ISSN 2007-9737

Cyberbullying Detection in a Multi-classification Codemixed Dataset

Sahinur Rahman-Laskar[1,*], Gauri Gupta[1], Ritika Badhani[1], David Eduardo Pinto-Avendaño[2]

[1] UPES, School of Computer Science,
India

[2] Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
Mexico

{sahinurlaskar.nits, gauri17gupta, ritika.badhani12, davideduardopinto}@gmail.com

Abstract. In an era characterized by digital communication and social media, the concept of cyberbullying has arisen as a social concern, impacting individuals of all ages. It refers to the act of using digital communication tools like, social media, and messaging apps, to harass intimidate or harm someone. Codemixed cyberbullying refers to the use of multiple languages or a mix of languages in online communications and the use of multiple languages or a mix of languages can sometimes make it challenging for content moderators or automated systems to detect and address cyberbullying effectively. The challenges include the availability of standard codemixed datasets, especially for Indian languages. This paper investigates cyberbullying detection in Hinglish, a code-mixed language of Hindi and English. We have created a novel multi-class Hinglish dataset, annotated across seven cyberbullying categories: age, gender, religion, mockery, abusive, offensive, and not cyberbullying, and explored different machine learning-based models. We have performed a comparative analysis based on the standard evaluation metrics and achieved a state-of-the-art result on a multi-class codemixed Hinglish dataset.

Keywords. Cyberbullying, codemixed, Hinglish, machine learning.

1 Introduction

Cyberbullying has emerged as a pressing concern in the digital era, particularly on online communication platforms.

It manifests in various forms, such as threats, hate speech, and harassment, inflicting detrimental effects on victims [28]. Identifying and addressing cyberbullying is imperative for creating a secure and inclusive online environment for all users. Detecting cyberbullying involves locating instances of the phenomenon in online content, such as messages, comments, and social media posts. Natural language processing, sentiment analysis, and artificial intelligence techniques offer promising approaches for this task [20].

The primary objectives are to accurately identify cyberbullying incidents and classify them as threats, hate speech, and harassment. Machine learning models have demonstrated remarkable potential in cyberbullying detection by discerning patterns from large datasets, a task that can be challenging for humans. The initial crucial step in developing a cyberbullying detection system is acquiring a comprehensive and diverse dataset of online content, including labeled instances of various cyberbullying forms.

This dataset should encompass a representative sample of threats, hate speech, harassment, and other cyberbullying manifestations. Data preprocessing follows, involving cleaning, deduplicating, and formatting the data into a machine-learning-compatible format.

Relevant features, such as part-of-speech tagging, sentiment scores, and word frequencies, are then extracted. Subsequently, machine learning algorithms are employed to build and train models on the preprocessed data and extracted

doi: 10.13053/CyS-28-3-4989

Table 1. Data statistics of train and test set

Split	No. of Comments
Train	7400
Test	1000
Total	8400

features. The models' performance is evaluated using a separate dataset of online content, considering metrics such as recall, accuracy, and F1-score.

Detecting cyberbullying in code-mixed languages like Hinglish poses additional challenges due to the potential ambiguity in meaning [25]. However, developing models capable of accurately identifying cyberbullying in multilingual and code-mixed contexts is crucial for promoting a safer and more inclusive online environment.

Identifying cyberbullying is a critical step towards providing support and resources to victims and perpetrators alike. Such resources may include counseling services, online forums, and educational materials aimed at mitigating the detrimental effects of cyberbullying and fostering a more positive online experience.

In summary, detecting cyberbullying through machine learning techniques is an essential endeavor in creating a secure and welcoming cyberspace. Developing accurate and efficient models can enable the timely identification of cyberbullying incidents [5]. Providing support and resources to victims is paramount to alleviating the harmful impacts of cyberbullying and promoting a more conducive online environment. The key contributions to this work are as follows:

- Created a multi-classification Hinglish code-mixed dataset, namely, MC-Hinglish1.0 which will be publicly available[1].
- Performed a comparative analysis by exploring different machine learning models for cyberbullying detection on the developed Hinglish code-mixed dataset.
- Proposed the use of an ensemble model and achieved state-of-the-art results for cyberbullying detection on the multi-classification Hinglish code-mixed dataset.

The rest of the paper is organized as follows. In Section 2, existing works related to cyberbullying and the relevant work on Hinglish cyberbullying detection are discussed briefly. The preparation of the dataset MC-Hinglish1.0 is described in Section 3. The ensemble model to combine the different machine learning models, namely, Support Vector Machines (SVM), Random Forest, Logistic Regression, Multinomial Naive Bayes, and XGBoost, are presented in Section 4. Section 5 presents experimental results and Analysis and Section 6, concludes the paper with future work.

2 Related Work

Fuzzy logic and multinomial Naive Bayes classification are suggested in [3] as ways to identify various forms of cyberbullying in Facebook comments. Before acquiring characteristics like adjective and noun frequency, preprocessing procedures like tokenization and stopword elimination are used. The classifier recognizes various forms of bullying, including shame, racism, and sexual harassment.

Bullying severity is assessed by fuzzy rules based on age and the number of expletives. The model outperforms an SVM technique, achieving 88.76% accuracy on a benchmark dataset. In order to detect cyberbullying across languages, the paper shows how to modify natural language processing (NLP) approaches such as Naive Bayes. Bullying severity nuances are captured by fuzzy logic. Both culture detection and model accuracy can be enhanced by additional work.

The research in [9] looks into ways to make it easier to spot instances of cyberbullying on social media sites like Twitter. The research expands on what is already known about the characteristics, trends, and detection strategies of cyberbullying.

The first section of the study discusses the pervasiveness of cyberbullying and emphasizes its negative impacts, especially on vulnerable populations like adolescents. It highlights the

[1] github.com/sahinurlaskar/MC-Hinglish1.0

Table 2. Class level data statistics

Classes	Age	Gender	Religion	Offensive	Mockery	Abusive	Not-Cyberbullying
Train	977	1004	979	979	980	989	1491
	143	141	127	128	135	115	211
	1120	1145	1106	1107	1115	1104	1702
Test	129	141	131	124	127	148	200

significance of social media sites like Twitter, which help incidences of cyberbullying spread quickly. The researchers use a variety of data, such as thoughts, emotions, and Twitter-specific traits, to enhance detection.

User personalities have been established on the Big Five and Dark Triad models. The process of extracting pertinent features, including text/content, users, and information about networks, from Twitter data is described in the paper. It explores the relationship between Dark Triad constructions and the Big Five personality traits and talks about how these characteristics might be used to spot cyberbullying activities.

To comprehend the subtleties of the discourse around cyberbullying, the research also investigates the application of emotion and sentiment analysis. It draws attention to the shortcomings of the methods used now and makes suggestions for how to get around them. In summary, by combining innovative features and utilizing cutting-edge analytical methods, our work advances the creation of reliable and precise cyberbullying detection systems.

It hopes to accomplish this by offering insightful information on the intricate dynamics of bullying and assisting in the development of safer online spaces. The crucial issue of identifying cyberbullying in the Indian languages of Marathi and Hindi is the subject of study in [24]. Cyberbullying, or the act of repeatedly injuring victims via the use of digital media, has grown to be a significant social problem. The authors give background information on the emergence of cyberbullying, the harm it causes to victims' physical and emotional health, particularly young people, and the necessity of automatic detection systems. They claim that despite the large number of non-English-speaking users on social media, the majority of detection research focuses on English material.

According to the authors' survey of related work, the majority of machine learning research on textual features used for cyberbullying identification is conducted for English language users. Naive Bayes, SVM, decision tree models, and logistic regression are among the approaches used to conduct promising research in other languages, such as Arabic and Turkish. Sentiment analysis, network patterns, and keywords are important characteristics. BullyBlocker is a technology that recognizes and notifies parents of cyberbullying on Facebook.

The authors [24] gather datasets in Hindi and Marathi from a variety of sources, including newspapers, reviews, and social media. Bullying is manually classified in the data. Synthetic oversampling is employed to balance classrooms because bullying accounts for just 9% of texts. Bag-of-words feature extraction is carried out following preprocessing.

A split of the data into 80/20 is used for training and testing various models, such as logistic regression, stochastic gradient descent, and multinomial Naive Bayes. Based on synthetic data, experiments indicate that logistic regression performs best, achieving up to 97% accuracy and 96% F1 score. This method works for all languages and domains. Larger datasets will be used in future research, along with linguistically specific sentiment analysis, comparisons with NLP techniques, and real-time social media integration.

The study in paper [22] suggested utilizing classifiers like Naive Bayes and Random Forest with psychological variables like sentiment and personality of Twitter users to increase

Table 3. Class level data statistics

Class Name	Example
AGE	Behan meri 40 yr maid looks better than you
GENDER	Electric scotty chalane vale ladke gay hote h
RELIGION	Na pakistani pta paye na Indian, na angrez
MOCKERY	Arey bc ye icon kya hai yaar, tujhe itna bhi nahi pata
OFFENSIVE	Dettol ke add m kitanu ka role krogi
ABUSIVE	Bc mene teko call bhi kiya tha
NOT_CYBERBULLYING	Nai na, bahut garmi ho raha hai

cyberbullying detection accuracy to 91.88%. However, this strategy is limited by the need for sentiment data.

By combining contextual embeddings from approaches such as BERT and VecMap, [18] created a multitask framework that achieved over 80% accuracy in sentiment analysis and cyberbullying detection. More study is necessary because cyberbullying is multimodal. Numerous research works have explicitly addressed the Bangla/Bengali language.

Using CNN models, [2] obtained 84% and 80% accuracy for the Bangla and Romanized Bangla datasets. With just 1,339 comments, [1] created SVM and Naive Bayes models that achieved up to 72% accuracy. [13] discovered a maximum accuracy of 78.1% when investigating several ML algorithms using TF-IDF characteristics.

