Detection of Cyberbullying on Twitter using Machine Learning: A Comparative Analysis and Performance Evaluation

[1]Salisu Suleiman, [2]Ms. Prashansa Taneja, [3]Ms. Ayushi Nainwal

[1]M.Tech (CSE) Student 4th Semester, [2,3]Assistant Professor

Dept. of Computer Science and Engineering, Alakh Prakash Goyal University, Shimla, India

[1]salisusumar@gmail.com, [2]prashansa@agu.edu.in, [3]ayushinainwal@gmail.com

Abstract - Due the significant development of technology and high demand for internet usage by over billions of people globally have resulted in the occurrences of so many online criminal offences such as cyberbullying, cyberstalking, internet fraud, file sharing and piracy among others. The means of using electronic devices to harm, harass or insult other people by sending harmful and abusive messages is termed as cyberbullying. Cyberbullying is among the most prevalent issues on social networks nowadays and it is hard to be identified with human effort alone. In order to make social media a safer place for communication, machine intervention in the prevention and control of such issues is necessary. Two methods of supervised machine learning were employed in our research namely; Traditional supervised learning and Ensemble supervised learning. The traditional methods used three Machine Learning classifiers: Gaussian Naïve Bayes (GNV), Logistic Regression (LR), and Decision Tree (DT) whereas the ensemble method used Random Forest (RF) and Adaboost classifiers. We used our dataset to train and test our model for detecting and classifying bullying content into two categories: bullying and non-bullying (binary classification model) and Term Frequency-Inverse Document Frequency (Tf-idf) was used for extraction of features from twitter dataset downloaded from kaagle. The purpose of this research is to compare and evaluate the performances of each machine learning algorithm used in this study. The result revealed that the overall best performance was shown by Random Forest classifier giving accuracy of 92% and on the same dataset; Nave Bayes performed the poorest, with an accuracy of only 62%. The ensemble methods performed better than the traditional supervised learning method interms of every performance metrics. In this study, we employed Jupyter notebook 6.4.5 from Anaconda navigator as a coding environment.

Keywords: Cyberbullying, Cyberstalking, Machine learning, Social Media, Ensemble Methods.

I. INTRODUCTION

Social media can be termed as a platform that allows users to share anything they want, like; images, documents and videos as well as communicate with others [1]. People engage with one another on social media sites via computers or cellphones. The most frequently and globally used social media includes Facebook, Twitter, Instagram, TikTok and so on.

With the rapid advancement of internet and technology, our everyday lives have now become more reliant on social media. It lets users to exchange information with one another with just a few taps and/or clicks using a variety of applications [2]. There are around four billion Internet users, three billion social media users and five billion mobile users, according to [3] in the world. Despite its importance, though, social media comes with a slew of problems and challenges. For example, many antisocial behaviors on social media may include; cyberbullying, cyberstalking, and cyberharassment. These behaviours have now become inculcated in our culture and are no longer limited to youth; anybody can be affected.

According to a study taken by [4], cyberbullying affects nearly half of all youths in America. The victim of bullying suffers both physically and mentally due to the harmful nature of the bullying. Because the misery of cyberbullying is too great to bear, victims of cyberbullying commit self-destructive actions such as suicide.

Consequently, detecting and combating cyberbullying is critical for teens' safety. In this era of web 4.0, where people live on digital and online platforms; it is highly

 DOI : 10.35291/2454-9150.2022.0327

difficult to protect society from the frightening rise of cybercrime.

To detect cyberbullying content, we suggested a cyberbullying detection model based on two supervised machine learning algorithms (standard and ensemble methods). For our supervised learning technique analysis, we employed Naïve Bayes (NB), Logistic Regression (LR), and Decision Tree (DT) as conventional techniques while Random Forest (RF) and AdaBoost (AB) Classifiers were used as ensemble methods. In our research, we compared the performance of all the classifiers used and found that the Gaussian Naïve Bayes classifier was the poorest, whereas the Random Forest Classifier gave the best result in terms of every metric. The result’s evaluation also shows that Ensemble supervised method performed better than traditional supervised methods. In our research, we used Jupyter notebook 6.4.5 from Anaconda navigator, a prominent data science toolbox, as a coding environment.

II. LITERATURE SURVEY

Lots of researches have been carried out to discover possible solutions to cyberbullying attacks occurring in social networking sites.

According to [1], Four powerful ML algorithms such as: SVM, RF, NV and DT were used to identify abusive and bullying messages on social media in English using two features such as Bag of Words (BoW) and TF-IDF to analyse the level of accuracy of four Machine learning algorithms used. Facebook and twitter dataset were successfully downloaded from kaggle.com. SVM outperformed all other machine learning classifiers in the study, according to the results.TF-IDF outscored BoW in the same way.

