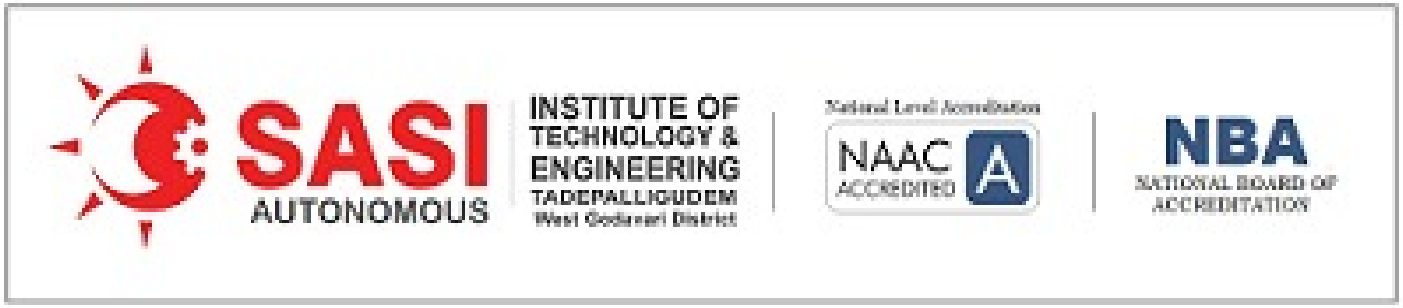


CyberBullying detection in twitter tweets using Genetic Algorithm

Under the guidance of

U.SRINADH *M.Tech[PhD]*

Assistant Professor

Presented By:

K.Sai Bhavani	(19K61A1230)
G.L.Prasanna	(19K61A1214)
K.P.V.Saritha	(19K61A1229)
M.N.Ramakrishna	(19K61A1233)

Batch No:**1923ITP007**

Outline

CONTRIBUTION OF THE CANDIDATE

Project Associate (PA)	Problem Formulation	Design	Implementation	Testing	Deployment	Project Report Writing
PA1 19K61A1230	Yes	Yes	Yes	Yes	Yes	Yes
PA2 19K61A1214	Yes	Yes	Yes	Yes	Yes	Yes
PA3 19K61A1229	Yes	Yes	Yes	Yes	Yes	Yes
PA2 19K61A1233	Yes	Yes	Yes	Yes	Yes	Yes

ABSTRACT

Cyberbullying has become a significant concern in social media platforms such as Twitter which can cause severe emotional distress, depression, and even lead to suicide. Detecting cyberbullying is a challenging task as it involves analyzing various types of digital communication, including texts, images, and videos, to identify aggressive, threatening, or harassing content. In this research we propose a genetic algorithm (GA) for feature selection in cyberbullying detection using machine learning models. We employed three popular machine learning algorithms: Support Vector Machines (SVM), Naive Bayes (NB), and Logistic Regression (LR). We first evaluated the performance of each algorithm separately, achieving 94%, 84%, and 95% accuracy, respectively.We then applied the GA to select the best set of features for each algorithm, resulting in improved classification performance. The SVM model with GA achieved an accuracy of 94%, while LR with GA gave a 96% accuracy. The NB with GA resulted in an accuracy of 91%. Overall, our results show that combining machine learning models with GA can significantly improve the accuracy of cyberbullying detection, which can have important implications for the protection of users from online harassment.