[27] compared LSTM and CNN to ML models like SVM and Naive Bayes using data from Bangla YouTube comments. The accuracy of the LSTM using word embeddings was 65–67%. In order to identify abusive Bangla comments in Facebook postings based on text and emojis, [11] examined MNB, SVM, and CNN-LSTM models; SVM proved to be the most successful with 78% accuracy. Larger datasets and more intricate deep learning architectures require further investigation.

With the help of a dataset of 12,282 social media comments, this study creates a Bi-LSTM model with two thick layers for the identification of cyberbullying in Bangla. Model training precedes the application of preprocessing techniques like tokenization, stemming, and padding sequences.

The Adam optimizer yields a higher testing accuracy of 95.08% when compared to the SGD optimizer's performance. 94.31% average accuracy is achieved with five-fold cross-validation. A benefit over earlier work is a bigger dataset and less feature engineering. Techniques like attention processes can still be used to improve accuracy.

The research in [8] provides a sentiment analysis approach to identify cyberbullying in tweets by utilizing machine learning techniques. The authors gather a corpus of tweets and manually label them as neutral, negative, or positive comments.

The tweets undergo several preparation procedures, such as cleaning, tokenization, removal of stop words, normalization, named-entity recognition, and stemming. The preprocessed tweets are used to extract N-gram characteristics ranging from 2 to 4 grams. Chi-square feature selection is used, and information acquired is utilized to eliminate features that aren't important.

Accuracy, precision, recall, F1-score, and ROC curve are used to evaluate the Naive Bayes (NB) and Support Vector Machine (SVM) classifiers after they have been trained on the features. Tests reveal that SVM beats NB for every n-gram model, with 4-grams exhibiting the greatest results.

The suggested model results in a somewhat better detection of cyberbullying than earlier research, with SVM achieving 91.64% accuracy and 88.93% ROC as opposed to 83.46% and

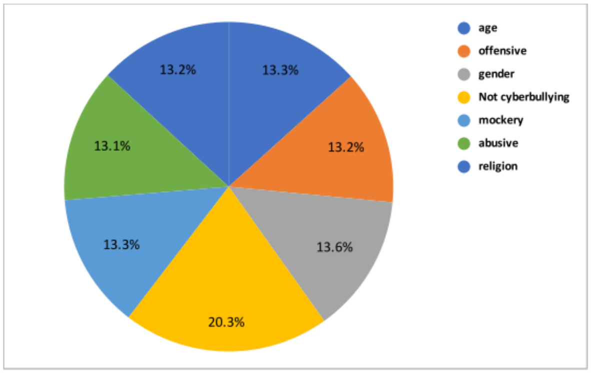

Fig. 1. Train data distribution

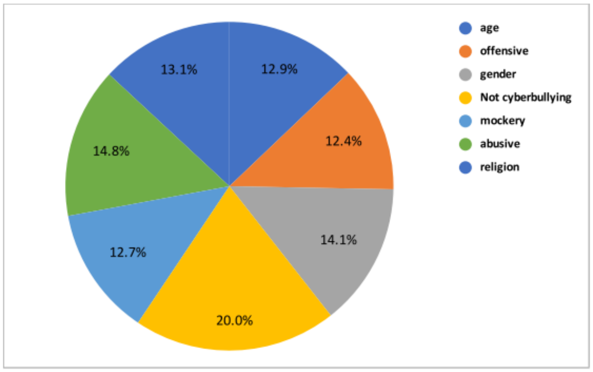

Fig. 2. Test data distribution

85.71% from earlier research. Overall, the sentiment analysis model outperforms both NB and earlier methods in the efficient real-time identification of cyberbullying tweets. It uses an SVM classifier with 4-gram features.

Benchmark research comparing multiple neural networks and machine learning models for the binary categorization of cyberbullying in social media texts is presented in this publication [29]. The authors make use of seven real-world datasets that are currently available from various social media sites, such as Twitter, Instagram, Vine, Ask.fm, Formspring, and Ask.fm. Following preprocessing and data cleaning, over $390,000$ sentences were classified as either non-cyberbullying or cyberbullying in the combined datasets. The authors utilize random oversampling to correct for class imbalance in the datasets. Four models are compared: a Support Vector Machine (SVM) baseline, a BiLSTM network, and tuned versions of the BERT and HateBERT transformer models. For two to four training epochs, hyperparameters are adjusted. After being trained on combined datasets from several platforms, the models are assessed on holdout test sets from each platform.

The development of cyberbullying detection models that are indifferent to platforms is initiated by this cross-platform assessment. Based on test sets that are matched, the optimized HateBERT model outperforms the others overall, with F1 scores as high as 0.81 and as low as 0.69–0.76 on sets that are mismatched. The maximal F1 scores of the SVM and BiLSTM models are approximately 0.70. Applying models across platforms significantly reduces performance. The authors draw the conclusion that HateBERT has potential for widespread cyberbullying detection.

Seven machine learning classifiers for cyberbullying detection on Twitter are compared in the work [21]. A dataset including 37,373 tweets—70% for training and 30% for prediction—is used in the study. The usage of TF-IDF and Word2Vec algorithms, as well as the significance of pre-processing and feature extraction in cyberbullying detection, are covered in the paper. The study contrasts the effectiveness of several classifiers, including Support Vector Machine, Random Forest, and Naive Bayes. using an accuracy of 95.5%, the SVM classifier using a linear kernel outperforms the others, according to the results. The study comes to the conclusion that machine learning methods can be useful in identifying harassment on social media sites, and the suggested model can serve as a foundation for further studies in this field.

2.1 Existing Work on Hinglish Codemixed Dataset

In order to detect offensive content in tweets in paper [17] investigate the use of a variety of classifiers, including Naive Bayes and Support Vector Machine. It also explores the stages of feature extraction and preprocessing, emphasizing how crucial it is to extract pertinent features and clean text in order to achieve precise classification.

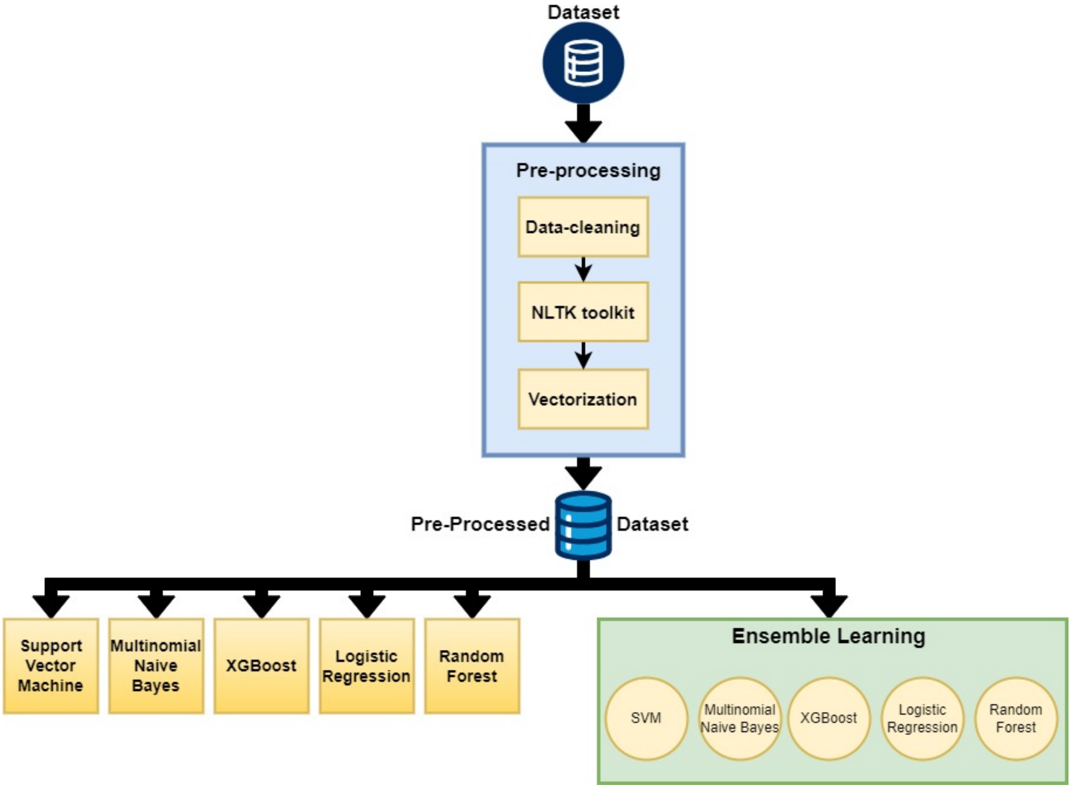

Fig. 3. Methodology

The work uses a data set of 37,373 tweets to assess the effectiveness of several classifiers and methods for extracting features, and in the end, it suggests a model for social media platform cyberbullying detection.

BullyExplain is a unique benchmark dataset for explainable cyberbullying detection in code-mixed language, and it is presented in the publication [16]. The multitask issue is reframed as a text-to-text creation task using the authors' novel unified generative framework, GenEx. Applied to the BullyExplain dataset, the suggested method outperforms other baseline models and existing state-of-the-art methods in a number of evaluation measures.

The study draws attention to the frequency of code-mixing in cyberbullying as well as the necessity of offering explanations for machine learning decisions. There are 6,084 samples in the dataset overall; 3,034 of them are classified as non-bully, and the remaining 3,050 samples are classified as bullies. A commonsense-aware unified generative framework called GenEx is also introduced by the authors. It uses commonsense information to improve the context and richness of tweets that are usually concise and to the point.