Most cyberbullying detection studies, according to [6,] were done in a single language. Because of the danger of cyberbullying, a model capable of detecting it in many languages, including Hindi and Marathi, was developed. The datasets were collected from various sources including newspaper reviews, manually collected tourist reviews, and tweets acquired from the Twitter API. The result revealed that F1-score had up to 97% and accuracy was measured to be 96%. The percentages were obtained in both Hindi and Marathi. Likewise, Logistics Regression (LR) outperforms SGD and MNB in all the three different dataset used.

In a study conducted by [7], four machine learning algorithms such as SVM, LR, RF, and Multilayered perceptron algorithms were used for detection of bullying text in English, three distinct textual features such as Word2Vec, TF-IDF, and BoW were used, and the dataset was obtained from Wikipedia and Twitter. The results indicated that the Twitter dataset had over 90% accuracy while the Wikipedia dataset had 80% accuracy when applying the same machine learning classifiers and BoW and Tf-Idf features considerably outperformed Word2Vec feature.

According to [8] it is necessary to detect cyberbullying on many social media platforms, hence several machine learning techniques such as SVM and Naïve Bayes are employed to recognize the presence of bullying messages on Twitter and Wikipedia social platforms in both Arabic and English languages. In both the twitter and wikipedia datasets utilized, NV outperformed SVM with 90.8 percent accuracy.

[9] built a model for identifying cyberbullying on Twitter that used a range of linguistic features. They were able to design a series of machine learning models, including linear, tree-based, and deep learning models, with the best scoring above 90% on the four criteria of accuracy, precision, recall, and F-measure during their research.

Authors in [10] used Python and Tensor-Flow to implement their cyberbullying model. They compared DNN's performance to standard machine learning models, and found that DNN-based models are more flexible to new datasets and outperformed conventional machine learning models on the Twitter dataset.

Support Vector Machine Classifier (SVM) was used to detect cyberbullying in English and Dutch in another research [11]. They carried out their investigation using linear support vector machines which leverage a large feature set and explore which information sources contribute the most to the task. For the English and Dutch languages, the classifier produces f1-score of 64 percent and 61 percent, respectively.

[12] used the SVM, Radial Basis Function, MLP, LR, and SGD algorithms to develop a model that identifies comments in datasets according on whether or not they involve cyberbullying.

In order to reduce classification time, Chi2, Support Vector Machine-Recursive Feature Elimination (SVM-RFE), Minimum Redundancy Maximum Relevance (MRMR), and Relief feature selection approaches were employed to evaluate the classifiers' performance. The classification times for YouTube, Formspring.me, Myspace, and Formspring.me were lowered by 20 times, 2.5 times, and 10 times, respectively, after employing feature selection algorithms. The results showed that classifiers with an F-measure value above 0.930, such as Stochastic Gradient Descent (SGD) and Multilayer Perceptron, outperformed classifiers with an f-measure value less than 0.930. (MLP), outperformed other classifiers, and the SVM-RFE algorithm, which uses the selected 500 features, produces the best results.

To address cyberbullying issues, the authors in [13] suggested a convolutional neural network cyberbullying

 DOI : 10.35291/2454-9150.2022.0327

detection model (CNN-CB). The dataset used in the studies was retrieved from twitter using the twitter streaming API, and the results indicated that the CNN-CB algorithm outperformed traditional content-based cyberbullying detection in all the three performance evaluation measures with a 95 percent accuracy.

As machine learning classifiers, the authors of [14] utilised Random Forest (RF), AdaBoost (ADB), Naïve Bayes (NB), Logistic Regression (LR), Light Gradient Boosting Machine (LGBM), Stochastic Gradient Descent (SGD), and Support Vector Machine (SVM). Each of these methods was tested using all the performance metrics to determine the classifiers' recognition rates applied to the global dataset. The seven classifiers employed in the study were tested on a large dataset of 37,373 tweets. Logistic regression had the highest F1-score of 0.928, Stochastic Gradient Descent (SGD) had the best precision of 0.968 and Support vector Machine (SVM) had the best recall of 1.00 among the classifiers. Finally, with a median accuracy of around 90.57 percent, the studies revealed that Logistic Regression is superior.

III. METHODOLOGY

Our proposed approach for detecting cyberbullying is detailed in this section. The datasets and algorithms we employed were also described.

Figure 1 depicts the suggested framework for identifying cyberbullying.

