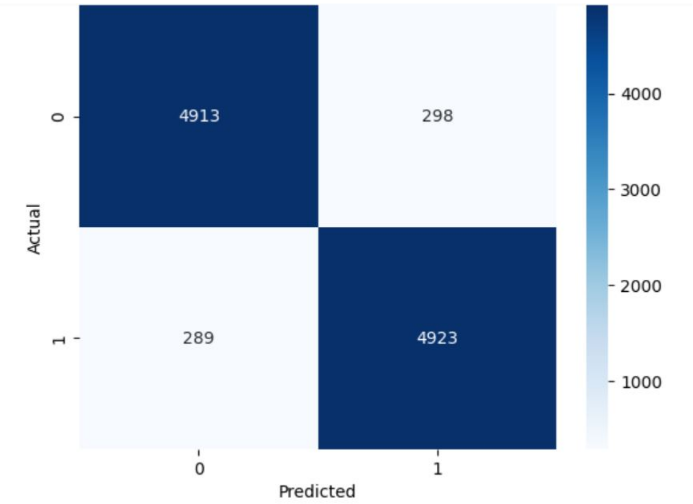


Fig. 5 Confusion Matrix for Hate Speech Identification using the Stacking Ensemble Model on Balanced Datasets

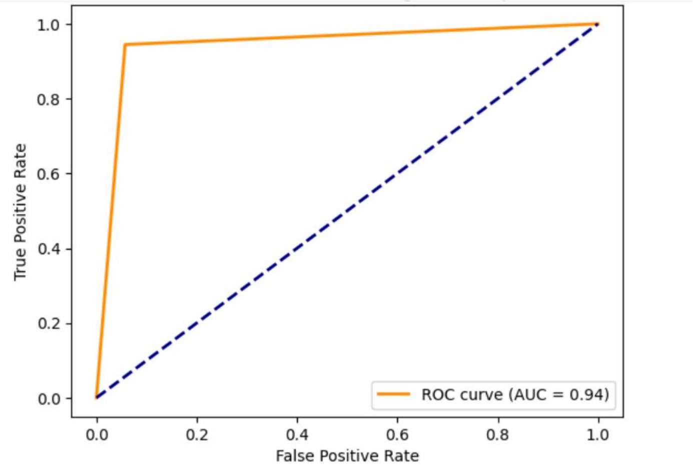


Fig. 6 ROC Curves for Hate Speech Identification using the Stacking Ensemble Model on Balanced Datasets.

Overall, the analysis indicates that dataset balancing improves both the confusion matrix and ROC curve, leading to a more robust stacking model for hate speech identification.

4.2 Result Analysis for Hate Speech Categorization

As shown in the below Table 2, the stacking ensemble model achieves the highest performance, with an F1-score of 89.32% on the imbalanced dataset, which further improves to 95.40% after balancing the dataset. Although traditional models like SVM and LR yield strong results, the stacking approach achieves superior performance. The considerable improvement in balanced data underscores the need to address class imbalance for more reliable hate speech categorization.

Table 2 Summary of Hate Speech Categorization Performance at an 80:20 Split Ratio

Model + Feature	**Imbalanced Dataset**				**Balanced Dataset**			
	A (%)	P (%)	R (%)	F1 (%)	A (%)	P (%)	R (%)	F1 (%)
RF + F2	76.35	88.49	85.85	86.5	90.21	95.24	94.35	94.7
SVM + F2	80.07	91.09	87.79	89.23	87.49	94.73	92.72	93.66
LR + F2	78.91	90.60	87.02	88.54	89.24	95.57	93.69	94.56
MNB + F1	63.44	84.09	74.28	73.44	74.56	89.50	83.47	84.13
DT + F1	66.63	83.25	82.02	82.61	84.69	92.54	91.50	92.02
Bi-LSTM + MFT	74.42	87.31	84.40	85.54	77.09	89.66	84.70	86.91
CNN + MFT	69.53	85.66	80.53	82.46	82.58	92.61	88.57	90.44
Bagging + F2	73.96	87.55	85.39	86.32	87.32	94.50	92.64	93.53
Stacking + F2	**80.07**	**90.10**	**88.65**	**89.32**	**91.29**	**96.17**	**94.69**	**95.40**