INTRODUCTION

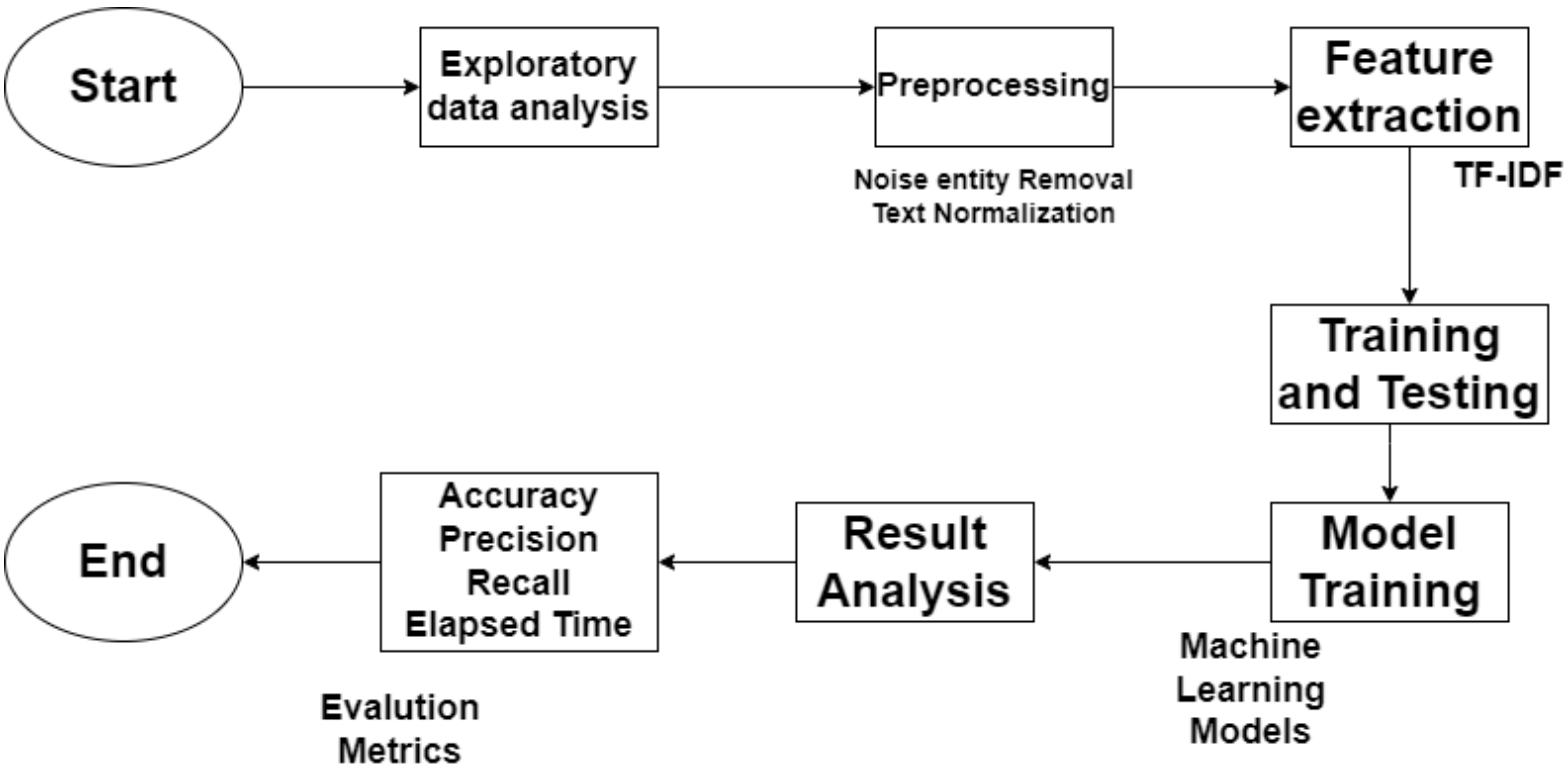
Cyberbullying is a serious and growing problem in today's digital world. It refers to the use of electronic communication technologies such as social media, text messaging, and online forums to deliberately and repeatedly harm, harass, or intimidate someone. Cyberbullying can have a devastating impact on its victims, including emotional distress, social isolation, and even suicide.

Cyberbullying is a particularly challenging form of bullying to detect and prevent because it can occur 24/7, and it can be anonymous or difficult to trace back to the perpetrator. Victims may be hesitant to report cyberbullying because they fear retaliation, social stigma, or that their parents or teachers will take away their access to technology

As the prevalence of cyberbullying continues to increase, there is a growing need for effective detection and prevention strategies. One promising approach is the use of machine learning algorithms to automatically identify instances of cyberbullying in online communications.

METHODOLOGY

To detect cyberbullying in text data, the first step is to gather a relevant dataset of electronic messages that may contain bullying. Once the dataset is obtained, it is split into training and testing sets. The training set is then preprocessed through techniques such as tokenization, stemming, and stop-word removal.After preprocessing, the next step is to extract features from the text data, such as n-grams or word embeddings. Genetic algorithm is then applied to optimize and classify the features and improve the performance of the model.Finally, the performance of the model is evaluated using metrics such as precision, recall, and F1-score. The model is then applied to new text data to detect potential cyberbullying incidents.