Among the components that make up our system are the gathering of raw datasets, Natural Language Processing (NLP), Machine Learning Model, and Result Analysis.

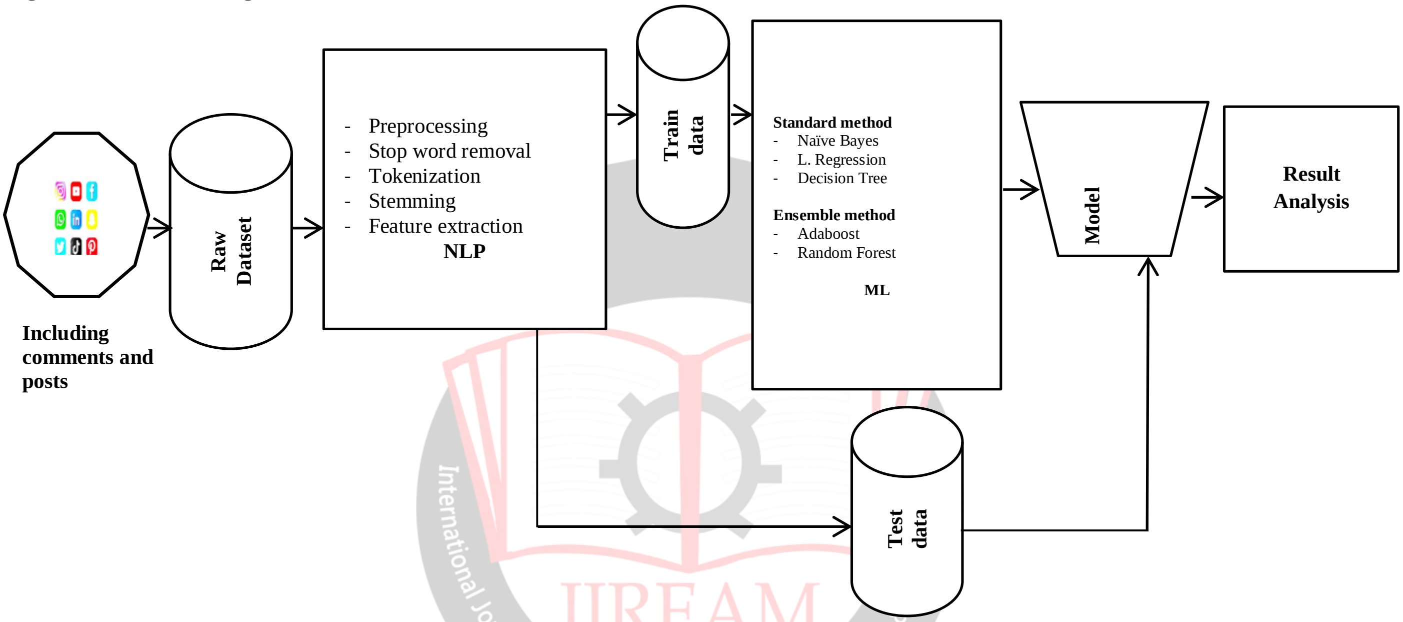

Figure 1: Proposed Frame work for Cyberbullying Detection

3.1 Dataset

In this research, to get the final results, we used Dataturks' Tweet Dataset for Cybertroll Detection obtained from Kaggle [15]. Because of the importance of the problem we were aiming to tackle, we needed to choose a dataset that was complete, reliable, relevant, and to the point.

Here is the Description of the dataset:

1) It is a partially manually labelled dataset.

2) Total Instances: 20001.

The dataset is described having 2 attributes tweet and label [0 corresponds to No while 1 corresponds to Yes].

3.2 Data Collection

We collected Dataturks' Tweet Dataset for Cybertroll Detection from Kaggle [15] for both training and testing of our model which contains about 20001 total instances. At first, we considered many other datasets as well; many of them either had missing attributes, were too low in quality, or were found to have irrelevant data after manual inspection. Thus, after having tried out of many other open sourced datasets, we came down to [15] as it seemed in line with all the parameters we required.

3.3 Data Cleaning

The dataset used was set in a json format. Since the fields of the dataset were relatively simple to interpret, the original set of fields in the annotation attribute was removed, and filled with the label values to simplify the next step. The number of instances for each class is mentioned in table 3.1.

Table 1: Twitter Dataset Instances

	Twitter
Total instances	20001
Cyberbullying instances	7822

 DOI : 10.35291/2454-9150.2022.0327

International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-08, Issue-03, June 2022

Non- Cyberbullying instances	12179

3.4 Data Preprocessing

The preprocessing steps were done as follows using the nltk library along with regex:

1) Word Tokenization: A token is a single thing that serves as a phrase or paragraph's building block. Word Tokenization converts our text to separate words in a list.