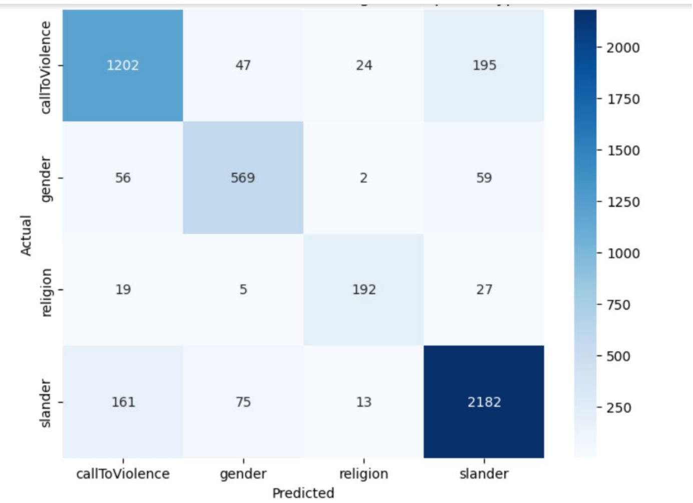


Fig. 7 Confusion Matrices for Hate Speech Categorization using the Stacking Ensemble Model on Imbalanced Datasets

The confusion matrix for the imbalanced dataset Fig. 7 reveals significant misclassification across multiple categories. The Stacking ensemble model's confusion matrix, however, provides a clearer view of misclassification ratios across true and predicted class levels. Despite some misclassification ratios, the model demonstrates a balanced ability to accurately classify both true and predicted instances. True positives are recorded at 1202, 569, 192, and 2182 cases for the "callToViolence", "gender", "religion", and "slander" classes, respectively, indicating accurate predictions for these classes.

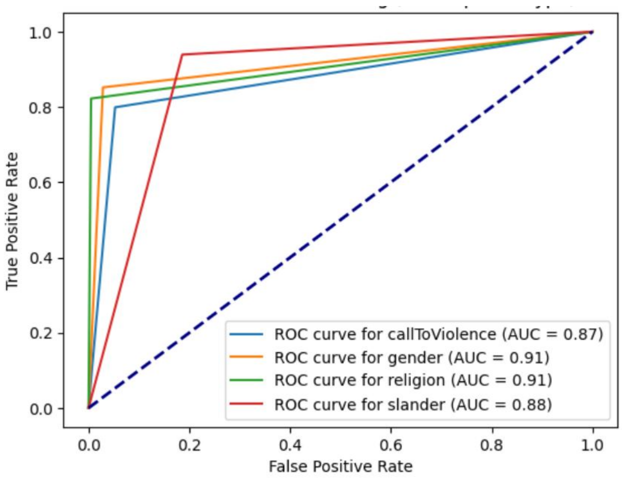


Fig. 8 ROC Curves for Hate Speech Categorization using the Stacking Ensemble Model on Imbalanced Datasets

The ROC curve in Fig. 8 illustrates the demonstrates the model's performance in distinguishing hate speech types across all categorie. The AUC for hate speech categories such as “callToViolence”, “gender”, “religion”, and “slander” increases, with scores of 0.87, 0.91, 0.91, and 0.88, respectively. Notably, the “gender” and “religion” classes show the highest AUC of 0.91, demonstrating strong model performance.

