Cyber bullying detection for Hindi-English language using machine learning

Karan Shah
Electronics Engineering
KJ Somaiya College of Engineering
Mumbai, India
karan19@somaiya.edu

Chaitanya Phadtare
Electronics Engineering
KJ Somaiya College of Engineering
Mumbai, India
cphadtare64@gmail.com

Keval Rajpara
Electronics Engineering
KJ Somaiya College of Engineering
Mumbai, India
keval.rajpara@somaiya.edu

Ninad Mehendale*
Electronics Department
KJ Somaiya College of Engineering
Mumbai, India
ninad@somaiya.edu
* Corresponding Author

***Abstract*—Nowadays, the internet is the world's biggest platform for communicating, connecting as well as sharing ideas, content, photos, videos, views, and daily updates globally. Social media is multifarious, which makes it extensive and interactive. Twitter, YouTube, Instagram, Linked In, Facebook and Whatsapp are some of the largest platforms. However, the enhancement of social connectivity often creates negative impacts on society that contribute to a couple of bad phenomena such as online abuse, harassment cyberbullying, cybercrime and online trolling. Cyberbullying frequently leads to serious mental and physical distress, particularly for women and children, and even sometimes force them to attempt suicide.Therefore, the identification of bullying text or message on social media has gained a growing amount of attention among researchers. Therefore our focus is to design and develop an effective technique to detect online abusive and bullying messages by merging Natural language processing and machine learning to develop a model that can detect offensive or hateful words in English and Hinglish language.**

***Index Terms*—Cyberbullying, Machine Learning, Natural language processing, Social media**

I. Introduction

Social media is a platform that allows people to post anything like photos, videos, documents extensively and interact with society [1]. People connect with social media using their computers or smartphones. The most popular social media includes Whatsapp, Facebook, Twitter, Instagram, You-tube and so on. In today's date, almost every person uses social media. According to statistics, nearly **2** billion users used social media networking sites in **2015**, and the figure has now increased to **3.96** billion. According to Backlinko, **58.11**% of the world population using social media. [1]

Social media has its perks, but it has negative aspects as well. Social media is misused and people are targeted based on caste, color, creed, gender, orientation, culture, and background. Various contents and ideas are neglected and criticized. However, this criticism has gone to the extent of bullying. This bullying through social media is termed as Cyberbullying. It causes damage to the reputation and image of the victim. The person being bullied falls into depression, inflicting self-harm, and in worst cases commits suicide. Thus, Cyberbullying is a severe problem that needs to be taken care of considering the severity of damage it causes to an individual's mental well-being.

Recent studies show that **36.5%** of people experienced Cyberbullying in their lifetime. **60%** of teenagers have experienced some sort of Cyberbullying. **87%** of young online users have accepted that they are witnesses of some kind of Cyberbullying occurring online. [2] Girls are more likely to become a victim of Cyberbullying as compared to boys.Overall, **36%** of girls have reported being cyberbullied compared to **26%** of boys. [3] The amount of Cyberbullying that now takes place has caused health issues for those targeted. **64%** of Cyberbullying victims are more likely to cause adolescents,depression, anxiety, self-esteem, emotional distress,mental, and behavioral problems.

Various solutions in the form of third-party applications have been deployed but the problem with these applications is that they are based on a simple keyword matching technique which gives less accurate results. Manually adding data in the database could be a subjective point of view of the creator and majority of the drawbacks include lack of labeled database or using a biased data set. Few implementations include lexicons, which is a common way to use databases for the detection of abusive words. However, it limits the scope of the application and disregards the statements which may not use abusive words but have hurtful meanings.

Therefore, this paper contributes to solving the problem by identifying and classifying text or messages of an intimidating or threatening nature when it is being drafted in real-time. Our aim is to build and developed an efficient technique to classify online abusive and bullying messages to critically monitor on various social media platforms and prevents the user from devastating after-effects of Cyberbullying by addressing the problem at its root by notifying the user which will help to create an awareness in the society. It proposes the use of Natural language processing techniques to provide a method for feature selection and the use of Machine Learning to perform the classification of the text if they have offensive words or meanings and would increase the accuracy of our models which was not embedded in the previous models. This study holds practical importance and served as a reference for new researchers in the domain of Cyberbullying detection.

This rest of the paper is organized as: Section II highlights the related works. Section III discusses the methodology. Sections IV, V, and VI explain the experimental settings, results, and discussion. Finally, Section VII discusses the limitation, future work, and conclusion as well.

