ISSN No:-2456-2165

Machine Learning based Cyber Bullying Detection

Kundharapu Vasudeva, Bestha Raghavendra Raj Kiran, Shaik Vaseem Akram, Bandaru Vijaya Prakash
Dept. of CSE
Madanapalle Institute Of Technology and Science

Abstract:- Cyber bullying is a serious issue that affects individuals of all ages, particularly children and teenagers who are more vulnerable to online harassment. With the growing use of social media and other online platforms, it has become increasingly important to develop effective methods to detect and prevent cyber bullying. In this project, we propose a machine learning-based approach for cyber bullying detection. The proposed system uses natural language processing (NLP) techniques to analyse text messages and identify patterns of abusive and aggressive behaviour. We apply various classification algorithms, such as Logistic Regression, Decision Trees Classifier and Gaussian Naïve bayes, to train our model and evaluate its performance. We also explore the use of ensemble methods, such as Random Forest classifier and adaboost classifier, to improve the accuracy of our model. We use publicly available datasets to test our system and compare its performance with other existing approaches. Our results show that the proposed machine literacy- grounded approach can effectively identify cyber bullying with high delicacy, perceptivity, and particularity. This project has significant implications for the development of automated systems that can help protect individuals from online harassment and promote a safer and more inclusive online environment.

Keywords:- *Cyberbullying, Harassment, Machine Learning, Natural Language Processing, social media analysis, Text classification, Logistic Regression, Decision Tree Classifier, Gaussian Naïve Bayes, Ensemble Methods, Adaboost classifier, Random Forest Classifier, Sentiment analysis and Behavioural analysis.*

I. INTRODUCTION

Cyber bullying is a type of online importunity that involves the use of electronic communication to bully, intimidate, or hang others. It can take various forms, such as sending threatening messages, sharing personal information without consent, spreading rumours, or posting insulting comments on social media platforms. Cyber bullying can have severe consequences, including depression, anxiety, low self-esteem, and even suicide. Thus, it's essential to descry and help cyber bullying to ensure the safety and well-being of individualities who use online platforms.

Traditional approaches to detecting cyber bullying involve manual monitoring of online platforms, which can be time- consuming and expensive. With the growing volume of online content, it is becoming increasingly challenging to monitor and moderate online platforms effectively. Therefore, there is a need for automated systems that can identify and flag potentially abusive content quickly and accurately.In recent years, machine learning techniques have shown great promise in detecting cyber bullying. These techniques use natural language processing (NLP) algorithms to analyse text messages and identify patterns of abusive and aggressive behaviour. A significant advantage of machine learning-based methods over traditional rule-based systems is their ability to adjust to evolving trends and patterns of cyberbullying, making them more efficient.

In this project, we propose a machine learning-based approach for cyber bullying detection. We aim to develop an automated system that can accurately detect and flag potentially abusive content on online platforms. We apply various classification algorithms, such as logistic regression, decision trees, and gaussian naïve bayes, to train our model and evaluate its performance. We also explore the use of ensemble methods, such as Random Forest classifier, to improve the accuracy of our model.In the next section, we provide a brief overview of related work in cyber bullying detection. We then describe our proposed machine learning-based approach in detail and discuss the datasets and evaluation metrics used in our experiments. We present and analyse the results of our experiments and compare our approach's performance with existing approaches. Finally, we conclude the project and discuss future work.

II. RELATED WORK

[1]Cyberbullying is a growing concern with the increased use of social media and online communities.Detecting and preventing cyberbullying is crucial in ensuring the mental and physical well-being of individuals, especially children and women. [2] To address this issue, various studies have proposed the use of machine learning and natural language processing techniques to automatically detect cyberbullying.

[3] In a study conducted in May 2022, the authors proposed the use of Support Vector Machines (SVM) to identify cyberbullying in Twitter, and Optical Character Recognition (OCR) to detect image-based cyberbullying. [4] They categorized existing approaches into four main classes, including supervised learning, lexicon-based, rule-based, and mixed-initiative approaches.