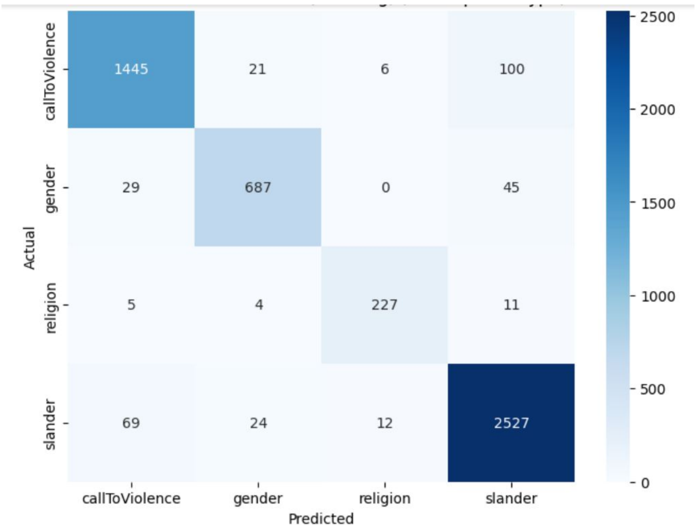


Fig. 9 Confusion Matrices for Hate Speech Categorization using the Stacking Ensemble Model on Balanced Datasets

After balancing the dataset Fig. 9, false positives and false negatives decrease, suggesting an improvement in model accuracy across all hate speech categories. The

Stacking model's confusion matrix shows improved true positives: 1445, 687, 227, and 2527 for the "callToViolence", "gender", "religion", and "slander" categories. This indicates that balancing the dataset enhances classification accuracy and reduces misclassification.

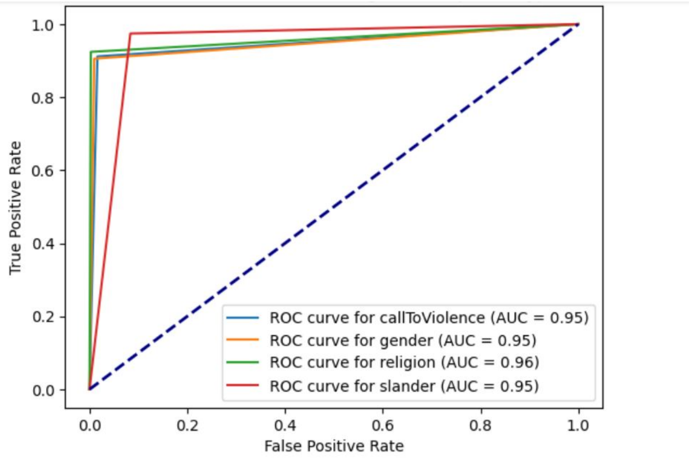


Fig. 10 ROC Curves for Hate Speech Categorization using the Stacking Ensemble Model on Balanced Datasets

The ROC curve in Fig. 10 shows improved AUC scores after balancing the dataset. "CallToViolence", "gender", and "slander" achieve an AUC of 0.95, while "religion" reaches 0.96, indicating enhanced model performance.

These results highlight that dataset balancing enhances classification performance, leading to better hate speech categorization.

4.3 Result Analysis for Hate Speech Target Identification

Table 3 summarizes the performance for identifying the target of hate speech. Here, the stacking ensemble model (using F3) clearly outperforms other methods, reaching an accuracy of 69.74% and an F1-score value of 73.93% for the imbalanced dataset, which improves to 85.78% and 88.38%, respectively, on the balanced dataset. Traditional models such as SVM and LR perform moderately, while MNB consistently underperforms. These results indicate that ensemble methods are particularly effective for complex tasks like target identification, especially when class imbalance is mitigated.

The confusion matrix for the imbalanced dataset Fig. 11 indicates notable misclassification, particularly among different target groups. The matrix displays true and predicted class levels along with misclassification ratios. Despite the observed misclassification percentages, the model demonstrates a balanced ability to accurately classify both true and predicted instances. True positives are recorded at 939, 240, 1239, and 1066 cases for the "female", "group", "ind", and "male" classes, respectively, indicating accurate predictions for these classes.