II. Literature Survey

Cynthia Van Hee , Gilles Jacobs [4], proposed a model for automatic cyberbullying detection in social media text by modelling posts written by bullies, victims by standards of online bullying. two corpora were constructed by collecting data from social networking sites like Ask. fm. developed a model using tokenization, PoS-tagging, and lemmatization for pre-processing. Models were developed for English and Dutch to test for language conversion and subsequent accuracy. ML algorithm SVM gave the accuracy for the English language - 64% and for the Dutch language - 61%.

Mohammed Ali Al-Garadi, et al. [5], implemented a model to reduce textual cyberbullying because it has become the dominant aggressive behaviour in social media sites.They extracted data from Wikipedia, you- tube Twitter, Instagram and developed a model using tokenization lemmatization and N-gram was used up to 5 levels to calculate TF IDF and count vector for pre-processing. They gave a comparative analysis of ML algorithms using SVM , K clustering, Random forest, Decision Trees and concluded that SVM worked best amongst the four machine learning models. Kshitiz Sahay,et al. [6], Their focus was to identify and classify bullying in the text by analyzing and studying the properties of bullies and aggressors and what features distinguish them from regular users. The dataset they used was obtained from Wikipedia, YouTube, Twitter. In preprocessing removed URL and tags from dataset and performed Count Vectors and TF- IDF vectors. For classification they used Logistic Regression, SVM, Random Forest and Gradient Boosting.

Michele Di Capua [7], implemented a model inspired by Growing Hierarchical SOMs, which are able to efficiently cluster documents containing bully traces, built upon semantic and syntactic features of textual sentences. They followed an Unsupervised approach with Syntactic, Semantic, Sentiment analysis. In Pre-processing stop word removal, punctuation removal was done to generate word clusters. Social features were extracted. Convolutional neural networks were applied using Kohonen map (or GHSOM).

Homa Hosseinmardi, et al. [8], They proposed a model to automatically detect cyberbullying text in Instagram by modelling posts written by bullies. developed a system for deciding posts based on shortlisting words of caption. The paper suggested using image processing on Instagram posts for deciding emotional response or test response in case of text pictures.

Vijay Banerjee Jui Telavane et al. [9] developed the cyberbullying detection model using Convolution Neural Network and compared the accuracy with previous models. They used the twitter dataset which consists of 69874 tweets which converted to vectors. The accuracy of this model was 93.97% which was greater than other models.
Noviantho, S. M. Isa and L. Ashianti [3] created a classification model for cyberbullying using Naive Bayes method and Support Vector Machine (SVM).The dataset they used was collected from Kaggle which provides 1600 conversations in Formspring.me in which question and answer are used as labels. This consists of 12729 data of which 11661 data is labeled non-cyberbullying and 1068 is labeled cyberbullying.In data cleaning they removed the words like ‘haha’, ‘hehe’ , ‘umm’ etc. For balancing dataset they formed classification: 2 classes(cyberbullying and non -cyberbullying), 4 classes (non-cyberbullying, cyberbullying with low,middle and high severity level), 11 classes (non-cyberbullying, cyberbullying with 1-10 severity level). In preprocessing they used tokenizations, Transfer case, stop word removal, filter token, stemming, and generating n-gram. For classification they used Naive Bayes and SVM with linear,poly and sigmoid kernels. The SVM kernel with poly kernel gave most average accuracy 97.11%.

H. Watanabe, M. Bouazizi and T. Ohtsuki [10] their aim was to detect hate speech on Twitter. Their technique is based on unigram and patterns that are automatically collected from the dataset. Their aim was to classify tweets as clean, offensive and hateful. They used 3 types of datasets the first dataset was from crowdflower contains 14000 tweets are classified into clean,offensive and hateful; second was also from crowdflower tweets classified into offensive,hateful and neither; third dataset was from github in which tweets were classified into sexism, racism and neither. They combined 3 datasets to make a bigger dataset. In preprocessing removed URL and tags from tweets also they did tokenization, Part of Speech Tagging, and lemmatization.They used binary classification and ternary classification to identify sentiment-

based features, semantic features, Unigram features and pattern feature.Their proposed model gave accuracy of 87.4% for binary classification to classify tweets into offensive and non-offensive and 78.4% for ternary classification to classify tweets into hateful, offensive and clean.