[5] Another study conducted in December 2021 highlighted the research gap in resource-poor languages such as Roman Urdu, which is widely used in South Asian countries. The authors performed extensive pre-processing on the Roman Urdu microtext, including the creation of a slang-phrase dictionary and elimination of cyberbullying domain-specific stop words.[6] They experimented with different models, including RNN-LSTM, RNN-BiLSTM, and CNN models, achieving validation accuracy of up to 85.5%.

ISSN No:-2456-2165

[7] A study from April 2021 proposed a technique to detect online abusive and bullying messages using natural language processing and machine learning. The study used Bag-of-Words (BoW) and term frequency-inverse text frequency (TFIDF) features and evaluated the accuracy level of four machine learning algorithms.[8] The authors concluded that their proposed technique can accurately detects instances of cyberbullying language on social media platforms.

[9] Other existing systems include cyberbullying detection using deep transfer learning, cyberbullying identification system based on deep learning algorithms, and social media cyberbullying detection using machine learning. [10] While some studies have achieved promising results, the detection of cyberbullying remains a challenging task due to the ever-evolving nature of language and the complexity of social dynamics on online platforms.

Overall, the proposed techniques and algorithms for cyberbullying detection provide a foundation for ongoing research efforts to combat cyberbullying and online harassment effectively.

III. OTHER EXISTING SYSTEMS

- Deep Learning Algorithm for Cyberbullying Detection
- Social Media Cyberbullying Detection using Machine Learning
- Cyber-Bullying Detection using Machine Learning Algorithms
- Detection of Cyberbullying on social media Using Machine learning
- Cyberbullying Detection on Social Networks Using Machine Learning Approaches
- Cyberbullying Identification System Based Deep Learning Algorithms
- Cyberbullying detection using deep transfer learning.