2) Stop Words Removal: This is done using nltk.corpus.stopwords.words('english') to fetch a list of stopwords in the English dictionary, after which they are removed. Stop words are non-significant words like "the," "a," "an," and "in" that have no bearing on the meaning of the data to be interpreted.

3) Punctuation removal: Only characters that are not punctuation are saved here, which may be verified using string.punctuation

4) Stemming: A linguistic normalisation procedure in which terms are reduced to their underlying word. We stem the tokens using nltk.stem.porter.PorterStemmer to get the stemmed tokens. The phrases "connection," "connected," and "connecting," for example, may all be reduced to the basic word "connect."

5) Digit removal: We removed any numerical data since it does not contribute to cyberbullying.

6) Feature Extraction: The next step is to extract features so that it can be used with ML algorithms, for which we used Term Frequency-Inverse Document Frequency (TF-IDF) Transformer using Python's sklearn libary. The TF-IDF is a statistical tool for determining the importance of a word.

The inverse of the term's document frequency is multiplied by the number of times a word appears in the document.

TF-IDF utilizes a strategy that minimizes the weight (importance) of words that appear in numerous texts in common, declaring them incapable of differentiating the documents, rather than computing the frequency of words like CountVectorizer does. The outcome matrix consists of each document (row) and each word (column) and the importance (weight) computed by tf * idf values of the matrix. If a term has a high tf-idf in a document, it has most likely appeared in that document and must be missing in others. As a result, the words must be a signature word.

Attribute evaluation is done manually as can be seen where we have printed the top 25 words according to the calculated tf-idf scores. Some Top ranked words for the dataset were: [hate, fuck, damn, suck, ass, that, lol, im, like, you, it, get, what, no, would, bitch].

3.5 Data Resampling

As the data was skewed, Resampling had to be performed on the training data, Firstly, the data was split into Training and Testing in 80:20 ratio and resampling was performed on the training data.

• As we had ample data to work with, we used oversampling of the minority class. This means that if the majority class has 1,000 instances and the minority class only has 100, this technique will oversample the minority class to give it 1,000.

• For Oversampling, RandomOverSample function is used from imblearn package for all the "not majority" classes which in our case, was only the 1 minority class.

The training data contained 9750 CB and NON-CB occurrences after resampling.

3.6 Machine learning algorithms

Machine Learning (ML) is defined as the ability of a computer to teach itself how to take a decision using available data and experiences [6]. The data is known as *training data*. Decisions to be taken in ML might be classification or prediction. The computer classifies a new piece of data by training models using learning algorithms. If the learning algorithm (or the training model) is depending on labelled data, then this algorithm is considered a supervised algorithm [6].

There might be a corpus of data manually tagged as containing or not containing damage in cyberbullying detection, as mentioned later. When the training data is unlabeled, then the algorithms are called Unsupervised Learning algorithms [6]. They learn how to classify by themselves based on similarities and differences between data. When both supervised and unsupervised learnings are combined together, the algorithm is known as Semi-supervised Learning algorithm [6].

In order to detect cyberbullying from social media texts, some typical machine learning classifiers employed in this research are discussed below:

A. Gaussian Naïve Bayes

Gaussian Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem of mathematics. The Bayes' theorem, in basic words, determines the probability of an event based on prior knowledge of factors that may be important to the event. It's a group of algorithms that all work on the same premise: that each pair of classified features is independent of the others. For binary (two-class) and multi-class classification issues, Gaussian Naive Bayes is an appropriate classification algorithm. When stated using binary or categorical input values, the approach is easier to understand. Gaussian Naïve Bayes gets its name from the

 DOI : 10.35291/2454-9150.2022.0327

fact that the probability computation for each hypothesis is simplified to make it tractable and it is often extended to real-valued attributes by assuming a Gaussian distribution. This extension of Gaussian Naive Bayes is called Gaussian Naïve Bayes. There are also Multinomial Gaussian Naive Bayes and Bernoulli Gaussian Naive Bayes, in addition to Gaussian Naive Bayes. Because we simply need to estimate the mean and standard deviation from the training data, we chose Gaussian Naive Bayes as the most extensively used and one of the easiest to apply. The classifier was implemented using sklearn.naive Bayes_package.