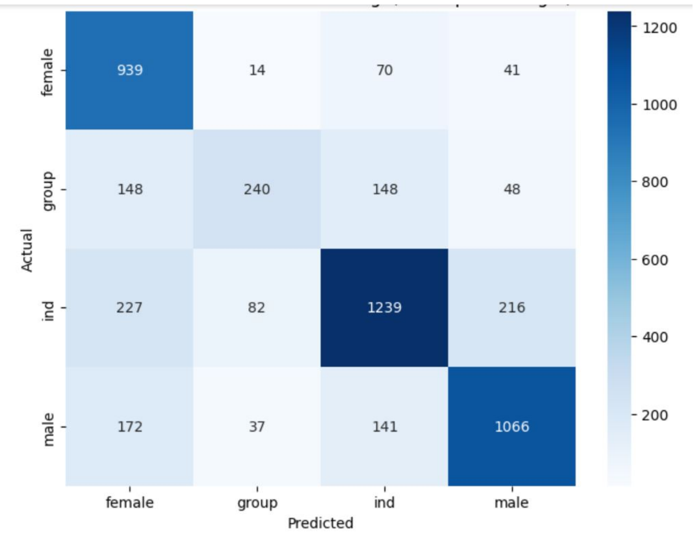


Fig. 11 Confusion Matrices for Hate Speech Target Identification using the Stacking Ensemble Model on Imbalanced Datasets.

Table 3 Summary of Hate Speech Target Identification Performance at an 80:20 Split Ratio

Model + Feature	**Imbalanced Dataset**				**Balanced Dataset**			
	A (%)	P (%)	R (%)	F1 (%)	A (%)	P (%)	R (%)	F1 (%)
RF + F1	55.2	74.97	59.32	65.77	80.79	90.42	83.08	86.45
SVM + F3	64.04	75.90	68.49	71.93	80.95	87.64	83.81	85.66
LR + F2	61.39	77.57	64.51	70.29	80.78	89.55	82.53	85.84
MNB + F1	29.18	84.75	29.18	41.99	51.77	90.72	51.97	64.83
DT + F1	50.29	63.70	65.35	64.51	77.34	83.74	84.46	84.08
Bi-LSTM + MFT	62.45	74.50	62.98	67.40	63.81	75.56	64.62	68.45
CNN + MFT	56.88	72.33	57.64	61.81	72.16	81.84	75.49	78.47
Bagging + F2	56.88	74.23	62.44	67.46	78.53	88.72	81.10	84.63
Stacking + F3	**69.74**	**77.54**	**71.08**	**73.93**	**85.78**	**90.72**	**86.34**	**88.38**

The ROC curve in Fig. 12 demonstrates the model's performance in distinguishing hate speech targets across all categories. The AUC for "female" and "male" is 0.88 and 0.84, respectively, indicating good performance. For "group" and "ind," the AUC scores are 0.69 and 0.79, with "group" having the lowest AUC.

After balancing Fig. 13, false positives and false negatives are significantly reduced, leading to improved model performance. The confusion matrix shows true positives at 1128, 518, 1578, and 1326 cases for the "female," "group", "ind", and "male" classes, respectively, reflecting more accurate predictions and a reduction in misclassification.

The ROC curve in Fig. 14 demonstrates the connection between the True Positive Rate and the False Positive Rate, showing that dataset balancing improves AUC scores across all target categories. A higher AUC indicates better model performance. For "female", "ind", and "male", the AUC scores are 0.94, 0.91, and 0.93, respectively,

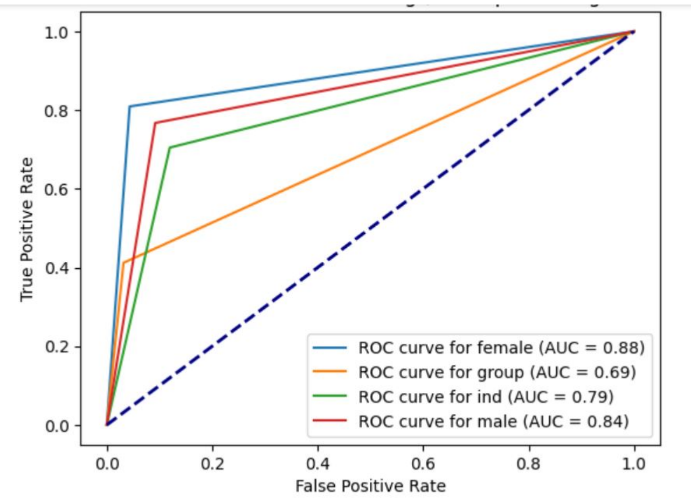


Fig. 12 ROC Curves for Hate Speech Target Identification using the Stacking Ensemble Model on Imbalanced Datasets.