The study in [12] addresses the challenge of detecting cyberbullying in Hinglish, a code-mixed language combining Hindi and English. They created a new dataset called ”CMDL-Cyberbullying” with 20,000 manually annotated Hinglish comments from social media platforms. Using this dataset, they experimented with various machine learning models like logistic regression, naive Bayes, decision trees, random forest, and support vector machines (SVMs) for cyberbullying classification. Different features such as n-grams, TF-IDF, and pre-trained word embeddings were evaluated. The SVM model with character n-grams and TF-IDF features performed

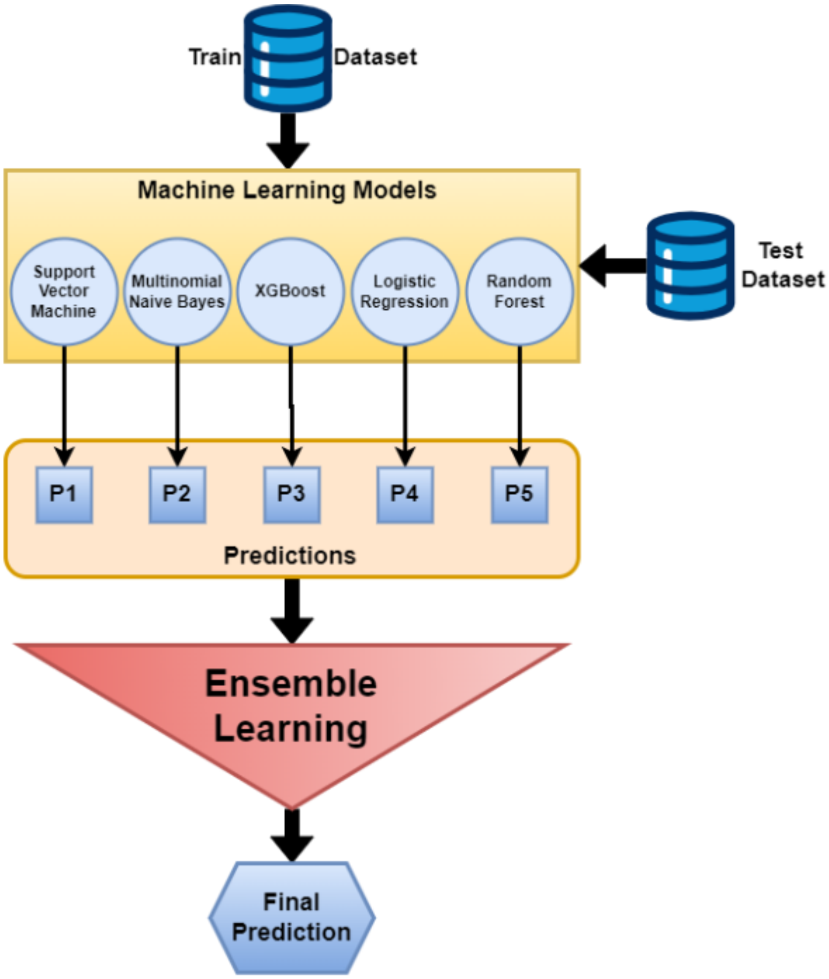

Fig. 4. Ensemble learning model

best, achieving an F1-score of 0.78. However, the presence of English words, phonetic typing of Hindi, and Romanized Hindi posed challenges.

The authors highlight the importance of detecting cyberbullying in code-mixed languages and the need for further research to improve model performance in such scenarios. The paper [19] proposes a novel approach to detect cyberbullying in Hinglish (Hindi-English code-mixed) text by leveraging emojis, sentiment analysis, and emotion detection. The authors created a new dataset called ”HingC” containing 16,988 Hinglish comments annotated for cyberbullying, sentiment, emotions, and emojis.

They developed an ensemble model that combines features derived from emojis, sentiment scores, emotion scores, and text representations. For text representation, they used multilingual BERT and contextual string embeddings from the Flair library. Emoji representations were obtained from the DeepMoji model. The sentiment and emotion analysis components utilized transfer learning from English to Hinglish.

The ensemble model, which combined all these features, achieved state-of-the-art performance on the HingC dataset with an F1 score of 0.81 for cyberbullying detection. The authors demonstrated that incorporating emojis, sentiment, and emotion

Table 4. Experimental result

	Logistic Regression	Random Forest	XGBoost	Naive Bayes	SVM	Voting(Hybrid)
Accuracy	0.580	0.570	0.602	0.512	0.564	0.609
Recall	0.5643	0.5619	0.5908	0.4963	0.5524	0.5940
F1-score	0.5540	0.5585	0.5804	0.4731	0.5475	0.5879

information significantly improved the cyberbullying detection performance compared to using only text features. The paper highlights the importance of these multimodal signals for understanding the nuances of code-mixed languages like Hinglish in cyberbullying contexts.

This paper [10] focuses on developing an efficient model to detect the presence of Hinglish text (a code-mixed language of Hindi and English) in YouTube data, such as video titles, descriptions, and comments. The authors highlight the importance of identifying Hinglish text for tasks like content moderation, targeted advertising, and understanding user preferences. The proposed model uses a combination of rule-based and machine learning approaches.

First, a set of rules are applied to filter out text that is purely in English or Hindi. For the remaining text, which potentially contains Hinglish, character-level and word-level features are extracted. These features include n-grams, distribution of Hindi and English characters, presence of named entities, and part-of-speech tags.

The extracted features are then fed into several machine learning classifiers, including Logistic Regression, Support Vector Machines (SVMs), and ensemble methods like Random Forest and XGBoost. The models are trained and evaluated on a manually annotated dataset of YouTube data. The authors report that their best performing model, an ensemble of XGBoost and Logistic Regression, achieved an F1-score of 0.93 in detecting the presence of Hinglish text.

They also conducted experiments to analyze the impact of different features and found that character-level n-grams and the distribution of Hindi and English characters were the most important features for this task. The paper [17] proposes a novel generative approach for explainable cyberbullying detection in Hinglish (Hindi-English code-mixed) text.

The authors use a variational autoencoder (VAE) model to learn disentangled latent representations that capture different attributes like cyberbullying, sentiment, and emotions present in the input text. During inference, they generate counterfactual samples by manipulating these latent representations to analyze how changes in attributes like sentiment impact the cyberbullying probability. This provides explainable insights into the model's predictions.

On their curated HingC dataset, their VAE-based model achieved state-of-the-art F1 score of 0.82 for cyberbullying detection, while also enabling explainable AI capabilities lacking in previous black-box methods. The authors demonstrate how their approach can highlight words/phrases responsible for cyberbullying flags and generate revised non-cyberbullying versions through latent edits, aiding transparency and safer content generation.

However, it is evident from the reviewed literature that there is a significant lack of research focusing on the development and utilization of multi-classified Hinglish datasets, which is a unique blend of Hindi and English languages widely used in various online platforms and social media. This research gap underscores the need for further exploration and construction of robust multi-classified Hinglish datasets to facilitate more effective and accurate cyberbullying detection models tailored to the diverse linguistic landscape of the Indian subcontinent.

3 Dataset Creation: MC-Hinglish 1.0:

The dataset utilized in this paper was created to facilitate the task of cyberbullying detection in the Hinglish language. Initially, a binary classified

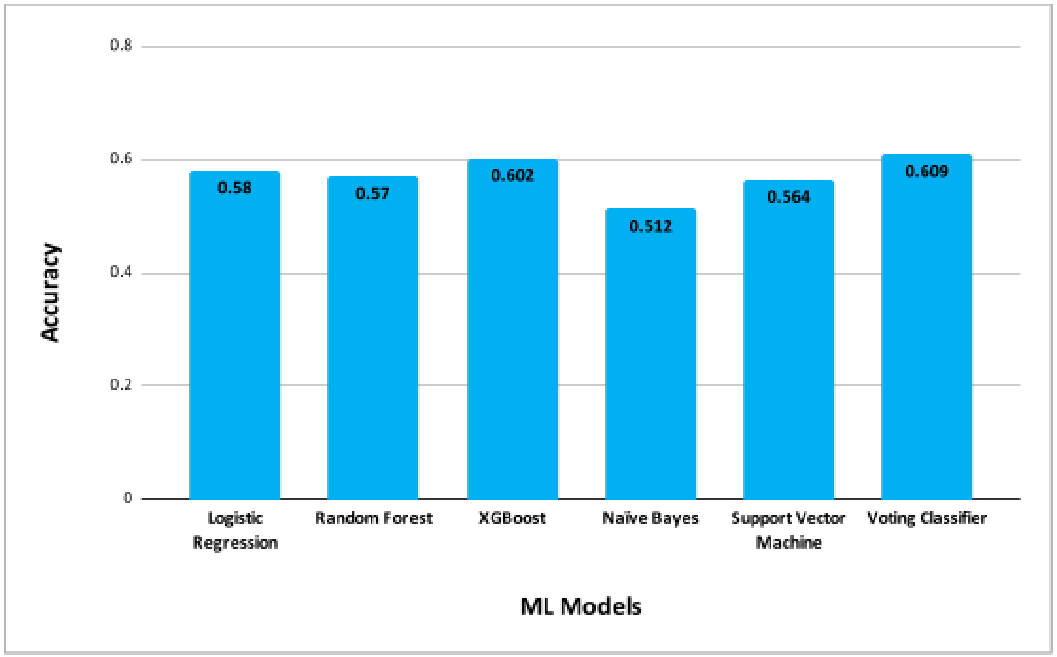

Fig. 5. Accuracy graph

dataset was obtained from GitHub[2], consisting of comments labeled as either cyberbullying or non-cyberbullying.

Subsequently, this dataset was manually annotated and modified to transform into a multi-classification dataset with seven distinct categories: age, gender, religion, mockery, abusive, offensive, and not-cyberbullying. The annotation process aimed to provide a more granular understanding of the types of cyberbullying present in the dataset.

3.1 Data Collection

The original binary dataset was sourced from the aforementioned GitHub repository. This dataset contained a collection of comments written in Hinglish language, spanning Twitter.