J. Yadav, D. Kumar and D. Chauhan [11] developed a model to classify cyberbullying using a pre-trained BERT model. BERT model is a recently developed learning model by Google researchers. In this they use publicly available Formspring (a QA forum) and Wikipedia talk pages (collaborative knowledge repository) datasets and both datasets were manually labelled and also pre-processed . The Formspring dataset contains 12773 question-answer pair comments of which 776 are bully posts and Wikipedia dataset contains 115864 discussion comments which are manually annotated by ten persons of which 13590 comments which are labelled as bully. Their model gave accuracy of 94% for formspring dataset which is oversampled 3 times and 81% for wikipedia dataset.

S.E.Vishwapriya, Ajay Gour et al. [12], implemented a model for detecting hate speech and offensive language on twitter using machine learning. Datasets were taken from crowd flower and GitHub. Crowd flower dataset had tweets with labels Hateful, offensive and clean whereas GitHub dataset had columns tweet id and class such as sexism, racism and neither. Tweets were fetched by the tweet id using twitter API. These datasets were then combined. Tweets were converted to lowercase and Space Pattern, URLs, Twitter Mentions, Retweet Symbols and Stop words were removed. To reduce inflectional forms of words, stemming was applied. The dataset was then split into 70% training and 30% test samples. N-gram features from the tweets were extracted and were weighed according to their TF IDF values. Unigram, Bigram and Trigram features along with L1 and L2 normalization of TF IDF were considered. Logistic Regression, Naïve Bayes and Support vector machine algorithms were compared. 95% accuracy was obtained using Logistic Regression with L2 Normalization and n=3.

Lida Ketsbaia, Biju Issac et al. [13], proposed a model To detect hateful and offensive tweets using Machine Learning and Deep Learning. A data set created by the University of Maryland and Cornell University of about 35000 and 24000 tweets respectively was used with tweets labelled as Hate and Non hate. Tweets were converted into lowercase, numbers, URLs and user mentions, punctuation's, special characters and stop words were removed and contradictions were replaced. Data set was then balanced. Logistic Regression, Linear SVC, Multinomial and Bernoulli classifiers were applied in unigrams, bigrams and trigrams. Word2Vec technique was used to improve accuracy. Accuracy of 95% and 96% was achieved for the datasets.

Sindhu Abro, Sarang Shaikh et al. [14], implemented a model to detect cyberbullying via text using Machine Learning. CrowdFlower dataset was used. Tweets were converted into lowercase and URLs, usernames, white spaces, hashtags, punctuations and stop-words were removed. Tokenization and lemmatization was applied. Naives Bayes, Support Vector Machine, K Nearest Neighbour,Random forest and Logistic Regression were applied. N-gram with TFIDF, Word2vec and Doc2vec feature techniques were applied. SVM with a combination of bigram and TFIDF technique showed the best results.

III. Research Methodology

This paper suggests solutions for cyberbullying detection using various social media site like whatsapp, twitter and you-tube.

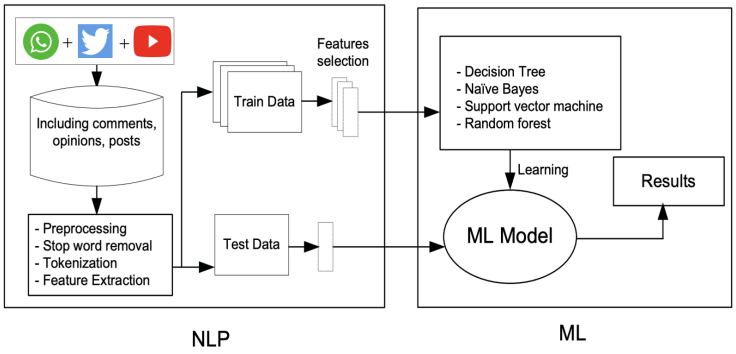

Fig. 1. Block Diagram

In this block diagram, we describe the cyberbullying detection framework which consist of two major part as follows :-
1. Natural Language Processing
2. Machine Learning.

In the first phase, we have collected real time tweets from Twitter, extracted Whatsapp chats and Youtube comments in English and Hinglish language. This real time data contains various unnecessary char- acters, so before applying to the machine learning algorithms to the comments, we need to clean and prepare the data for detection phase.

In the pre-processing stage we remove hashtags, stopwords, numeric data, hexadecimal patterns and convert the text into lower case. It is done by using numpy with the help of vectorize functions. We manually created a list of stopwords for English and Hindi English language and applied it to remove these words from the clean data because the presence of these unnecessary words adversely affects the accuracy and predictions of the model. We then applied NLP techniques like Tokenization to break raw text into words called as tokens, Lemmatization to remove a given word to its root word and vectorization for converting raw text into vectors or a number.