IV. PROPOSED SYSTEM

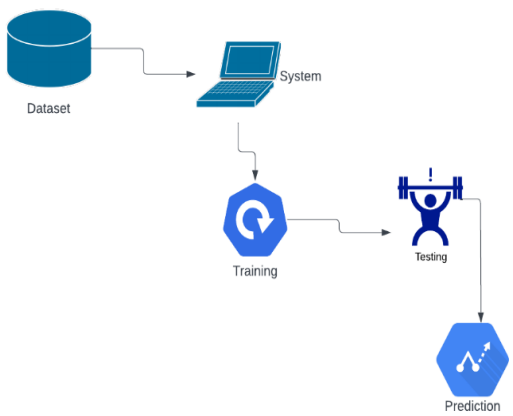


Fig. 1: Architecture

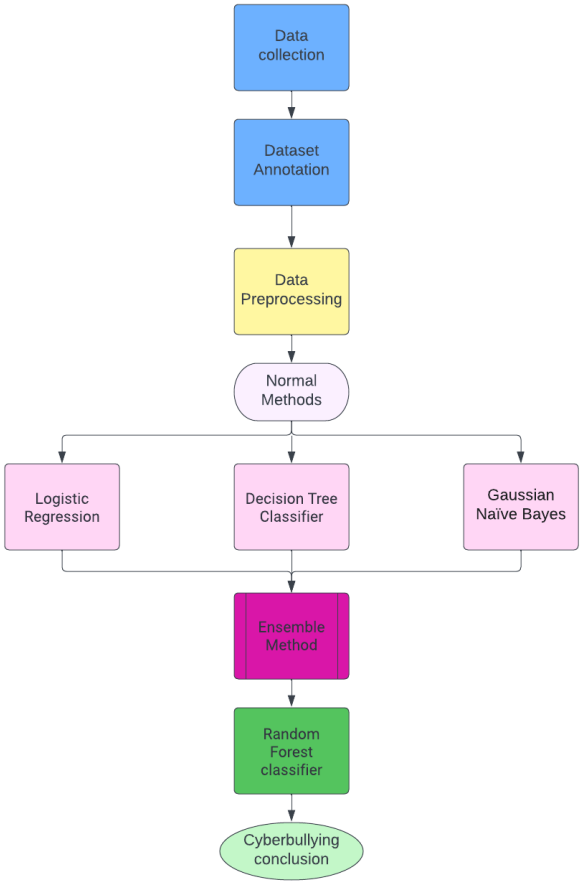


Fig. 2: Proposed System Model

- **Data Collection:** The first step in building a machine learning model for cyberbullying detection is to collect a dataset of text examples that have been labelled as either containing cyberbullying or not.This dataset can be collected manually by experts or using automated tools that scan social media platforms for cyberbullying-related posts.
- **Data Pre-processing:** Once the dataset is collected, the next step is to pre-process the text data.This includes tasks such as tokenization, removing stop words, stemming or lemmatization, and converting text into numerical representations such as word embeddings or Bag of Words.
- **Model Selection:** After pre-processing the data, the next step is to select an appropriate machine learning model for the task of cyberbullying detection. As mentioned earlier, anensemble method such as a Random Forest classifier is used for this project.
- **Model Training**: The selected model is then trained on the pre-processed data.The training process involves feeding the model, the pre-processed data along with their corresponding labels and updating the model parameters to minimize the classification error.
- **Model Evaluation:**After training the model, it's important to evaluate its performance on a separate dataset to measure its accuracy, precision, recall, and F1 score.If the model performance is not satisfactory, the model hyperparameters can be adjusted, or a different model architecture can be selected.

ISSN No:-2456-2165

V. BLOCK DIAGRAM

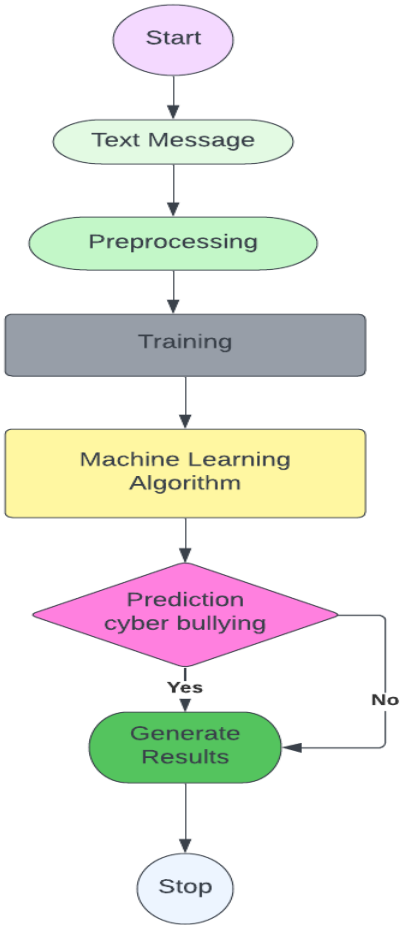


Fig. 3: Block Diagram

VI. METHODOLOGY AND ALGORITHMS

A. Logistic regression:

Logistic Regression is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used performing text classification using the TF-IDF method and random oversampling. Here is an overview of what this code does:

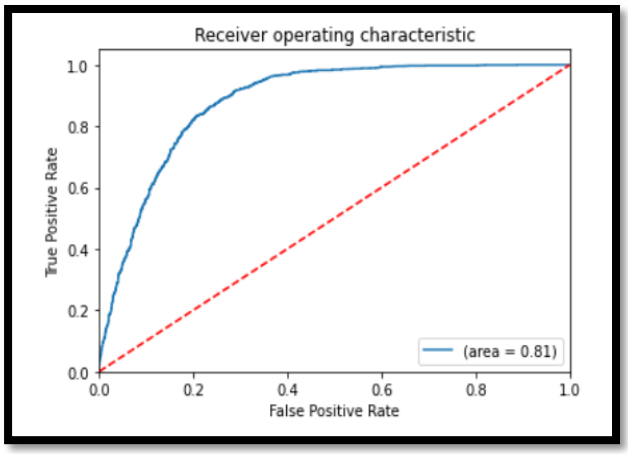
A logistic regression classifier (lgr) is instantiated.The random oversampled data (X_over, y_over) is used to train the classifier using the fit () method.The classifier is used to predict the target variable (y_pred) for the testing data (X_test) using the predict () method.The accuracy of the classifier is calculated using the accuracy_score () method from the Scikit-learn metrics module.The confusion matrix of the classifier's predictions is calculated using the confusion_matrix () method from the Scikit-learn metrics module.The getStatsFromModel () function is called to calculate and print additional evaluation metrics such as precision, recall, and F1-score.