Prior to annotation, basic preprocessing steps are applied to clean the text data and remove unwanted symbols.

3.2 Data Statistics

MC-Hinglish1.0 dataset comprises a total of 8400 annotated comments, with 7400 comments allocated for the training set, and 1000 comments reserved for the testing set. The dataset (MC-Hinglish 1.0) was split into training, and test sets in the ratio of $88:12$.

Care was taken to maintain the class distribution across the splits to prevent any bias in model training and testing (as shown in Fig. 1, 2). The data statistics are presented in Table 1.

3.3 Class Description

The manual annotation process involved categorizing each comment into one of the seven predefined classes based on its content. The manually annotated dataset (MC-Hinglish1.0) is categorized into 7 classes: age, gender, religion, mockery, abusive, offensive, and not-cyberbullying. This annotation process took approximately 60 days of rigorous effort and ensured no duplicate comments. Each comment is labeled according

[2] http://surl.li/trwil

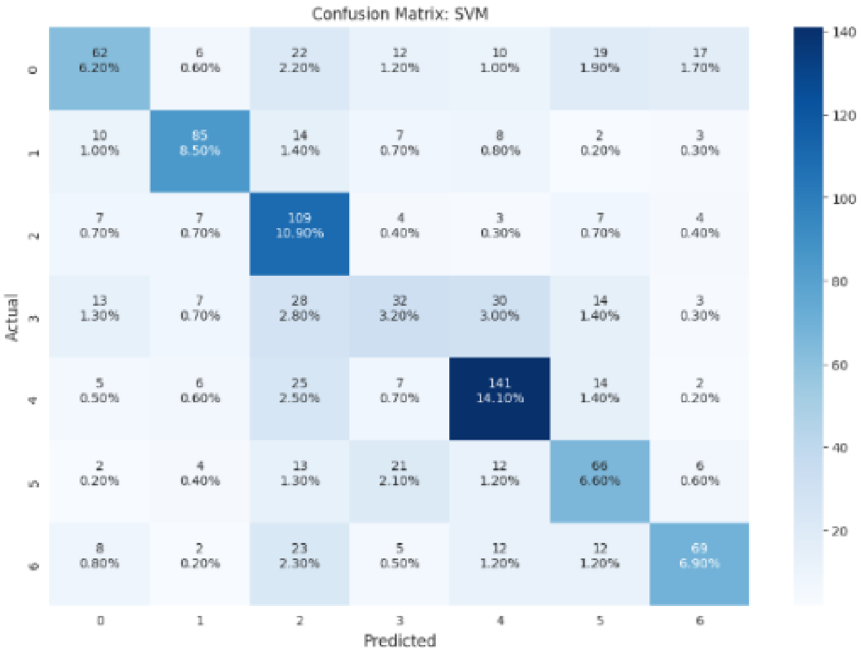

Fig. 6. Confusion matrix for SVM model

to its predominant theme or content which is as shown below:

- **Age:** Text related to age-based discrimination or stereotypes.
- **Gender:** Text containing gender-based biases or stereotypes.
- **Religion:** Text pertaining to religious affiliations or beliefs.
- **Mockery:** Text involving mocking or ridiculing individuals or groups.
- **Offensive:** Text with offensive language or content.
- **Abusive:** Text containing abusive or derogatory language.
- **Not-Cyberbullying:** Text not classified as cyberbullying based on the defined criteria.

Table 3 presents examples of sentences from each class category and Table 2 presents each class level data statistics.

4 Methodology

The following section presents a comprehensive overview of various techniques that are utilized for the prediction of cyberbullying on the MC-Hinglish1.0 dataset. We have used five machine learning algorithms, namely Support Vector Machines (SVM), Random Forest, Logistic Regression, Multinomial Naive Bayes, and XGBoost, to improve the prediction's accuracy. These algorithms are commonly employed in machine learning and have proven to be highly effective in earlier research.

Furthermore, we have applied an ensemble or hybrid model to combine five different machine learning models to enhance the model's performance (as shown in Fig. 3). By combining the predictions of several models, ensemble methods are known to improve the predictive ability of machine learning models. All things considered, putting these strategies to use can help reduce the harm that cyberbullying on online platforms causes.

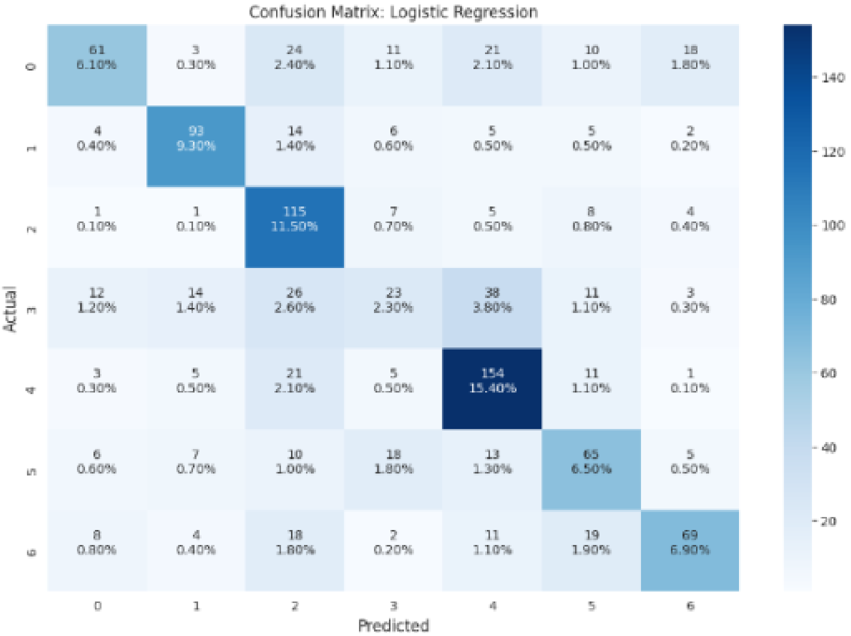

Fig. 7. Confusion matrix for logistic regression model

4.1 Machine Learning Models

SVM: This kind of binary linear classification model analyzes and classifies data by identifying patterns in it through the use of a learning algorithm. Finding decision boundaries and classifying a dataset are the primary goals of support vector machines (SVM) [7].

SVM is renowned for its accuracy even though its training time can be lengthy. SVM looks for the decision boundary that maximizes the difference in distance between the classes. SVM classifiers usually have the parameter values C=0.1 and kernel=rbf set.

Finding a hyperplane with the largest possible margin between the classes is the goal of the SVM equation. Support vectors, a subset of the training data that consists of the points that are closest to the decision boundary, are what determine the hyperplane. New data points can be accurately classified by SVM by optimizing the margin between the support vectors on either side of the hyperplane.

SVM is a learning algorithm-based binary linear classification model that is used to analyze, classify, and identify patterns in data [7]. SVM is used to identify decision boundaries and divide datasets into classes; despite its sometimes sluggish training period, SVM is an accurate algorithm:

$$\overrightarrow{W} = \sum_j \alpha_j c_j \overrightarrow{d}_j, \alpha_j \geq 0, \tag{1}$$

where the data's polarity (positive and negative) is represented by vector $\mathbf{w}$, which is a hyperplane $\in \{-1, 1\}$. By solving the dual optimization problem, α_j are obtained [7].

The only document vectors contributing are those whose value is greater than zero, and they are referred to as support vectors.

The process of classifying test instances involves identifying the side of the hyperplane on which they land.

Logistic Regression: It is a kind of regression model that is frequently applied to problems of

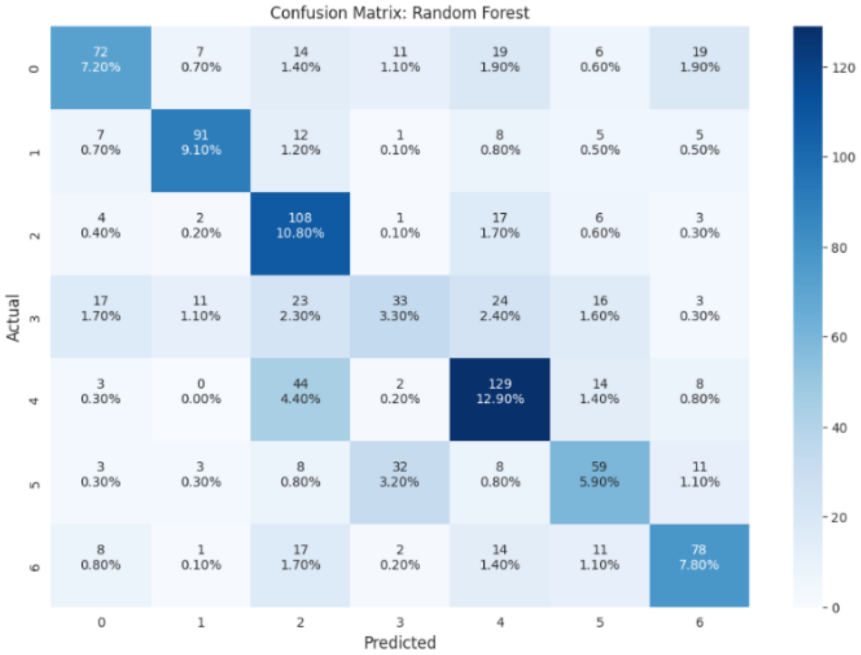

Fig. 8. Confusion matrix for random forest model

classification. It is especially helpful in scenarios where we have to forecast a binary result, meaning that the dependent variable can only have two possible values: 0 or 1. One or more independent variables and one categorical dependent variable are usually related using this model [4]. Selecting a hyper-plane that maximizes the gap between the classes' separation is the aim of logistic regression.