B. Logistic Regression

Regression analysis is a predictive modelling technique that investigates the relationship between the objective or dependent variable of the dataset and its independent factors. When the target and independent variables have a linear or non-linear relationship, and the target variable has continuous values, regression analysis techniques are applied. Regression analysis involves determining the best fit line, which is a line that passes through all of the data points while keeping the gap between the line and each data point as minimal as possible. When the dependent variable is discrete, one form of regression analysis approach is used: logistic regression. For instance, 0 or 1, true or false, and so forth. This signifies that the target variable can only have two values and is a sigmoid. A sigmoid curve represents the connection between the goal and independent variables, converting any real value to a value between 0 and 1. We picked Logistic Regression since our data set was huge and had about equal incidence of values in target variables. Furthermore, there was no association between the dataset's independent variables.

The classifier was implemented using sklearn.linear model package.

C. Decision Tree Classifier

A Decision Tree is constructed by posing a series of questions to the dataset. A follow-up question is asked after each response until a conclusion concerning the record's class label is made. The series of questions and possible responses may be grouped into a decision tree, which is a hierarchical structure made up of nodes and directed edges. Root nodes, internal nodes, and leaf nodes are the three types of nodes found in it. A class label is assigned to each leaf node in a decision tree. The non-terminal nodes, which include the root and other internal nodes, include attribute test criteria needed to differentiate between records with different properties. We start at the root of the tree and split the data on the feature with the most information gathered, using the decision process (IG) and reduction in uncertainty towards the final decision. The splitting technique can then be repeated recursively at each child node until the leaves are pure. This signifies that all of the samples at each leaf node are from the same class.

The classifier was implemented using sklearn.tree package.

D. Adaboost Classifier

AdaBoost classifier refers to ensemble algorithm that is iterative. The main principle behind boosting methods is to progressively train predictors, with each correcting the preceding one. AdaBoost classifier by merging numerous low-performing classifiers, creates a strong classifier resulting in a high-accuracy strong classifier. Adaboost's main concept is to establish the weights of classifiers and train the data sample in each individual iteration to ensure accurate predictions of odd observations. Any machine learning algorithm that takes weights on the training set can be used as the basic classifier. Both AdaBoost and Random Forest are the same as they count the predictions given by each decision tree inside the forest to get the final classification. There are, nevertheless, some slight differences. The decision trees in AdaBoost have a depth of one (i.e. 2 leaves). Consequently, Each decision tree's predictions have different consequences on the model's final prediction. Rather of taking the average of each decision tree's predictions (or the majority in the case of classification), the AdaBoost method has each decision tree contribute a different amount to the final prediction.

The classifier was implemented using sklearn.ensemble package.

E. Random Forest Classifier

The Random Forest Classifier, as the name implies, is made up of a huge number of individual decision trees that work together as an ensemble. Each tree in the random forest produces a class prediction, and the class with the most votes becomes the prediction of our model. The low correlation across models is important because it allows them to make ensemble forecasts that are more accurate than any individual prediction because the trees protect each other from their particular errors.

The classifier was implemented using sklearn.ensemble package.

IV. EXPERIMENTAL RESULTS

For our supervised learning technique analysis, we've used Gaussian Naive Bayes, Logistic regression, and Decision Tree as the standard methods. As Ensemble methods, we have used Random Forest and AdaBoost Classifiers. In our research, we found that the Gaussian Naive Bayes classifier performed the poorest, whereas the Random Forest Classifier gave the best result in terms of every metric. As contained in fig1 & fig 2. It wasn't surprising to see the Random Forest classifier performing the best. The Decision Tree classifier performed better than Naive Bayes classifier and Logistic Regression. The Random Forest Classifier topped all performance metrics, as predicted, considering that it is an extension of the Decision Tree classifier, averaging out results from numerous recursions of the same problem.

 DOI : 10.35291/2454-9150.2022.0327

ISSN : 2454-9150 Vol-08, Issue-03, June 2022

The Metrics used for determining the performance of models are as follows:

1. Accuracy: Accuracy measures the amount of accurate or correct predictions made by the model. It is formulated as:

   Accuracy = (TP + TN) / T

2. Precision: Precision is the measure of bullying tweets correctly predicted by the algorithm. It is formulated as: Precision = TP / (TP + FP).

3. Recall: Recall is the ratio of how many bullying tweets, out of all available ones, are actually detected by the algorithm. It is formulated as:

   Recall = TP / (TP+FN).

4. F1-Score: Provides unbiased class-wise results. It considers false positives and false negatives and gives the weighted average of Precision and Recall. It is calculated as:

   F1 = 2*((Precision * Recall)/

   (Precision + Recall)).

5. ROCArea: Denotes the area under the curve formed by plotting TP rate.

Where:

TP = No. of True Positives

TN = No. of True Negatives

FP = No. of False Positives

FN= No. of False Negatives. Figure 1 and Figure 2 shows a graphical comparison between the aforementioned algorithms.