The performance of a logistic regression classifier that was trained using the random oversampled data. The confusion matrix provides a detailed breakdown of the true positive, true negative, false positive, and false negative predictions, while the additional evaluation metrics provide more insights into the classifier's performance, including its ability to detect cyber bullying accurately. Depending on the performance metrics, further adjustments may be needed to optimize the model, such as using a different algorithm or tweaking the hyper parameters.

```
Accuracy:  0.8077980504873782
Confusion Matrix:
 [[1920  509]
 [ 260 1312]]
              precision    recall  f1-score   support

           0       0.88      0.79      0.83      2429
           1       0.72      0.83      0.77      1572

    accuracy                           0.81      4001
   macro avg       0.80      0.81      0.80      4001
weighted avg       0.82      0.81      0.81      4001
```



B. Naive Bayes classifiers:

Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems. The performance of a Gaussian Naive Bayes classifier for detecting cyberbullying. Here is an overview of what this code does:A Gaussian Naive Bayes classifier (gnb) is instantiated.The random oversampled data (X_over, y_over) is used to train the classifier using the fit () method.The classifier is used to predict the target variable (y_pred) for the testing data (X_test) using the predict () method.

The score of the classifier is calculated using the score () method from the Gaussian Naive Bayes model.The confusion matrix of the classifier's predictions is calculated using the confusion_matrix () method from the Scikit-learn metrics module.The getStatsFromModel () function is called to calculate and print additional evaluation metrics such as precision, recall, and F1-score.

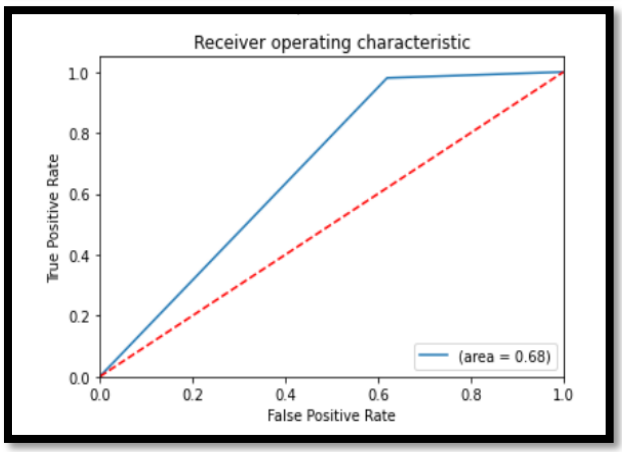
The performance of a Gaussian Naive Bayes classifier that was trained using the random oversampled data. The score indicates the percentage of correctly classified instances in the testing data. The confusion matrix provides a detailed breakdown of the true positive, true negative, false positive, and false negative predictions, while the additional evaluation metrics provide more insights into the classifier's performance, including its ability to detect cyberbullying accurately.

The effectiveness of the Gaussian Naive Bayes classifier in detecting cyberbullying can be evaluated based on these performance metrics. Depending on the performance metrics, further adjustments may be needed to optimize the model, such as using a different algorithm or tweaking the hyperparameters.

```
Score: 0.6160959760059985
Confusion Matrix:
 [[ 924 1505]
 [  31 1541]]
              precision    recall  f1-score   support

           0       0.97      0.38      0.55      2429
           1       0.51      0.98      0.67      1572

    accuracy                           0.62      4001
   macro avg       0.74      0.68      0.61      4001
weighted avg       0.79      0.62      0.59      4001
```



C. Decision Trees Classifier:

The performance of a decision tree classifier for detecting cyberbullying. Here is an overview that can be included in a report:

The decision tree classifier (dtc) is instantiated.