After pre-processing we split the dataset into two parts i.e training data and testing data. Next, we applied the two important features selection of text, which are:

1. Count Vectorizer

2. Term frequency- Inverse frequency.

In the second phase, we applied various machine learning approaches like Linear SVC, Decisison Tree, Naive Bayes, Bagging classifier, Logistic Regression, Random Forest, MultinomailNB, K Neighbours Classifier and Adaboost classifier to train the model and find the accuracy for each model based on the literature survey we conducted. We also calculated F1 score for evaluation purposes and improved accuracy by repeating the stages again. We wanted to select best pair between feature selection like TF-IDF and count vectorizer and machine learning model. For this we have done a comparative analysis between count vectorizer and TF-IDF, from this comparative analysis we found out the best pair which has higher accuracy and less prediction time and made its pickle file. After that we passed the testing data to the models to compare the accuracy of various algorithms with each other. After following these stages, our model is able to predict whether the text enter is toxic i.e bullying and harmful for the society or non - toxic i.e non-bullying in Hinglish language.

A. Data extraction

The dataset was taken from social media platform. For the English data set we scraped real time tweets from twitter and also took dataset from Kaggle. The dataset consists of actual tweets and messages which are extracted from various social media networking platforms. It has about 15,307rows. For the Hinglish dataset we have extracted real time tweets from Twitter, extracted chats from whatsapp and Youtube comments. It has around 3000 rows. We then merged them together to get a larger data set. The dataset has two columns namely headlines and Label. The label consists of -1 and 0 which indicates toxic i.e bullying and non-toxic i.e non-bullying sentences respectively as shown in Fig.2.

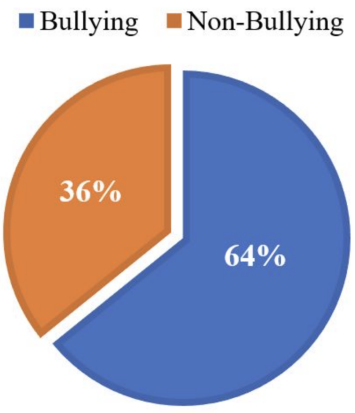

Fig. 2. Data Classification

Our dataset consist of 64.2% has toxic sentences and 35.8% has non-toxic sentences which is a diverse collection of negative words which are most commonly used by people in their day to day life.After extracting the data the next step is preprocessing the data. It is done because real world data contains lots of unnecessary characters, so we need to clean and prepare them for the detection phase. This required a lot of pre-processing before it could be used for testing or training purposes. So, we had to clean the data set and perform pre-processing techniques to achieve better outcomes.

B. Data Cleaning

The data is required to clean before passing through multiple ML models. as shown in Fig 3, this steps are necessary to removed from the data because they do not contribute for classification phase.This step is necessary to clean the raw data specifically.

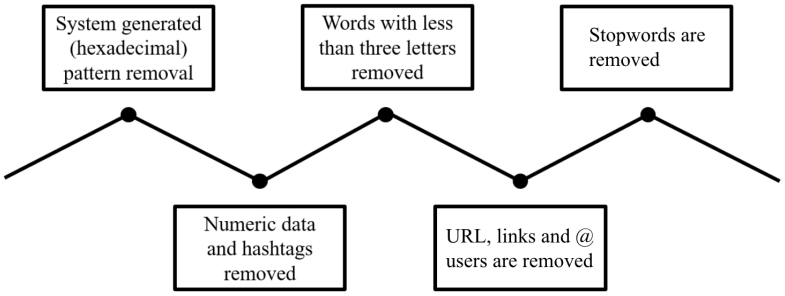

Fig. 3. Data Cleaning

When raw data of various users are imported from social media sites, it is collected with multiple characters and encoding. In this stage, we cleaned the data by removing punctuation marks, special characters, retweet symbols, hashtags, numeric values, hexadecimal values and URLs as they do not influence the meaning of the sentence. Words smaller than three letters long were eliminated. Also, the sentences are converted into lowercase to avoid duplication. We also manually created a list of stopwords for English and Hinglish language and applied them to remove these words from the clean data because presence of these unnecessary words adversely affects the accuracy and predictions of the model.

C. Preprocessing techniques

After cleaning the data we have applied Natural language processing techniques because the machine learning algorithm cannot work directly with the raw text that is they cannot understand the whole sentences given to it, so we transform these sentences into understandable format by using pre-processing techniques. This was followed by 3 key processes as shown in Fig:4 -

- **Tokenization** - Tokenization was used to each phrase throughout the tweet. Tokenization is the process of splitting a text sequence in smaller chunks, including sentences, words, terms, symbols that are called tokens.