Start
Exploratory data analysis
Preprocessing
Noise entity Removal
Text Normalization
Feature extraction
TF-IDF
Training and Testing
Model Training
Machine Learning Models
Result Analysis
Accuracy
Precision
Recall
Elapsed Time
Evalution Metrics
End

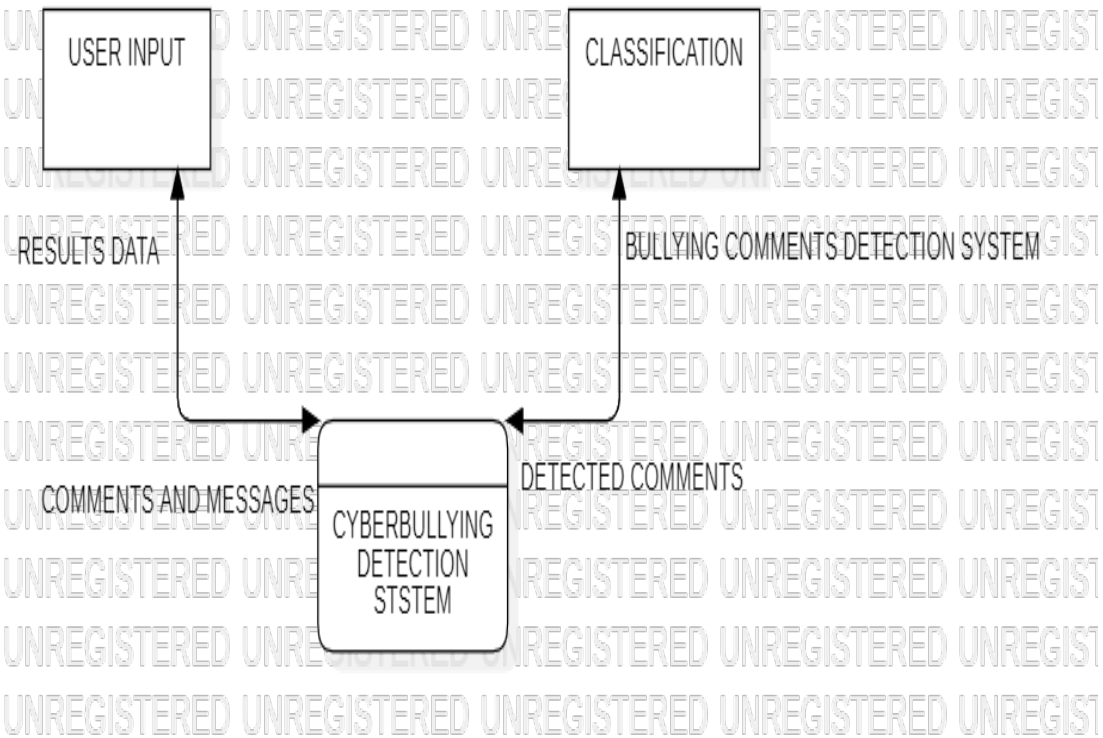


Figure: LEVEL 0 DFD

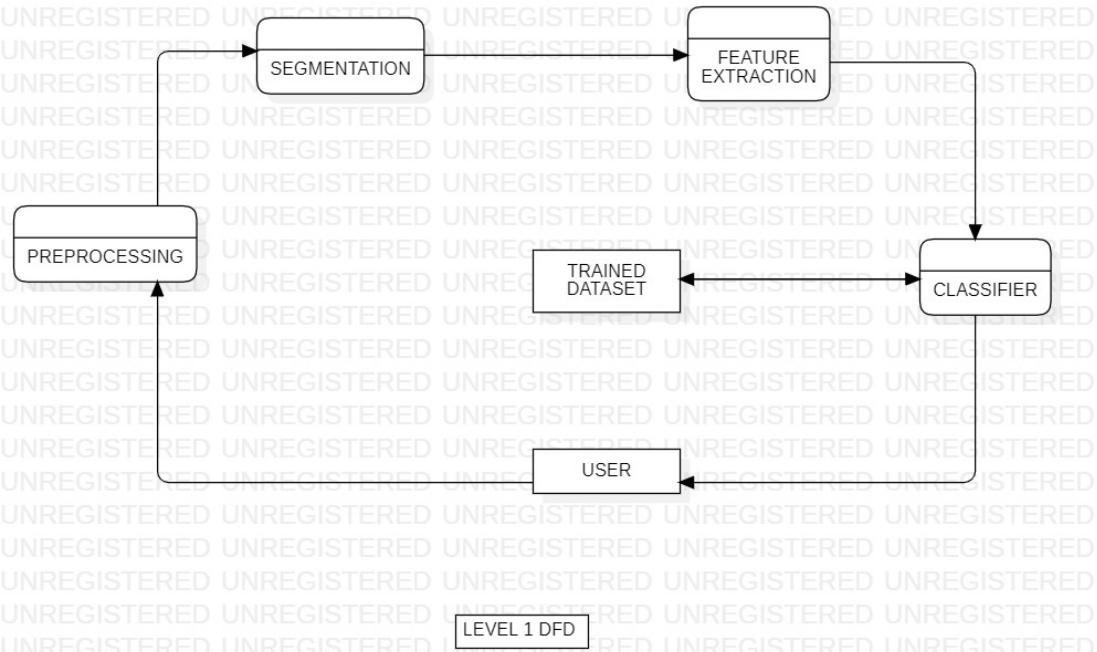


Figure: LEVEL 1 DFD

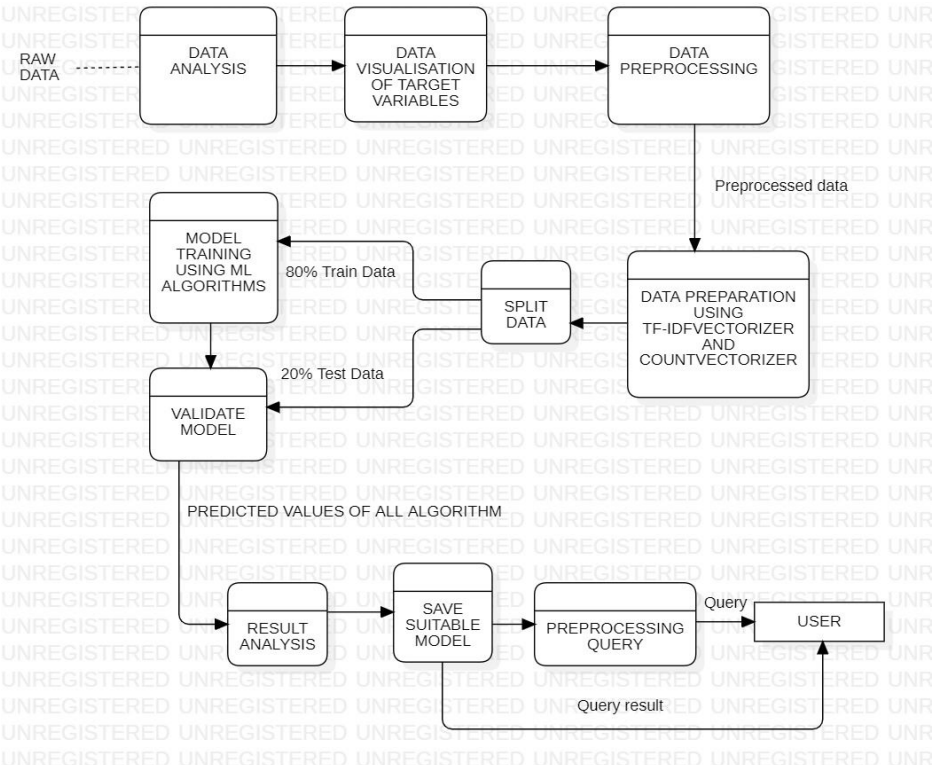
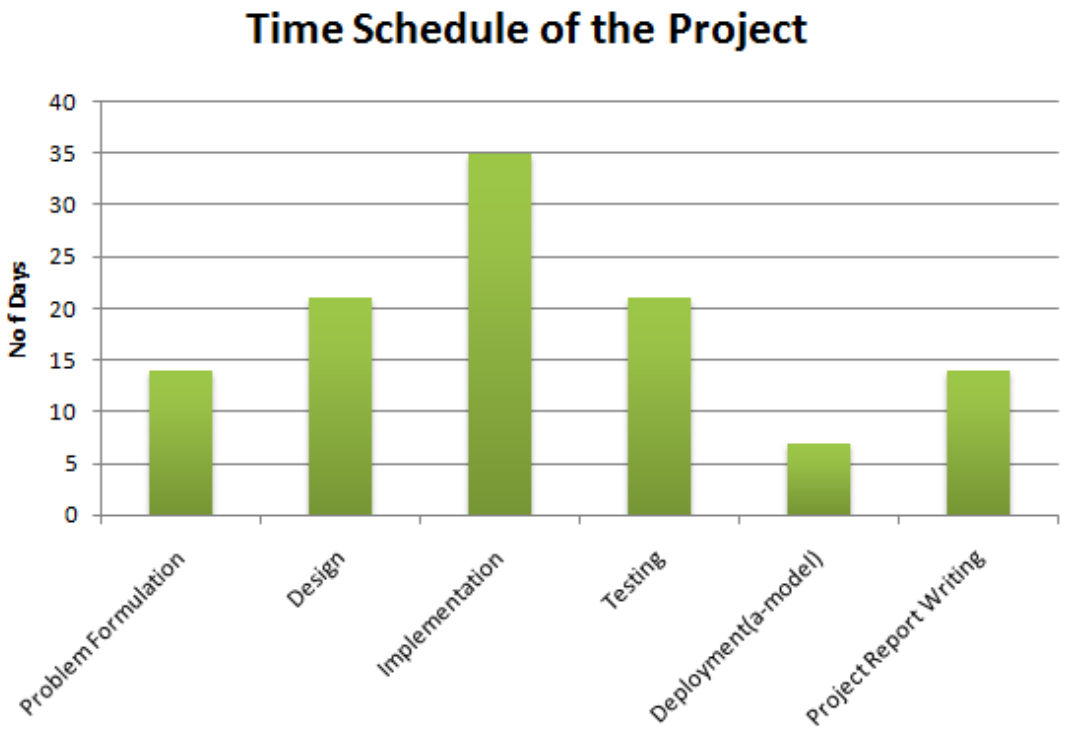