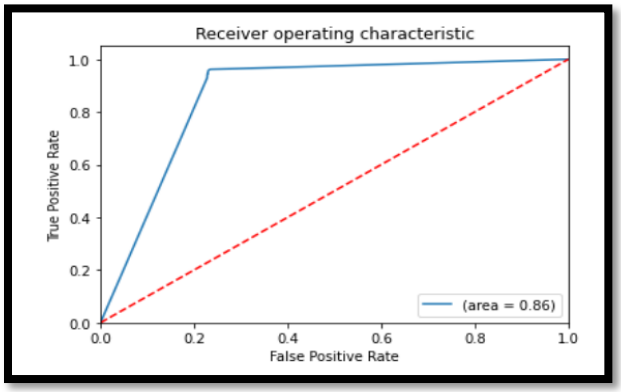
The random oversampled data (X_over, y_over) is used to train the classifier using the fit () method.The classifier is used to predict the target variable (y_pred) for the testing data (X_test) using the predict () method.The accuracy of the classifier is calculated using the accuracy_score () method from the Scikit-learn metrics module. The confusion matrix of the classifier's predictions is calculated using the confusion_matrix () method from the Scikit-learn metrics module. The get Stats From Model () function is called to calculate and print additional evaluation metrics such as precision, recall, and F1-score.

The accuracy score and the confusion matrix can be presented to provide an overview of the classifier's performance. The accuracy score indicates the percentage of correctly classified instances in the testing data. The confusion matrix provides a detailed breakdown of the true positive, true negative, false positive, and false negative predictions, which can help in understanding the classifier's strengths and weaknesses. Additionally, the evaluation metrics such as precision, recall, and F1-score can be reported to provide a more detailed analysis of the classifier's performance. These metrics can help in identifying which class is being misclassified more often and can provide insights into areas where the model can be improved. The decision tree classifier can be evaluated based on these performance metrics to determine its effectiveness in detecting cyberbullying. Depending on the performance metrics, further adjustments may be needed to optimize the model, such as using a different algorithm or tweaking the hyperparameters.

```
Accuracy:  0.8430392401899525
Confusion Matrix:
 [[1863  566]
 [  62 1510]]
              precision    recall  f1-score   support

           0       0.97      0.77      0.86      2429
           1       0.73      0.96      0.83      1572

    accuracy                           0.84      4001
   macro avg       0.85      0.86      0.84      4001
weighted avg       0.87      0.84      0.84      4001
```



D. Random Forest:

The performance of a random forest classifier for detecting cyberbullying. Here is an overview of what this code does:A random forest classifier (rfc) is instantiated, with verbose set to True to show the training progress.The random oversampled data (X_over, y_over) is used to train the classifier using the fit() method.The classifier is used to predict the target variable (y_pred) for the testing data (X_test) using the predict () method.

The score of the classifier is calculated using the score () method from the random forest model.

The confusion matrix of the classifier's predictions is calculated using the confusion_matrix () method from the Scikit-learn metrics module.The getStatsFromModel () function is called to calculate and print additional evaluation metrics such as precision, recall, and F1-score.

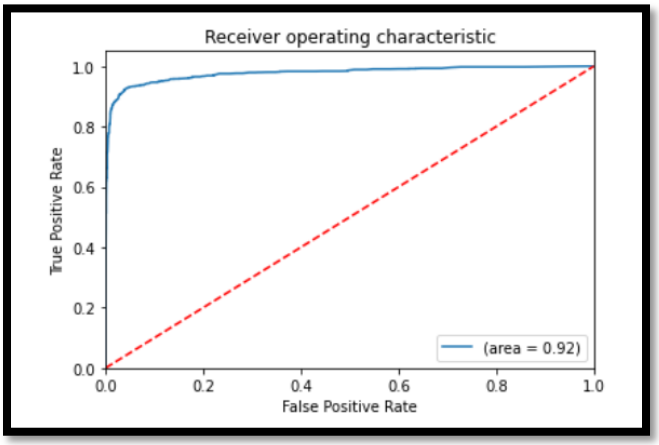
The performance of a random forest classifier that was trained using the random oversampled data. The score indicates the percentage of correctly classified instances in the testing data. The confusion matrix provides a detailed breakdown of the true positive, true negative, false positive, and false negative predictions, while the additional evaluation metrics provide more insights into the classifier's performance, including its ability to detect cyberbullying