- **Lemmatization** - Lemmatization is implemented after tokenization.this process is applied to reduce the

inflectional forms of each word into a root word.

- **Vectorization** - Finally, Vectorization was performed prior to sending the training and testing data set through the ML models. This process converts text into vector or a real numbers.This technique helps to measure the quality of the resulting vector representations.

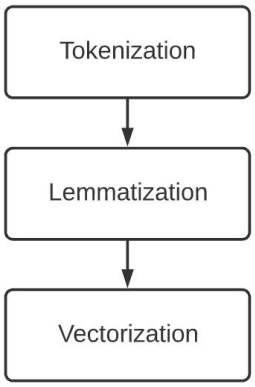

Fig. 4. Natural Language Processing Techniques

After data cleaning and applying pre-processing techniques as shown in Fig.3 and Fig.4, we split the data into training and testing.

D. Splitting the data

The datasets are divided into two types, i.e. training data set and testing data set. The testing datasets needs to be extracted from the platforms via text mining for a real time usage of the system. Both datasets pass through preprocessing techniques and various ML models.

E. Feature selection

After the splitting the data, we prepare the important features of the text, This technique helps to measure the quality of the resulting vector representations. This works with similar words that tend to close with words that can have multiple degrees of similarity. Vectorization is performed prior to sending the training and testing data set through the ML models.

1) Count Vectorization :-
Count Vectorization is used to convert the collection of words within a corpus into a vector of terms/term counts. The model will fit and learn the words from vocabulary and then try to make a word matrix in which the individual cells show the frequency of that word in a particular document, this is known as term frequency, and the columns are dedicated to each word in the corpus.

2) TF-IDF :-
TF-IDF means Term Frequency and Inverse Document Frequency, is a scoring measure which will evaluate how relevant the word in the document. This is done by multiplying two terms as follows:
1) The term frequency of a word in a document/text file.
2)The inverse document frequency of the word within a document/text file. It shows how rare or common the word is in the document. If the value is closer to 0 the more common the word is and vice versa.
Multiplying these two terms will gives the TF-IDF score of a word in the document.

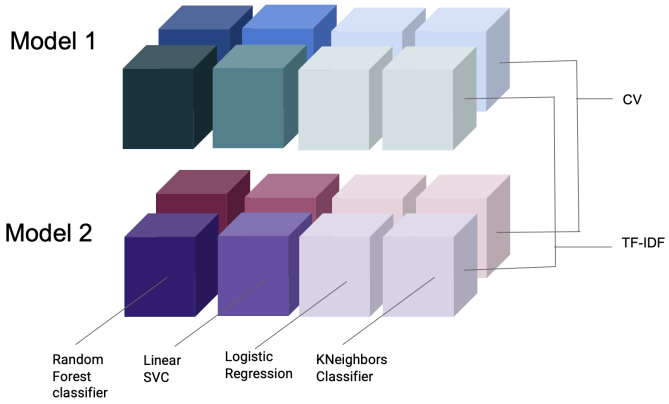

Fig. 5. Comparision between CV and TF-IDF

We have done a comparative analysis between this two feature extraction techniques with few algorithmsm through this we have observed that CV i.e. Count Vectorizer gives slightly better accuracy then TF-IDF i.e., Term Frequency- Inverse Document Frequency which is shown in the results section below in Fig.6 and Fig.7 Hence, for predicting bullying messages,comments, chats and tweets in Hinglish language and to build machine learning models for classification, we selected Count Vectorizer as our feature selection model.

F. Machine Learning Algorithms

After these steps of preprocessing and feature selection, various machine learning models were studied and identify 8 machine learning models to compare for functionality. These models were chosen on the basis of popularity, ease of use, back end working and research results of various authors. Following are different classifiers used in study:

1) Support Vector Machine (SVM) :-
Support Vector Machine(SVM) is a supervised machine learning classification algorithm whose objective is to fit data, and return a “best fit” hyperplane that divides or categorizes the data into different classes. After obtaining a hyper-plane, class can be predicted by some features of the classifier.
SVM is of two types: Linear SVM and Non-Linear SVM. Linear SVM is used for linearly separable data whereas non-Linear SVM is used for linearly non-separable data. In this study we only have two classes thus, we are going to linear SVM.