That is to say, it fits a line that is as accurate as possible in differentiating between the two classes. The recommended value for the LR classifiers' parameter values is $C = 3$, which is how this is accomplished [4]. This way, new data points are classified by the model in a way that allows it to predict an outcome's probability with accuracy. The output of the linear equation is converted into a probability value between 0 and 1 by the logistic regression model using a sigmoid function.

It is simpler to understand the output of the linear equation as a probability because the sigmoid function ”squashes” it into a range between 0 and 1 [4]. Based on a threshold value, the model can categorize the data point into one of the two classes after obtaining the probability.

Random Forest Classifier: For tasks involving classification, this well-liked machine learning algorithm is frequently employed. The model's accuracy and robustness are increased by utilizing an ensemble approach that combines a number of decision tree classifiers [7]. The Random Forest classifier, to put it simply, uses the training dataset to construct several decision trees, then gathers votes from each tree to determine the final label or class of the test object.

To lessen the chance of overfitting to the training data, each decision tree in the forest is trained using a random subset of both the features and the training data [7]. With a parameter value of n estimators = 40, the Random Forest classifier has been tuned for this particular scenario.

This indicates that 40 decision trees, each with a random subset of the training data and features, are constructed by the classifier using the training set. The Random Forest classifier can produce a more reliable and accurate classification model

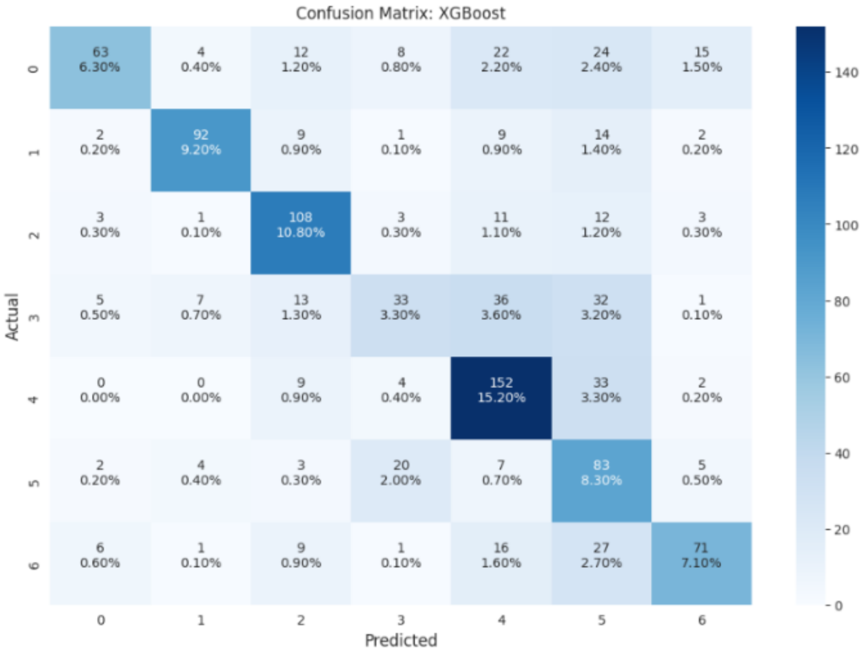

Fig. 9. Confusion matrix for XGBoost model

by integrating the predictions of these separate decision trees.

Multinomial Naive Bayes: MultinomialNB, part of machine learning's Naive Bayes family, is tailored for classification tasks involving multiple classes and features, assuming features' distributions follow a multinomial distribution. It computes probabilities for each class given input features using Bayes' theorem, making it particularly effective for text classification, such as sentiment analysis or document categorization [23]. Despite its ”naive” assumption of feature independence, MultinomialNB often yields strong performance, especially with high-dimensional data like word counts. Its simplicity, efficiency, and ability to handle sparse data make it a popular choice for many text-based machine learning applications.

XGBoost Classifier: The XGBoost (Extreme Gradient Boosting) classifier [14] is a powerful machine learning algorithm known for its effectiveness in classification tasks. With the provided parameters—n-estimators set to 500, max-depth at 6, and learning-rate set to 0.1—the classifier is configured to create a strong ensemble of decision trees.

The parameter n-estimators determine the number of boosting rounds or trees to be generated, in this case, a substantial 500, potentially enhancing the model's predictive capacity. Setting max-depth to 6 controls the maximum depth of each decision tree, preventing overfitting while allowing for sufficient complexity to capture patterns in the data.

A learning-rate of 0.1 dictates the step size during optimization, influencing the rate at which the algorithm adapts to minimize the loss function. This parameter choice is often balanced, as it's neither too aggressive nor too conservative, facilitating a stable learning process [23].

Overall, this configuration aims to strike a balance between model complexity, computational efficiency, and generalization performance, making it a robust choice for classification tasks across various domains.

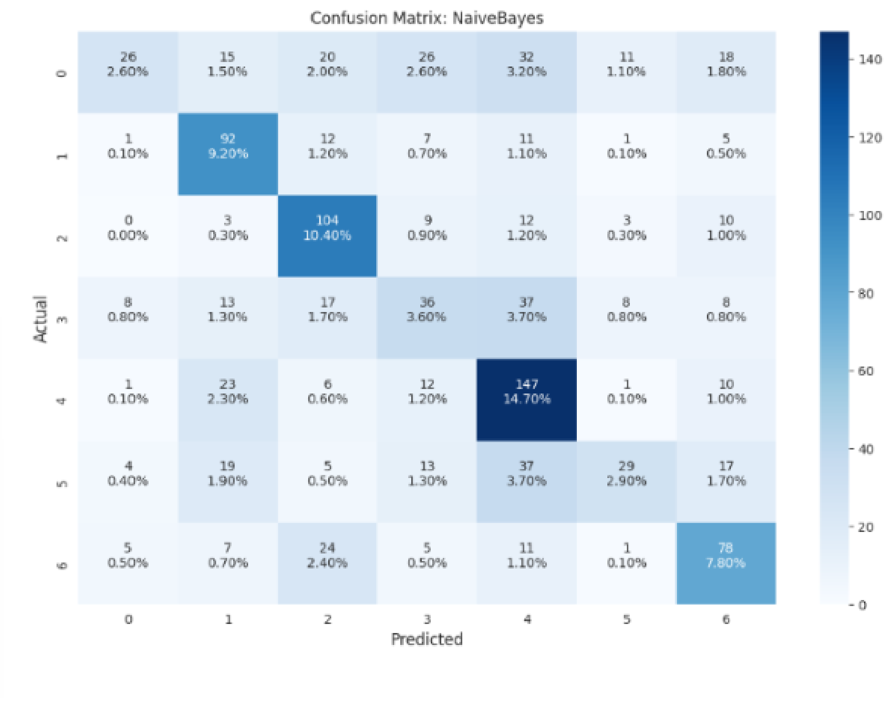

Fig. 10. Confusion matrix for naive bayes model

Ensemble Learning Model: The ensemble learning [6] is used to raise baseline learning techniques' accuracy and performance. SVM, Random Forest, Logistic Regression, Multinomial Naive Bayes, and XGBoost are the base learners utilized in this methodology.

For every base classifier in the ensemble classifier, the labels for age, gender, religion, mockery, abusive behavior, offensiveness, and not-cyberbullying are determined.

Ultimately, by combining the predictions from all of the classifiers to determine the outcome, a soft voting mechanism allows these classifiers to work together to create an ensemble model.

The final prediction is determined by tallying the votes cast by the base learners for the class label (as shown in Fig. 4).

5 Result and Analysis

5.1 Experimental Result

We evaluated the performance of five machine learning models - Multinomial Naive Bayes (MNB), Support Vector Machines (SVM), Random Forest (RF), XGBoost, and Logistic Regression (LR) and then applied an ensemble learning model to enhance the accuracy of cyberbullying detection.

The ensemble learning model has been built using five machine learning models, namely SVM, Multinomial Naive Bayes, XGBoost, Logistic Regression, and Random Forest. To assess the model's performance, we have considered standard evaluation metrics like accuracy, recall, and F1 scores. The experimental results are presented in Table 4. Accuracy measures the proportion of correctly classified instances out of all instances in a given dataset [6]. The text goes on to mention that the mean accuracy across all models is approximately 60.90%.

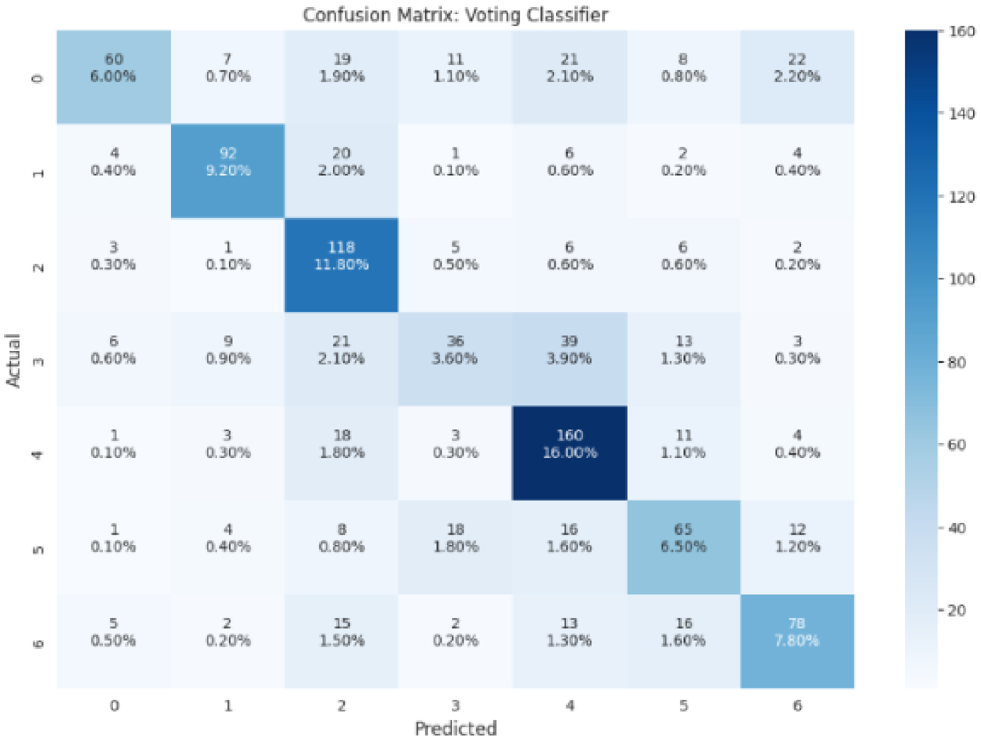

Fig. 11. Confusion matrix for ensemble model (voting classifier)

This means that the model is able to correctly classify around 61.00% of the instances on average. The recall metric, also referred to as sensitivity, assesses the accuracy of the model in identifying true positive instances out of all the actual positive instances [6]. A high recall value indicates that the model is successful in identifying a majority of positive instances.