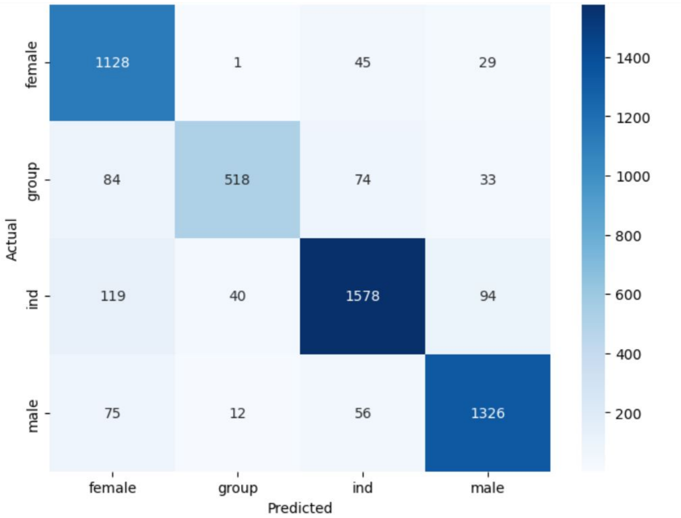


Fig. 13 Confusion Matrices for Hate Speech Target Identification using the Stacking Ensemble Model on Balanced Datasets.

reflecting excellent performance. The “group” category has a good AUC score of 0.86, with “female” achieving the highest score of 0.94.

These findings confirm that balancing the dataset improves the accuracy and consistency of identifying hate speech target.

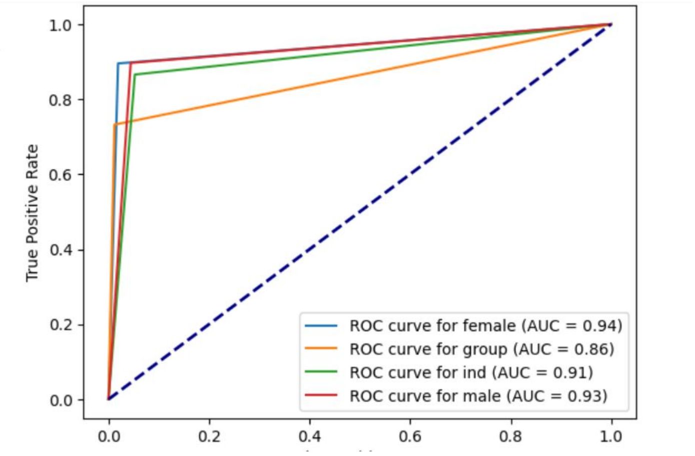


Fig. 14 ROC Curves for Hate Speech Target Identification using the Stacking Ensemble Model on Balanced Datasets.

4.4 Comparison with Existing Approaches

The comparison of Hate Speech Identification, Hate Speech Categorization, and Target Identification between our proposed method and existing approaches is given below. In this context, **U** denotes Word Unigrams (F1 feature) and **C** denotes Character n-grams (F2 feature).

Comparision for Hate Speech Identification:

A comparative analysis with Romim et al. [11] is presented in . Our stacking ensemble model achieves the highest accuracy (94.37%) and F1-score (94.37%), outperforming all existing SVM-based models.

Table 4 Comparison with Current Hate Speech Identification Methods

Reference	Model + Feature	Accuracy (%)	F1-score (%)
Romim et al. [11]	SVM + U	–	88.70
	SVM + U + C	–	90.80
	SVM + C	–	90.90
Proposed Approaches	SVM + U	90.55	90.50
	SVM + U + C	93.20	93.20
	SVM + C	92.55	92.55
	Stacking Ensemble Model	**94.37**	**94.37**

Comparision for Hate Speech Categorization:

Table 5 compares the categorization performance. Our proposed stacking ensemble model achieves the best F1-score of 95.4%, significantly improving upon previous methods.

Comparison for Hate Speech Target Identification:

Table 6 presents the evaluation of different models in identifying the target of hate speech. The stacking ensemble model outperforms all baselines, achieving an F1-score of 88.38%.