Note: Table 2&3 represent the weighted average using both the classes (Bullying and non-bullying) for Precision, Recall, and F1 score.

First column and row of the confusion matrices represents Cyberbullying class whereas the second row and column represents Non-cyberbullying class.

Table 2: Supervised Traditional Method

SUPERVISED TRADITIONAL METHODS			
	NaiveBayes	L.Regression	DecisionTree
Accuracy	0.62	0.80	0.85
Precision	0.79	0.81	0.88
Recall	0.62	0.80	0.85
F1-Score	0.59	0.81	0.85
ROCArea	0.68	0.81	0.87
Confusion Matrix	925 1504	1920 509	1896 533
	31 1541	274 1298	67 1505

Table 3: Supervised Ensemble Method

SUPERVISED ENSEMBLE METHODS			
	Adaboost		Random Forest
Accuracy	0.71		0.92
Precision	0.74		0.92
Recall	0.71		0.92
F1-Score	0.72		0.92
ROCArea	0.73		0.92
Confusion Matrix	1616813	2175254	
	3321240	731499	

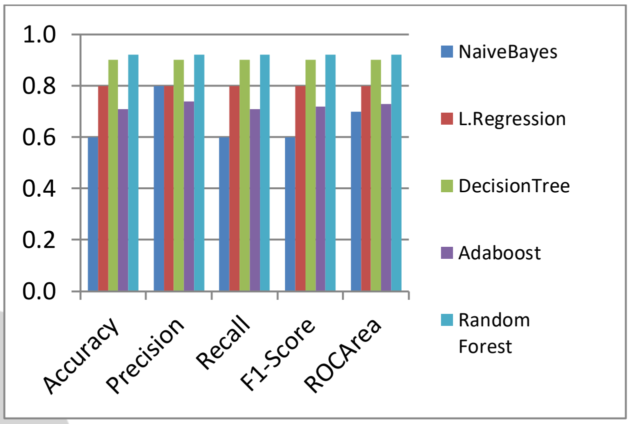

Figure 2: Precision, Recall, F1-Score & ROC Area

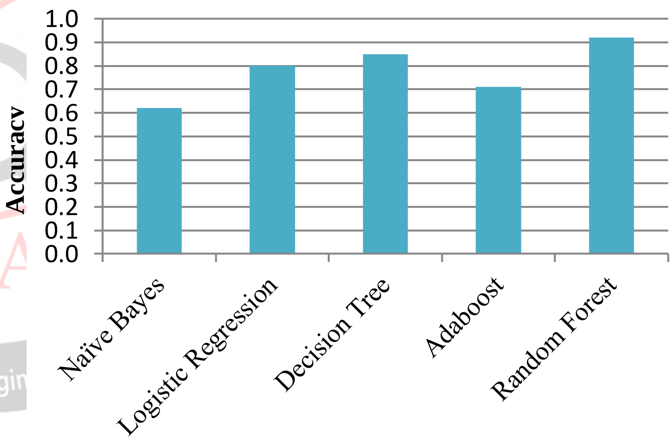

Figure 3: Graphical Representation of each Classifier's Accuracy

1. CONCLUSION AND FEATURE WORK

With the increasing prominence of social networking sites and growing social media usage by teenagers, cyberbullying has become increasingly common and has begun to create severe social issues. To prevent the harmful repercussions of cyberbullying, it is critical to build an automatic cyberbullying detection technique. Considering the significance of cyberbullying detection, in this research work, we did a comparative study between various Supervised algorithms and also comparing various Supervised Ensemble methods. The overall best performance was shown by Random Forest classifier, giving an accuracy of about 92%. The Ensemble methods

DOI : 10.35291/2454-9150.2022.0327

performed better than the supervised methods. Naïve Bayes performed the worst, giving just an approximated value of 62% accuracy.

For the future work, following are some observations made to improve the quality of the detection of cyberbullying content:

- According to our extensive review on the related literatures, most of the researches done previously on cyberbullying detection, were text based, so our next target is the development of multimedia (image, audio and video) based detection model and this can be achieved by switching from conventional machine learning approaches to deep learning techniques like CNN and DNN which are found good in dealing with any multimedia content.
- Our cyberbullying detection model is binary classification based (bullying or non-bullying), so multi-class classification approach could be also the direction of our future study.
- Lack of resources also led to our inability to analyze the performance of SVM (Support Vector Machine) and Multi-Layer Perceptron (Neural Networks) classifiers. They were nevertheless, mentioned in our study for references.