On average, the recall score stands at approximately 59.40%, which means that the model accurately detects around 59.00% of the actual positive instances. The F1-score is a calculation that combines precision and recall in a balanced way. It provides a fair evaluation of the model's performance by considering both precision and recall. The average F1-score is around 58.79%, which shows that the model's precision and recall are reasonably balanced.

5.2 Analysis

A confusion matrix is a table that is used to evaluate the performance of a classification model on a set of test data for which the true values are known. It shows the number of correct and incorrect predictions made by the classification model, broken down by each class.

The confusion matrix is typically represented as an $N * N$ matrix, where N is the number of classes in the classification problem. Each row in the matrix represents the instances of an actual class, while each column represents the instances of a predicted class [26]. The main entries in the confusion matrix are:

1. **True Positives (TP):** The number of instances that were correctly predicted as belonging to the positive class.

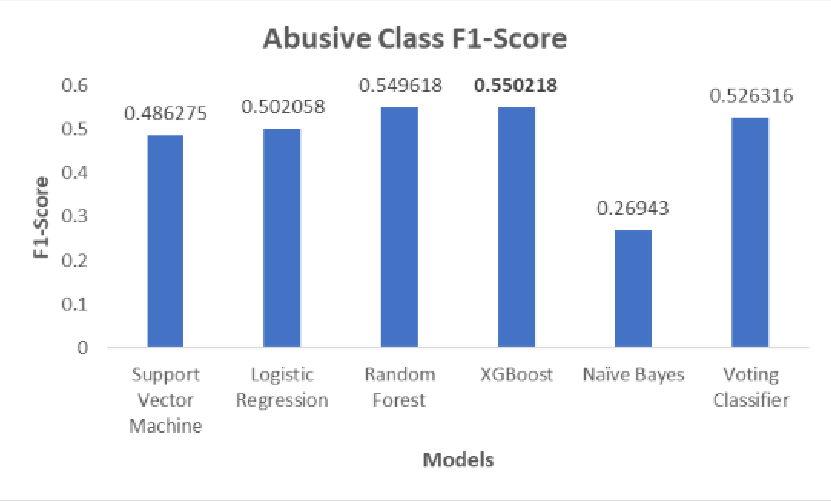

Fig. 12. Comparison of F1-Scores for abusive level classification across ML models

2. **True Negatives (TN):** The number of instances that were correctly predicted as belonging to the negative class.

3. **False Positives (FP):** The number of instances that were incorrectly predicted as belonging to the positive class.

4. **False Negatives (FN):** The number of instances that were incorrectly predicted as belonging to the negative class.

The values in the confusion matrix can be used to calculate various performance metrics for the classification model, such as accuracy, precision, recall, and F1-score [26]. The confusion matrix provides a comprehensive view of the model's performance, allowing for a better understanding of its strengths and weaknesses, and can guide further improvements or adjustments to the model.

The results showcase the performance of various classification models from the confusion matrix on a cyberbullying Twitter dataset with six classes: abusive, age, gender, mockery, not-cyberbullying, offensive, and religion. The Multinomial Naive Bayes (MNB) and Voting models have complete information for calculating overall accuracy and class-wise performance. The Voting model outperforms MNB in terms of overall accuracy, achieving 60.9% compared to 51.2% for MNB. However, upon closer inspection, the models exhibit varying strengths and weaknesses across different classes. For the abusive class, the Voting model correctly identified 80 instances, significantly higher than MNB's 45 true positives, indicating better performance in detecting abusive content.

Conversely, MNB excelled in identifying age-related instances (172 true positives) and mockery cases (108 true positives) compared to the Voting model's 118 and 76 true positives, respectively.

Interestingly, the Voting model demonstrated superior performance in recognizing gender-related instances (219 true positives) and offensive content (121 true positives), while MNB performed better in identifying non-cyberbullying instances (287 true positives) and religion-related cases (146 true positives).

From the confusion matrix, we can observe that the Voting Classifier performed well in identifying instances of ”Age,” ”Gender,” ”Religion,” ”Not-cyberbullying,” and ”Offensive” classes. However, it had some difficulty in distinguishing between ”Mockery” and ”Abusive” classes, as evident from the off-diagonal entries.

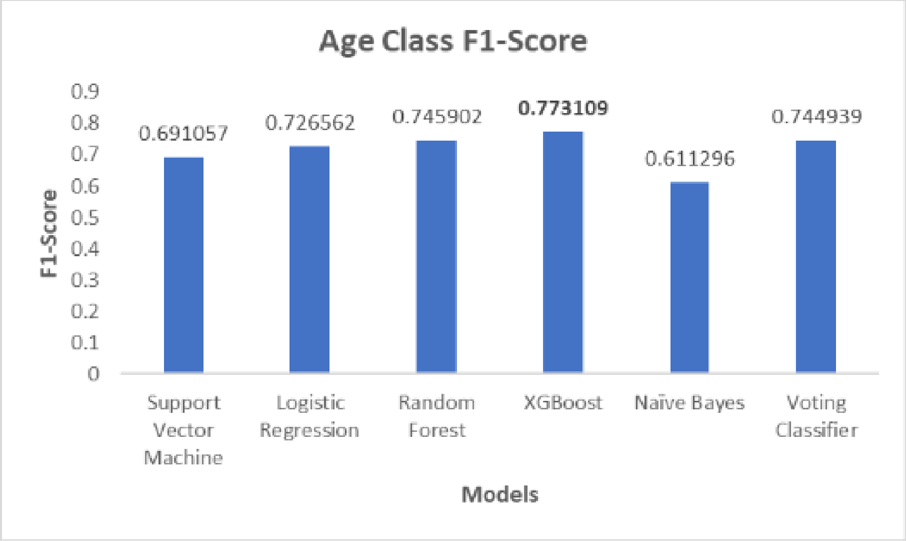

Fig. 13. Comparison of F1-Scores for age level classification across ML models

5.3 Individual Class Level Analysis

The graph shows the F1-scores for classifying abusive content using different models. The XGBoost model achieves the highest F1-score of around 0.55, it means that among all the models compared in the graph, the XGBoost model performed the best in correctly identifying and classifying abusive content.

An F1-score of 0.55 indicates that the XGBoost model strikes a good balance between correctly detecting true positives (abusive content identified as such) and avoiding false positives (non-abusive content misclassified as abusive). The Voting Classifier and Random Forest models have the next highest scores around 0.526. The Support Vector Machine model has the lowest F1-score of approximately 0.486.

The graph shows the F1-scores for classifying age content using different models. The XGBoost model achieves the highest F1-score of around 0.77, it means that among all the models compared in the graph, the XGBoost model performed the best in correctly identifying and classifying age content. An F1-score of 0.77 indicates that the XGBoost model strikes a good balance between correctly detecting true positives (age content identified as such) and avoiding false positives (non-age content misclassified as age). The Voting Classifier and Random Forest models have the next highest scores around 0.74. The Naïve Bayes model has the lowest F1-score of approximately 0.611.

The graph shows the F1-scores for classifying gender content using different models. The XGBoost model achieves the highest F1-score of around 0.71, it means that among all the models compared in the graph, the XGBoost model performed the best in correctly identifying and classifying gender content.

An F1-score of 0.71 indicates that the XGBoost model strikes a good balance between correctly detecting true positives (gender content identified as such) and avoiding false positives (non-gender content misclassified as gender). The Support Vector Machine model has the lowest F1-score of approximately 0.581.

The graph shows the F1-scores for classifying mockery content using different models. The Voting Classifier model achieves the highest F1-score of around 0.35, it means that among all the models compared in the graph, the Voting Classifier model performed the best in correctly identifying and classifying mockery content.

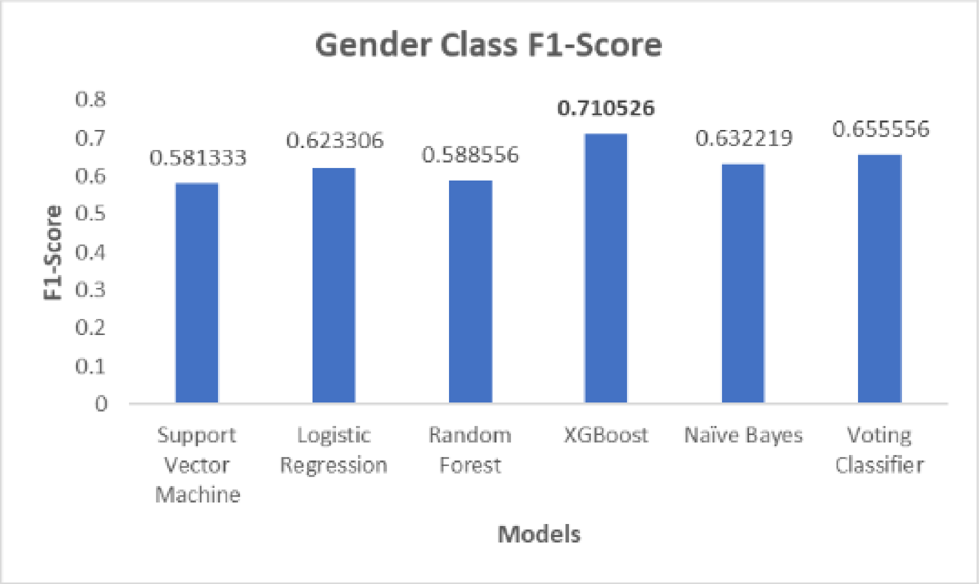

Fig. 14. Comparison of F1-Scores for gender content classification across ML models

An F1-score of 0.35 indicates that the Voting Classifier model strikes a good balance between correctly detecting true positives (mockery content identified as such) and avoiding false positives (non-mockery content misclassified as mockery). The Logistic Regression model has the lowest F1-score of approximately 0.231.