Table 5 Comparison with Current Hate Speech Categorization Methods

Reference	Model + Feature	Accuracy (%)	F1-score (%)
Romim et al. [11]	SVM + U	–	86.80
	SVM + C	–	89.10
	SVM + U + C	–	88.70
Proposed Approaches	SVM + U	84.42	91.93
	SVM + C	87.49	93.66
	SVM + C + U	88.87	94.50
	Stacking Ensemble Model	**91.29**	**95.40**

Table 6 Comparison with Current Hate Speech Target Identification

Reference	Model + Feature	Accuracy (%)	F1-score (%)
Romim et al. [11]	SVM + U	–	69.60
	SVM + C	–	72.60
	SVM + U + C	–	73.00
Proposed Approaches	SVM + U	73.40	79.59
	SVM + C	76.23	82.42
	SVM + U + C	80.90	85.66
	Stacking Ensemble Model	**85.78**	**88.38**

Our findings demonstrate that the stacking ensemble model, integrating multiple feature representations (TF-IDF, word-level unigrams, and character-level n-grams), outperforms traditional machine learning techniques and deep learning methods in hate speech detection. This stacking model demonstrated improved F1-scores and classification accuracy across all tasks. Adjusting the dataset distribution further enhanced performance, emphasizing the need to handle class imbalance when detecting hate speech. We have explored and split our dataset in three ways: 60:40, 70:30, and 80:20. We found the best results with the 80:20 ratio. These findings emphasize the potential of ensemble learning combined with feature engineering to enhance the detection, categorization, and target identification of hate speech in social media texts.

5 Conclusion

5.1 Conclusion

In this study, we proposed a comprehensive method for identifying Bengali hate speech through a three-class classification task. The dataset underwent extensive preprocessing, followed by feature extraction and experimentation with multiple machine learning (ML) and deep learning (DL) models. The models explored include Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LR), Multinomial Naïve Bayes (MNB), Decision Tree (DT), Bi-LSTM, and CNN.

To assess performance, we utilized key evaluation metrics such as accuracy, precision, recall, and F1-score. Among all models, the stacking ensemble approach delivered the best results, attaining an accuracy of 91.49% on the test set with the imbalanced dataset. Furthermore, when trained on an oversampled dataset, the ensemble method demonstrated a remarkable accuracy of 94.37% on the test set of the balanced dataset, surpassing all other models. These findings underscore the strength and efficiency of our proposed approach in detecting hate speech in Bengali text.

5.2 Future Work

Future work will refine our stacking ensemble method and explore advanced ensemble and transformer-based models to further enhance hate speech detection accuracy. We also plan to extend our approach for multilingual hate speech detection by incorporating larger, more diverse datasets to improve generalizability and address class imbalance.

Funding: No funding. This research was conducted without any financial support.

References

[1] Subaramaniam, K., Kolandaisamy, R., Jalil, A. B., & Kolandaisamy, I. (2022). Cyberbullying challenges on society: a review. Journal of positive school psychology, 6(2), 2174-2184.

[2] Giumetti, G. W., & Kowalski, R. M. (2022). Cyberbullying via social media and well-being. Current opinion in psychology, 45, 101314.

[3] Saksesi, A. S., Nasrun, M., & Setianingsih, C. (2018, December). Analysis text of hate speech detection using recurrent neural network. In 2018 international conference on control, electronics, renewable energy and communications (ICCEREC) (pp. 242-248). IEEE.

[4] Mondal, M., Silva, L. A., & Benevenuto, F. (2017, July). A measurement study of hate speech in social media. In Proceedings of the 28th ACM conference on hypertext and social media (pp. 85-94).

[5] Ghosal, S., Jain, A., Tayal, D. K., Menon, V. G., & Kumar, A. (2023). Inculcating context for emoji powered Bengali hate speech detection using extended fuzzy SVM and text embedding models. ACM transactions on Asian and low-resource language information processing.

[6] Al-Makhadmeh, Z., & Tolba, A. (2020). Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach. Computing, 102(2), 501-522.

[7] WeAre Social and Hootsuite, “2018 Digital Yearbook,” 2018, [Online].Available: https://digitalreport.wearesocial.com/.