2) K-Nearest Neighbors (KNN) :-
The K-Nearest Neighbors (KNN) is a simple text classification

algorithm, which categorizes the new data using some similarity measure by comparing it with all available data. It generates the classification rules in the tree-shaped form, where each internal node denotes attribute conditions, each branch denotes conditions for outcome and leaf node represents the class label. In this algorithm, distance is used to classify a new sample from its neighbor. Thus, it finds the K-nearest neighbors among the training set and places an object into the class that is most frequent among its k nearest neighbors.

3) Logistic Regression :-
Logistic regression is a supervised machine learning algorithm, which is used to predict categorical dependent variables by using a set of independent variables. It is a statistical approach with which we can easily foretell a data input based on previous examinations of the data set in use. It can classify the data in 0 or 1, Yes or No, true or false and so on but instead of giving exact values it gives the probability that the given data belongs to class '1'.

4) Random Forest classifier :-
Random Forest classifier consists of large numbers of decision trees and each splits out a class prediction. The class who has higher votes becomes the model's prediction. Since it consists of multiple decision trees it is possible that some of them give wrong predictions but many others will be right.
They are fast, scalable, robust to noise, do not over-fit, easy to interpret and visualize with no parameters to manage. However as the number of trees increases the algorithm becomes slow for real time prediction.

5) Bagging Classifier :-
Bagging classifier is a widely used ensemble machine learning algorithm. In ensemble algorithms a group models work together to make predictions. Advantage of this is that several different methods counteract each model's weakness resulting in less error. Each model is trained individually and combined in an averaging process. Bagging algorithm reduces model overfitting.

6) Stochastic Gradient Descent(SGD) Classifier :-
Stochastic Gradient Descent(SGD) is an optimization algorithm used to find parameters that will reduce a cost function. In other words, it is used for discriminative learning of linear classifiers under convex loss functions such as SVM and Logistic regression. SGD Classifier is a linear classifier like SVM, logistic regression but it is optimized by the SGD. in other words SGD is an optimization method, Logistic Regression or linear Support Vector Machine is a machine learning algorithm/model.

7) Adaboost Classifier :-
Adaboost is one of the ensemble boosting classifiers. Boosting algorithms try to build a strong learner or model from mistakes of other weaker models. It tries to reduce errors which arise when it is not able to identify trends in data. The AdaBoost algorithm uses one-level decision trees as weak learners then those are added sequentially to the ensemble.At each subsequent step, the model attempts to correct the predictions made by the model before it in the sequence. This is achieved by weighing the training dataset to put more focus on training examples on where previous models make predictable errors.

8) Multinomial NB Classifier :-
Multinomial NB classifier is a probabilistic machine learning algorithm which is mostly used for Natural Language Processing (NLP). The algorithm is based on Bayes theorem and predicts the tag of a text for example newspaper article or piece of mail. It calculates the probability of a given sample and returns a tag with high probability. It follows the principle that each feature being classified is not related to any other feature. The presence or absence of one feature does not affect the presence or absence of the other feature.

G. Evaluation Phase

The bullying detection algorithms are implemented using python machine learning packages. The performances are analyzed by calculating True Negatives (TN), False Positives (FP), False Negatives (FN) and True Positives (TP). These four numbers can be shown as a confusion matrix. Different performance metrics are used to assess the performance of the constructed classifier.Some common performance measures performances are analyzed with respect to the following metrics in text categorization are

- **Precision:** - Precision is also known as the positive predicted value. It is the proportion of predictive positives which are actually positive.

$$Precision = \frac{TP}{TP + FP}$$

- **Recall** - Recall is the proportion of actual positives which are predicted positive

$$Recall = \frac{TP}{TP + FN}$$

- **F-Measure** - F-Measure is the harmonic mean of precision and recall. The standard F-measure (F1) gives equal importance to precision and recall.

$$F - measure = \frac{2 * precision * recall}{precision + recall}$$

- **Accuracy** - Accuracy is the number of correctly classified instances (true positives and true negatives).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

IV. Experiment And Result

We have used the four machine learning algorithms like Support Vector Machines (SVM), Random Forest (RF), K-Nearest Neighbors (KNN) and Logistic Regression(LR) to choose best feature extraction model between count vectorizer and term frequency-inverse document frequency.
In this section, we describe:

A. Comparative Analysis between 2 feature extraction models.

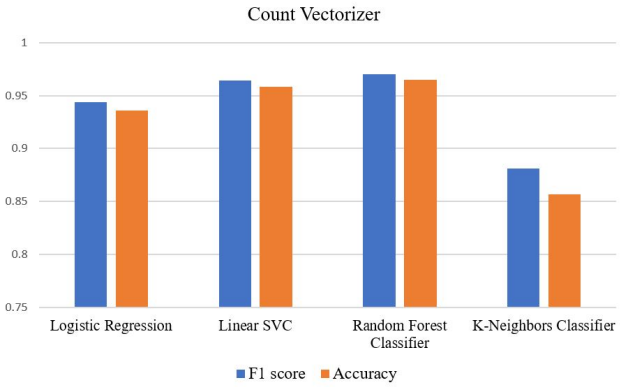

Fig. 6. Comparison of Algorithms with Count Vectorizer

From the above plot Fig.6 we can say that Logistic regression and Random forest classifier gives the highest accuracy than the other two algorithms. Best accuracy and F1 score is given by Random Forest with **96.5%** and **97.0%** respectively. Linear SVC gives best recall score with **96.37%** and also takes very less time for training and predicting the output. Although Random Forest provides best accuracy with **96.5%**, it takes more time for training and predicting the output.

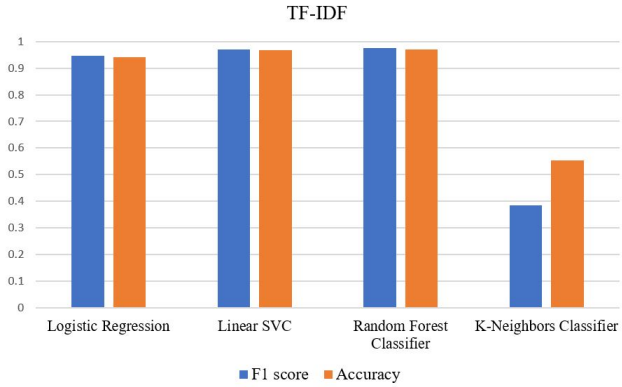

Fig. 7. Comparison of Algorithms with Term Frequency-Inverse Document Frequency

From the about plot Fig.7 we can say that Linear SVC and Random forest classifier gives the highest accuracy than the other two algorithms. Best accuracy and F1 score is given by Random Forest Classifier with **97.1%** and **97.2%** respectively. Linear SVC gives best recall score with **97.13%** and also takes very less time for training and predicting the output. Although Random Forest provides best accuracy with **97.1%**, it takes more time for training and predicting the output.

Algorithms	Logistic Regression	Linear SVC	Random Forest Classifier	K-Neighbors Classifier
CV	0.936	0.958	0.965	0.857
TF-IDF	0.940	0.967	0.971	0.554

From the about table, we observed that CV i.e. Count Vectorizer gives slightly better accuracy then TF-IDF i.e., Term Frequency -Inverse Document Frequency because count vectorizer picks out the **relevant words** while training the model, whereas TF-IDF does not because it tries to make the model building less complex by reducing input dimension's, due to which it losses the importance of words present in corpus.

So now we have move ahead to classify and predict bullying messages in comments, chats, tweets on various social media platforms in Hinglish language and we have move ahead to apply more ML algorithms like Multinomial NB, Decision Tree Classifier, Ada-boost classifier and Bagging classifier with TF-IDF as feature extraction model.

Algorithms	CV accuracy	TF-IDF accuracy	CV F1-score	TF-IDF F1-score
Decision Tree Classifier	0.955	0.962	0.965	0.968
Linear SVC	0.94	0.958	0.954	0.966
Bagging Classifier	0.955	0.956	0.961	0.965
Logistic Regression	0.935	0.944	0.949	0.949
Stochastic Gradient Classifier	0.933	0.943	0.942	0.947
Multinomial NB	0.890	0.907	0.903	0.918
Adaboost Classifier	0.827	0.830	0.832	0.851

From the above table, we observed that Decision Tree classifier provides highest accuracy among all the algorithms but has worst training and prediction time which is similar to random forest classifier. Linear SVC, Logistic Regression and SGDC Classifier have more or less similar performance in terms of accuracy and F1 score but among them Linear SVC

and logistic regression performs faster that is provides best training and prediction time. Adaboost Classifier provides less accuracy among all the algorithms. Linear SVC and SGD (stochastic gradient classifier) is able to give a comparatively better output when adjusted with parameters on the larger dataset because it takes less time for training the algorithms and provides better accuracy then the rest.

V. Limitation

The major limitations of this project revolves around acquiring labelled datasets in hinglish languages for training of Machine Learning models. This project is fairly new and there is a scarcity of larger datasets which are accurately labelled. Most of the ones available are results of projects made using smaller datasets, which result in not very accurate results. With a dataset of even 18,000 tweets, an accuracy of 95% results into unreliable output. Finding datasets is expensive, especially ones which are 100% correct. This is because these datasets are labelled manually which require a lot of human effort. Progress is being made slowly and steadily for the above reason, which will yield into much better results in the near future.