The graph shows the F1-scores for classifying not-cyberbullying content using different models. The Voting Classifier model achieves the highest F1-score of around 0.69, it means that among all the models compared in the graph, the Voting Classifier model performed the best in correctly identifying and classifying not-cyberbullying content.

An F1-score of 0.69 indicates that the Voting Classifier model strikes a good balance between correctly detecting true positives (not-cyberbullying content identified as such) and avoiding false positives (non- not-cyberbullying content misclassified as not-cyberbullying). The Naïve Bayes model has the lowest F1-score of approximately 0.603. The graph shows the F1-scores for classifying offensive content using different models.

The Voting Classifier model achieves the highest F1-score of around 0.53, it means that among all the models compared in the graph, the Voting Classifier model performed the best in correctly identifying and classifying offensive content. An F1-score of 0.53 indicates that the Voting Classifier model strikes a good balance between correctly detecting true positives (offensive content identified as such) and avoiding false positives (non-offensive content misclassified as offensive).

The Naïve Bayes model has the lowest F1-score of approximately 0.325. The graph shows the F1-scores for classifying religion content using different models. The XGBoost model achieves the highest F1-score of around 0.617, it means that among all the models compared in the graph, the XGBoost model performed the best in correctly identifying and classifying religion content.

An F1-score of 0.617 indicates that the XGBoost model strikes a good balance between correctly detecting true positives (religion content identified as such) and avoiding false positives (non-religion content misclassified as religion). The Naïve Bayes model has the lowest F1-score of approximately 0.563.

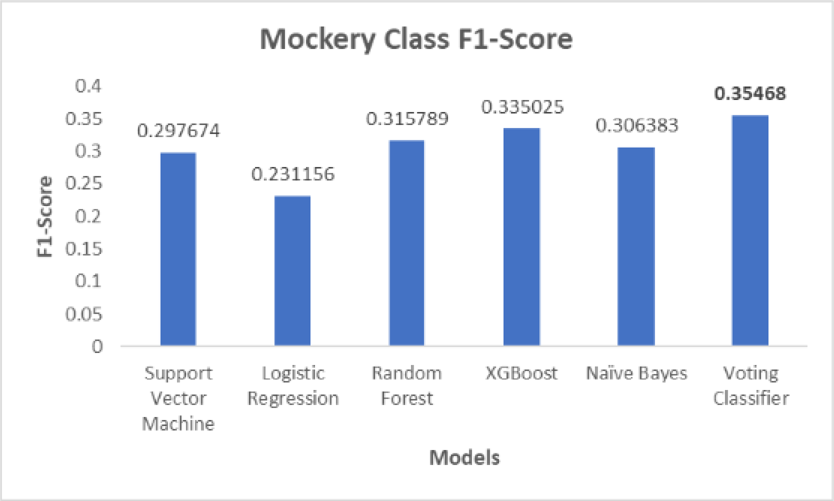

Fig. 15. Comparison of F1-Scores for mockery content classification across ML models

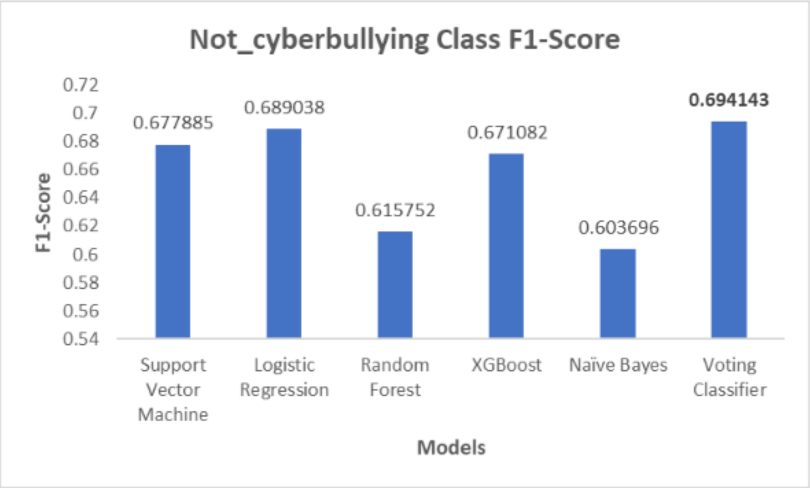

Fig. 16. Comparison of F1-Scores for not-cyberbullying content classification across ML models

6 Conclusion and Future Work

In this paper, we have presented a novel multi-class dataset, MC-Hinglish1.0 consisting of 8,400 Hinglish comments, annotated across seven cyberbullying categories: age, gender, religion, mockery, abusive, offensive, and not cyberbullying. We have evaluated the performance of different machine learning models, namely, Multinomial Naive Bayes, Support Vector Machines, Random Forest, XGBoost, and Logistic Regression – along with an ensemble model (voting classifier). Our experiments demonstrated the effectiveness of these models, with the XGBoost model achieving the highest individual performance with an accuracy of 0.602 and a weighted F1-score of 0.5804. However, the ensemble voting classifier outperformed all individual models, obtaining an accuracy of 0.609 and a weighted F1-score of 0.5879.

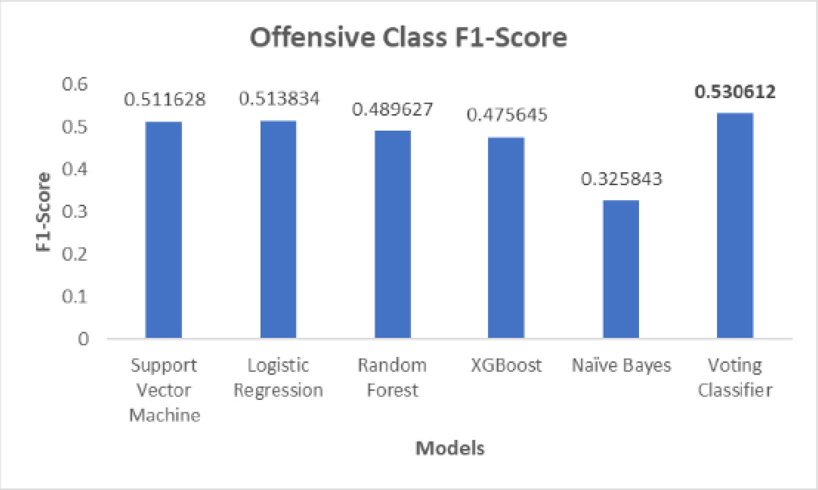

Fig. 17. Comparison of F1-Scores for offensive content classification across ML models

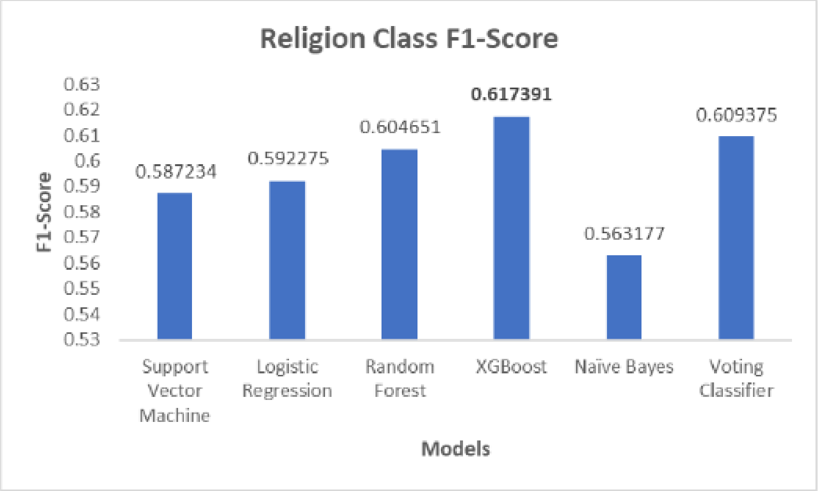

Fig. 18. Comparison of F1-Scores for religion content classification across ML models

The confusion matrix analysis revealed that while the Voting Classifier excelled in identifying instances of ”Age,” ”Gender,” ”Religion,” ”Offensive,” and ”Not Cyberbullying” classes, it faced challenges in distinguishing between certain ”Mockery,” and ”Abusive” classes. Our study highlights the importance of developing robust cyberbullying detection models tailored for code-mixed languages like Hinglish. The insights gained from this research can contribute to creating safer online spaces and promoting inclusivity in the digital realm. The future scopes of this work are summarized as follows:

– **Dataset Expansion:** As the need for robust cyberbullying detection systems grows, future work will involve expanding the dataset size. This includes acquiring additional data from diverse sources and languages to enhance the generalization capabilities of the model [15].

In addition to the Hinglish dataset, preparation of other multi-classification code-mixed Indic datasets, such as Binglish, will be undertaken. This expansion aims to capture the nuances and variations present in different languages and language mixes commonly used in online communication.