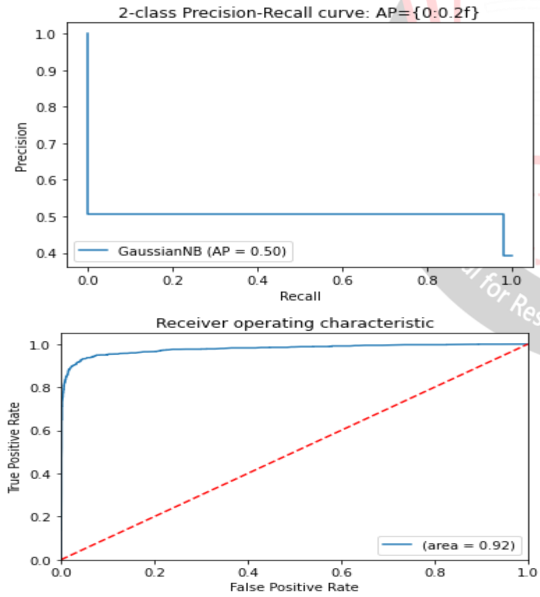

Figure 4: Naïve Bayes Classifier

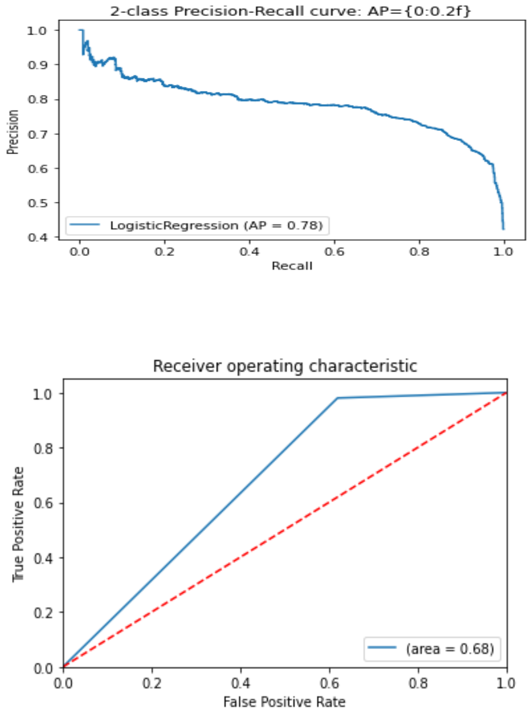

Figure 5: Logistic Regression

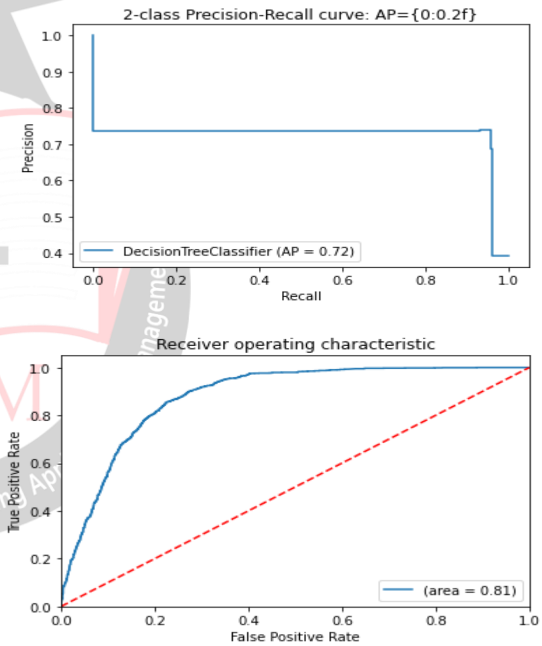

Figure 6: Decision Tree Classifier

 DOI : 10.35291/2454-9150.2022.0327

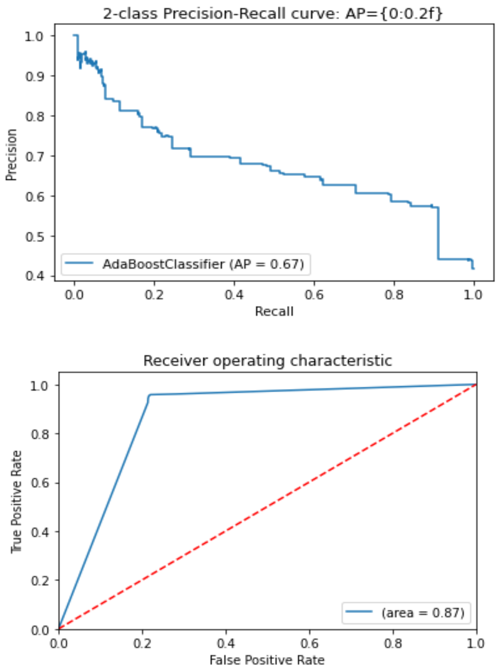

Figure 7: Adaboost Classifier

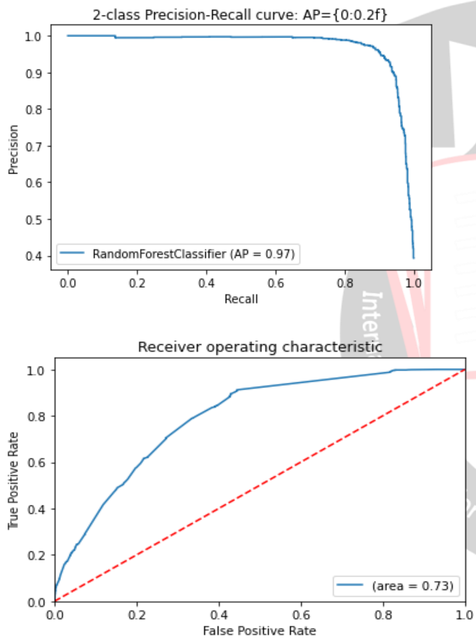

Figure 8: Random Forest Classifier

REFERENCES

[1] M. M. Islam, M. A. Uddin, L. Islam, A. Akter, S. Sharmin, and U. K. Acharjee, "Cyberbullying detection on social networks using machine learning approaches," *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, 2020.

[2] V. Jain, V. Kumar, V. Pal, and D. K. Vishwakarma, "Detection of cyberbullying on social media using machine learning," *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, 2021.

[3] S. Bastiaensens, H. Vandebosch, K. Poels, K. Van Cleemput, A. Desmet, and I. De Bourdeaudhuij, "Cyberbullying on social network sites. An experimental study into bystanders' behavioural intentions to help the victim or reinforce the bully," *Computers in Human Behavior*, vol. 31, pp. 259–271, 2014.

[4] D. L. Hoff and S. N. Mitchell, "Cyberbullying: Causes, effects, and remedies," *Journal of Educational Administration*, 2009.

[5] S. Hinduja and J. W. Patchin, "Bullying, cyberbullying, and suicide," *Archives of suicide research*, vol. 14, no. 3, pp. 206–221, 2010.

[6] Pawar, R., & Raje, R. R. (2019, May). Multilingual cyberbullying detection system. In *2019 IEEE International Conference on Electro Information Technology (EIT)* (pp. 040-044). IEEE.