accurately.The effectiveness of the random forest classifier in detecting cyberbullying can be evaluated based on these performance metrics. Depending on the performance metrics, further adjustments may be needed to optimize the model, such as using a different algorithm or tweaking the hyperparameters.

```
Score: 0.9127718070482379
Confusion Matrix:
 [[2158  271]
 [  78 1494]]
              precision    recall  f1-score   support

           0       0.97      0.89      0.93      2429
           1       0.85      0.95      0.90      1572

    accuracy                           0.91      4001
   macro avg       0.91      0.92      0.91      4001
weighted avg       0.92      0.91      0.91      4001
```



VII. RESULT

Based on the all algorithms used in this project, Random Forest Algorithm give more accuracy, more precision and support. So, we used Random Forest for the predict of Cyberbully.

Table 1: Algorithm Result

S.No.	Algorithm Name	Accuracy
1	Logistic Regression	81
2	Decision Tree	84
3	Gaussian Naïve Bayes	62
4	Random Forest Classifier	92

VIII. CONCLUSION AND FUTURE SCOPE

The research paper compares various supervised machine learning algorithms and ensemble methods for detecting cyberbullying. According to the study, the Random Forest classifier performed the best with a 92% accuracy rate while Naive Bayes was the least accurate with only a 61% accuracy rate. The future scope of the project is, to implement in real time and collaboration with companies.

REFERENCES

[1.] Miss. Jafri S A., Miss Wagh Roshani B., Miss Gaikwad V. Subodh., Miss Sonawane U D, International Research Journal of Modernization in Engineering Technology and Science

[2.] https://www.irjmets.com/uploadedfiles/paper/issue_5_may_2022/24749/final/fin_irjmets1653789970.pdf

[3.] Amirita Dewani, Mohsin A M., Sania B., "Journal of Big Data " , Cyberbullying detection: advanced pre-processing techniques & deep learning architecture for Roman Urdu data- Dec 2021.

[4.] https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00550-7

[5.] Md Manowarul., Md Ashraf., Linta., Arnisha., Selina., Uzzal., ''Cyberbullying Detection on Social Networks Using Machine Learning Approaches", 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE).

[6.] https://ieeexplore.ieee.org/document/9411601/authors#authors

[7.] Akhter A., Islam L., Uddin A Md., Islam M., "Cyberbullying Detection on Social Networks Using Machine Learning Approaches" Apr -2021

[8.] https://www.researchgate.net/publication/351131976_CSyberbullying_Detection_on_Social_Networks_Using_Machine_Learning_Approaches

[9.] Giovanni B., Chanhee Shin, Nishal K., ". Cyberbullying Detection System (JUN 2020)"

[10.] https://engineering.ucdenver.edu/docs/librariesprovider29/college-of-engineering-and-applied-science/sp2020-capstone/csci14-report.pdf?sfvrsn=d3731fb9_2

[11.] Monirah AAA., Mourad Y., "International Journal of Advanced Computer Science and Applications (IJACSA)", 2018.

[12.] John H., Mohamed N., Mostafa A., Zeyad E., Eslam A., Ammar M., "International Journal of Advanced Computer Science and Applications (IJACSA)", 2019.

[13.] Mangala K., Anvitha K., Deepa, Deepika K V., Divya C H, "Cyber-Bullying Detection using Machine Learning Algorithms ", "IJCRT".

[14.] https://jpinfotech.org/detection-of-cyberbullying-on-social-media-using-machine-learning/

[15.] https://www.irjet.net/archives/V9/i5/IRJET-V9I5562.pdf

[16.] https://www.mdpi.com/2079-9292/11/20/3273/pdf

[17.] https://link.springer.com/article/10.1007/s40747-022-00772-z