– **Investigation of Efficient Deep Learning-based Approaches:** To further improve cyberbullying detection accuracy and efficiency, future work will focus on exploring efficient deep learning-based approaches. This involves leveraging advancements in deep learning architectures, such as transformers, convolutional neural networks (CNNs), and recurrent neural networks (RNNs), tailored specifically for code-mixed text data [15].

  By designing models that effectively capture the complex linguistic patterns present in code-mixed text, the performance of cyberbullying detection systems can be significantly enhanced.

– **Incorporation of Multimodal Data:** Incorporating multimodal data, such as text, images, and user metadata, into the cyberbullying detection framework presents another avenue for future research [15].

  By analyzing multiple modalities simultaneously, the model can gain a more comprehensive understanding of online interactions, thereby improving the detection accuracy and robustness. Fusion techniques that combine information from different modalities will be explored to leverage the complementary nature of multimodal data [7].

– **Fine-tuning and Transfer Learning:** Fine-tuning pre-trained language models on cyberbullying detection tasks and leveraging transfer learning techniques will be investigated as a means to improve model performance. By leveraging large-scale pre-trained models such as BERT, GPT, or XLNet, the model can benefit from learning representations of text data across multiple languages and domains [7]. Fine-tuning these models on the specific cyberbullying detection task can help capture domain-specific features and improve overall performance.

– **Deployment and Real-world Application:** Finally, future work will involve the deployment and real-world application of the developed cyberbullying detection system. This includes integrating the model into social media platforms, online forums, and other digital communication channels to proactively identify and mitigate instances of cyberbullying [7].

  Continuous monitoring and refinement of the model based on user feedback and evolving linguistic trends will be essential to ensure its effectiveness in real-world scenarios.

References

1. **Ahammed, S., Rahman, M., Niloy, M. H., Chowdhury, S. M. M. H. (2019).** Implementation of machine learning to detect hate speech in bangla language. Proceedings of the 8th International Conference System Modeling and Advancement in Research Trends, pp. 317–320. DOI: 10.1109/SMART46866.2019.9117214.

2. **Ahmed, M. T., Rahman, M., Nur, S., Islam, A., Das, D. (2021).** Deployment of machine learning and deep learning algorithms in detecting cyberbullying in bangla and romanized bangla text: A comparative study. Proceedings of the International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies, pp. 1–10. DOI: 10.1109/ICAECT49130.2021.9392608.

3. **Akhter, A., Uzzal, K. A., Polash, M. (2019).** Cyber bullying detection and classification using multinomial Naïve Bayes and fuzzy logic. International Journal of Mathematical Sciences and Computing, Vol. 5, No. 4, pp. 1–12. DOI: 10.5815/ijmsc.2019.04.01.

4. **Alam, K., Bhowmik, S., Prosun, P. (2021).** Cyberbullying detection: An ensemble based machine learning approach. Proceedings of the third international conference on intelligent communication technologies

and virtual mobile networks, pp. 710–715. DOI: 10.1109/ICICV50876.2021.9388499.

5. **Ali, M. U., Lefticaru, R. (2024).** Detection of cyberbullying on social media platforms using machine learning. UK Workshop on Computational Intelligence, Springer Nature Switzerland, pp. 220–233. DOI: 10.1007/978-3-031-47508-5_18.

6. **Alqahtani, A. F., Ilyas, M. (2024).** An ensemble-based multi-classification machine learning classifiers approach to detect multiple classes of cyberbullying. Machine Learning and Knowledge Extraction, Vol. 6, No. 1, pp. 156–170. DOI: 10.3390/make6010009.

7. **Alqahtani, A. F., Ilyas, M. (2024).** A machine learning ensemble model for the detection of cyberbullying. DOI: 10.48550/arXiv.2402.12538e.

8. **Atoum, J. O. (2020).** Cyberbullying detection through sentiment analysis. Proceedings of the International Conference on Computational Science and Computational Intelligence, pp. 292–297. DOI: 10.1109/CSCI51800.2020.00056.

9. **Balakrishnan, V., Khan, S., Arabnia, H. R. (2020).** Improving cyberbullying detection using twitter users' psychological features and machine learning. Computers & Security, Vol. 90, pp. 101710. DOI: 10.1016/j.cose.2019.101710.

10. **Bhalla, A., Chadha, A. (2023).** An efficient model to detect the presence of hinglish text in youtube data. Proceedings of the International Conference on Advances in Computation, Communication and Information Technology, pp. 385–391. DOI: 10.1109/ICAICCIT60255.2023.10465821.

11. **Chakraborty, P., Seddiqui, M. H. (2019).** Threat and abusive language detection on social media in bengali language. Proceedings of the 1st International Conference on Advances in Science, Engineering and Robotics Technology, pp. 1–6. DOI: 10.1109/ICASERT.2019.8934609.

12. **Dubey, N., Kaushal, R. (2023).** Towards detection of cyberbullying in hinglish code mixed data. Proceedings of the 10th International Conference on Computing for Sustainable Global Development, pp. 1096–1100.

13. **Ghosh, R., Nowal, S., Manju, G. (2021).** Social media cyberbullying detection using machine learning in bengali language. International Journal of Engineering Research & Technology, Vol. 10, No. 5.

14. **Ji, C., Zou, X., Hu, Y., Liu, S., Lyu, L., Zheng, X. (2018).** XG-SF: an XGBoost classifier based on shapelet features for time series classification. 2018 International Conference on Identification, Information and Knowledge in the Internet of Things, Elsevier, Vol. 147, pp. 24–28. DOI: 10.1016/J.PROCS.2019.01.179.

15. **Kumar, S., Mondal, M., Dutta, T., Singh, T. D. (2024).** Cyberbullying detection in hinglish comments from social media using machine learning techniques. Multimedia Tools and Applications, pp. 1–22.

16. **Maity, K., Jain, R., Jha, P., Saha, S., Bhattacharyya, P. (2023).** Genex: A commonsense-aware unified generative framework for explainable cyberbullying detection. Conference on Empirical Methods in Natural Language Processing, pp. 16632–16645. DOI: 10.18653/v1/2023.emnlp-main.1035.

17. **Maity, K., Jha, P., Jain, R., Saha, S., Bhattacharyya, P. (2024).** "Explain thyself bully": Sentiment aided cyberbullying detection with explanation. Proceedings of the IEEE International Conference on Document Analysis and Recognition, pp. 132–148. DOI: 10.1007/978-3-031-41682-8_9.

18. **Maity, K., Kumar, A., Saha, S. (2022).** A multitask multimodal framework for sentiment and emotion-aided cyberbullying detection. IEEE Internet Computing, Vol. 26, No. 4, pp. 68–78. DOI: 10.1109/MIC.2022.3158583.

19. **Maity, K., Saha, S., Bhattacharyya, P. (2023).** Emoji, sentiment and emotion aided cyberbullying detection in hinglish. IEEE Transactions on Computational Social Systems, Vol. 10, No. 5, pp. 2411–2420. DOI: 10.1109/TCSS.2022.3183046.

20. **Mladenovic, M., Osmjanski, V., Vujicic-Stankovic, S. (2022).** Cyber-aggression, cyberbullying, and cyber-grooming: A survey and research challenges. ACM Computing Surveys, Vol. 54, No. 1, pp. 1–42. DOI: 10.1145/3424246.

21. **Muneer, A., Fati, S. M. (2020).** A comparative analysis of machine learning techniques for cyberbullying detection on twitter. Future Internet, Vol. 12, No. 4, pp. 187. DOI: 10.3390/fi12110187.

22. **Nath, S. S., Karim, R., Miraz, M. H. (2024).** Deep learning based cyberbullying detection in bangla language. arXiv. DOI: 10.48550/ARXIV.2401.06787.

23. **Nuthalapati, P., Abbaraju, S. A., Varma, G. H., Biswas, S. (2024).** Cyberbullying detection: A comparative study of classification algorithms. International Journal of Computer Science and Mobile Computing. DOI: 10.22541/au.170664263.38254624/v1.

24. **Pawar, R., Raje, R. R. (2019).** Multilingual cyberbullying detection system. Proceedings of the IEEE International Conference on Electro Information Technology (EIT), pp. 040–044. DOI: 10.1109/EIT.2019.8833846.

25. **Salawu, S., He, Y., Lumsden, J. (2020).** Approaches to automated detection of cyberbullying: A survey. IEEE Transactions on Affective Computing, Vol. 11, No. 1, pp. 3–24. DOI: 10.1109/TAFFC.2017.2761757.

26. **Singh, N. M., Sharma, S. K. (2023).** An efficient automated multi-modal cyberbullying detection using decision fusion classifier on social media platforms. Multimedia Tools and Applications, Vol. 83, No. 7, pp. 20507–20535. DOI: 10.1007/s11042-024-19031-z.

27. **Tripto, N. I., Eunus-Ali, M. (2018).** Detecting multilabel sentiment and emotions from bangla youtube comments. Proceedings of the International Conference on Bangla Speech and Language Processing (ICBSLP), pp. 1–6. DOI: 10.1109/ICBSLP.2018.8554875.

28. **Van-Hee, C., Jacobs, G., Emmery, C., Desmet, B., Lefever, E., Verhoeven, B., De-Pauw, G., Daelemans, W., Hoste, V. (2018).** Automatic detection of cyberbullying in social media text. PloS ONE, Vol. 13, No. 10, pp. 1–22. DOI: 10.1371/journal.pone.0203794.

29. **Verma, K., Milosevic, T., Cortis, K., Davis, B. (2022).** Benchmarking language models for cyberbullying identification and classification from social-media texts. Proceedings of the First Workshop on Language Technology and Resources for a Fair, Inclusive, and Safe Society within the 13th Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, pp. 26–31.

Article received on 22/05/2024; accepted on 06/06/2024.
**Corresponding author is Sahinur Rahman Laskar.*