VI. Conclusion

In particular, cyberbullying has become more common and has begun to raise significant social issues with the rising prevalence of social media sites and increased social media use by teenagers. There needs to design automatic cyberbullying detection method to avoid bad consequences of cyber harassment. Considering the significance of cyberbullying detection, in this study, we investigated that CV, Count Vectorizer provides slightly better accuracy then TF-IDF, term frequency-Inverse Term Frequency.The comparison of various ML models like Support Vector Machine, Logistic Regression, K Nearest Neighbor and Random Forest, Bagging Classifier,Decisison Tree Classifier, SGDC classifier, Multinomial Classifier, and AdaBoost Classifier we conclude that Linear SVC and SGD (stochastic gradient classifier) is able to give a comparatively better results in classifying and predicting bullying messages in hinglish languages and takes less time to train and predict then other alogrithms.This classification can be further improved with a deeper focus on sentiment, semantic and syntactic analysis. An integration with convolutional neural networks or deep learning will help increase the accuracy even further due to the complexity considered in the model. This model can be made an integral of various social media platforms to make it a necessary feature for all users. It can help reduce cyberbullying to a much greater extent and spread awareness about it.

References

[1] B. Dean, "How many people use social media in 2021? (65+ statistics)," Sep 2021. [Online]. Available: https://backlinko.com/social-media-users

[2] J. W. Patchin, "Summary of our cyberbullying research (2004-2016)," Jul 2019. [Online]. Available: https://cyberbullying.org/summary-of-our-cyberbullying-research

[3] Noviantho, S. M. Isa, and L. Ashianti, "Cyberbullying classification using text mining," in *2017 1st International Conference on Informatics and Computational Sciences (ICICoS)*, 2017, pp. 241–246.

[4] C. Van Hee, G. Jacobs, C. Emmery, B. Desmet, E. Lefever, B. Verhoeven, G. De Pauw, W. Daelemans, and V. Hoste, "Automatic detection of cyberbullying in social media text," *PloS one*, vol. 13, no. 10, p. e0203794, 2018.

[5] M. A. Al-Garadi, M. R. Hussain, N. Khan, G. Murtaza, H. F. Nweke, I. Ali, G. Mujtaba, H. Chiroma, H. A. Khattak, and A. Gani, "Predicting cyberbullying on social media in the big data era using machine learning algorithms: Review of literature and open challenges," *IEEE Access*, vol. 7, pp. 70 701–70 718, 2019.

[6] K. Sahay, H. S. Khaira, P. Kukreja, and N. Shukla, "Detecting cyberbullying and aggression in social commentary using nlp and machine learning," *International Journal of Engineering Technology Science and Research*, vol. 5, no. 1, 2018.

[7] M. Di Capua, E. Di Nardo, and A. Petrosino, "Unsupervised cyber bullying detection in social networks," in *2016 23rd International conference on pattern recognition (ICPR)*. IEEE, 2016, pp. 432–437.

[8] H. Hosseinmardi, S. A. Mattson, R. I. Rafiq, R. Han, Q. Lv, and S. Mishra, "Detection of cyberbullying incidents on the instagram social network," *arXiv preprint arXiv:1503.03909*, 2015.

[9] V. Banerjee, J. Telavane, P. Gaikwad, and P. Vartak, "Detection of cyberbullying using deep neural network," in *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*. IEEE, 2019, pp. 604–607.

[10] H. Watanabe, M. Bouazizi, and T. Ohtsuki, "Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection," *IEEE access*, vol. 6, pp. 13 825–13 835, 2018.

[11] J. Yadav, D. Kumar, and D. Chauhan, "Cyberbullying detection using pre-trained bert model," in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*. IEEE, 2020, pp. 1096–1100.

[12] A. Gaydhani, V. Doma, S. Kendre, and L. Bhagwat, "Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach," *arXiv preprint arXiv:1809.08651*, 2018.

[13] L. Ketsbaia, B. Issac, and X. Chen, "Detection of hate tweets using machine learning and deep learning," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2020, pp. 751–758.

[14] S. Abro, Z. S. Shaikh, S. Khan, G. Mujtaba, and Z. H. Khand, "Automatic hate speech detection using machine learning: A comparative study," *Machine Learning*, vol. 10, no. 6, 2020.