[7] Jain, V., Kumar, V., Pal, V., & Vishwakarma, D. K. (2021, April). Detection of Cyberbullying on Social Media Using Machine learning. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 1091-1096). IEEE.

[8] Haidar, B., Chamoun, M., & Serhrouchni, A. (2017, October). Multilingual cyberbullying detection system: Detecting cyberbullying in Arabic content. In *2017 1st Cyber Security in Networking Conference (CSNet)* (pp. 1-8). IEEE.

[9] Alduailej, A. H., & Khan, M. B. (2017, September). The challenge of cyberbullying and its automatic detection in Arabic text. In *2017 International Conference on Computer and Applications (ICCA)* (pp. 389-394). IEEE.

[10] Kargutkar, S., & Chitre, V. Implementation of Cyberbullying Detection using Machine Learning Techniques.

[11] Van Hee, C., Jacobs, G., Emmery, C., Desmet, B., Lefever, E., Verhoeven, B., ... & Hoste, V. (2018). Automatic detection of cyberbullying in social media text. *PloS one*, *13*(10), e0203794.

[12] Çiğdem, A. C. I., Çürük, E., & Eşsiz, E. S. (2019). Automatic detection of cyberbullying in Formspring. me, Myspace and Youtube social networks. *Turkish Journal of Engineering*, *3*(4), 168-178.

[13] Al-Ajlan, M. A., & Ykhlef, M. (2018). Deep learning algorithm for cyberbullying detection. *International Journal of Advanced Computer Science and Applications*, *9*(9), 199-205.

[14] Muneer, A., & Fati, S. M. (2020). A comparison of machine learning algorithms for detecting cyberbullying on Twitter. *Future Internet*, *12*(11), 187.

[15] DataTurks, "Tweets dataset for detection of cyber-trolls," *Kaggle*, 12-Jul-2018. [Online]. Available: https://www.kaggle.com/dataturks/dataset-for-detection-of-cybertrolls. [Accessed: 16-May-2022].

 DOI : 10.35291/2454-9150.2022.0327