[8] Ethnologue, “List of Languages by Number of Native Speakers,” 2020.

[9] Islam, J., Mubassira, M., Islam, M. R., & Das, A. K. (2019, February). A speech recognition system for Bengali language using recurrent neural network. In 2019 IEEE 4th international conference on computer and communication systems (ICCCS) (pp. 73-76). IEEE.

[10] Mumu, T. F., Munni, I. J., & Das, A. K. (2021). Depressed people detection from bangla social media status using lstm and cnn approach. Journal of Engineering Advancements, 2(01), 41-47.

[11] Romim, N., Ahmed, M., Islam, M. S., Sharma, A. S., Talukder, H., & Amin, M. R. (2022). Bd-shs: A benchmark dataset for learning to detect online bangla hate speech in different social contexts. arXiv preprint arXiv:2206.00372.

[12] Nugroho, K., Noersasongko, E., Fanani, A. Z., & Basuki, R. S. (2019, July). Improving random forest method to detect hatespeech and offensive word. In 2019 International Conference on Information and Communications Technology (ICOIACT) (pp. 514-518). IEEE.

[13] Karim, M. R., Chakravarthi, B. R., McCrae, J. P., & Cochez, M. (2020, October). Classification benchmarks for under-resourced bengali language based on multichannel convolutional-lstm network. In 2020 IEEE 7th international conference on Data Science and Advanced Analytics (DSAA) (pp. 390-399). IEEE.

[14] Abro, S., Shaikh, S., Khand, Z. H., Zafar, A., Khan, S., & Mujtaba, G. (2020). Automatic hate speech detection using machine learning: A comparative study. International Journal of Advanced Computer Science and Applications, 11(8).

[15] Islam, M., Hossain, M. S., & Akhter, N. (2022, May). Hate speech detection using machine learning in Bengali languages. In 2022 6th International conference on intelligent computing and control systems (ICICCS) (pp. 1349-1354). IEEE.

[16] Das, A. K., Al Asif, A., Paul, A., & Hossain, M. N. (2021). Bangla hate speech detection on social media using attention-based recurrent neural network. Journal of Intelligent Systems, 30(1), 578-591.

[17] Romim, N., Ahmed, M., Islam, M. S., Sharma, A. S., Talukder, H., & Amin, M. R. (2021). HS-BAN: A benchmark dataset of social media comments for hate speech detection in bangla. arXiv preprint arXiv:2112.01902.

[18] Emon, E. A., Rahman, S., Banarjee, J., Das, A. K., & Mittra, T. (2019, June). A deep learning approach to detect abusive bengali text. In 2019 7th International Conference on Smart Computing & Communications (ICSCC) (pp. 1-5). IEEE.

[19] Mnassri, K., Rajapaksha, P., Farahbakhsh, R., & Crespi, N. (2022, December). BERT-based ensemble approaches for hate speech detection. In GLOBECOM 2022-2022 IEEE Global Communications Conference (pp. 4649-4654). IEEE.

[20] Mridha, M. F., Wadud, M. A. H., Hamid, M. A., Monowar, M. M., Abdullah-Al-Wadud, M., & Alamri, A. (2021). L-boost: Identifying offensive texts from social media post in bengali. Ieee Access, 9, 164681-164699.

[21] Lauron, M. L. C., & Pabico, J. P. (2016). Improved sampling techniques for learning an imbalanced data set. arXiv preprint arXiv:1601.04756.

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

[23] Puvar, P., Patel, N., Shah, A., Solanki, R., & Rana, D. (2021). Heart disease detection using ensemble learning approach. International Research Journal of Engineering and Technology.

[24] Ishmam, A. M., & Sharmin, S. (2019, December). Hateful speech detection in public facebook pages for the bengali language. In 2019 18th IEEE international conference on machine learning and applications (ICMLA) (pp. 555-560). IEEE.

[25] Roy, P. K., Tripathy, A. K., Das, T. K., & Gao, X. Z. (2020). A framework for hate speech detection using deep convolutional neural network. IEEE Access, 8, 204951-204962.