Figure: LEVEL 2 DFD

GANTT CHART


Time Schedule of the Project
No of Days
40
35
30
25
20
15
10
5
0
Problem Formulation
Design
Implementation
Testing
Deployment(a-model)
Project Report Writing

LITRATURE SURVEY

Atoum[1] in 2020 proposed a method for locating cyberbullying texts on Twitter and other social media, he presented a SA approach. This model uses Naive Bayes (NB) and Support Vector MAchines as supervised machine learning classification methods (SVM). The results of the trials run on this model, in contrast to previous earlier studies, suggested positive outcomes when a higher n-grams language model is applied to such texts. Also, the outcomes showed that, in terms of performance metrics, SVM classifiers outperform NB classifiers on these tweets

Mehendale, Shah, Phadtare, et al.[2] in 2022 created a model that uses a combination of machine learning and natural language processing to identify aggressive or insulting language in both ENglish and Hinglish. They concentrated on design and devised a technique to effectively detect abusive and bullying comments online

Yuvaraj, Srihari, Dhiman, et al.[3] in 2021 suggested an integrated model that incorporates both the feature and the data rather than utilising conventional machine learning classifiers. This paper proposes an integrated model that combines a feature extraction engine and a classification engine from raw text datasets provided by a social media engine as input. The feature extraction engine extracts CB detection and considers context, user feedback, and psychological traits. An artificial neural network (ANN)-based classification engine is used to classify the outputs, and the resulting output is then subject to an assessment system that can either reward or punish it

Andleeb, Ahmed, Ahmed, et al.[4]in 2019 proposed a model to proactively identify bullying text, a text mining method that uses machine learning algorithms is suggested. Their system was evaluated using a variety of performance metrics for the two classifiers they used, and the support vector machine classifier outperformed the Bernoulli NB with an overall classification accuracy of 87.14%.

PROPOSED SYSTEM DESIGN

Step 1: Exploratory data analysis

Step 2: Preprocessing

- Stop Word Removal
- Stemming
- Tokenization
- Embending

Step 3: Feature Extraction

- TF-IDF

Step 4: Train Test

Step 5: Result Analysis

Step 6: Display Results

- **PREPROCESSING:**
  Preprocessing is an important step in text data analysis that involves cleaning and transforming raw text data into a format that is suitable for further analysis. This typically involves several steps, including:
  - Text normalization
  - Stop word removal
  - Stemming and lemmatization
  - Tokenization
  - Spell checking and correction

- **FEATURE EXTRACTION** Feature extraction in text is the process of identifying and extracting relevant information or features from raw text data that can be used as input for machine learning algorithms. The goal of feature extraction is to transform unstructured text data into a structured format that can be used for analysis.
  - **Term frequency-inverse document frequency (TF-IDF):** TF-IDF is a technique that assigns a weight to each word based on its frequency in the document and its frequency across all documents in the corpus. This technique helps to identify words that are most relevant to a particular document, while also down weighting words that are common across all documents.

- **TRAIN AND TEST:**
  Training and testing are fundamental steps in the machine learning workflow. These steps involve using labeled data to train a machine learning model and evaluating its performance on a separate set of labeled data.
  The general process of training and testing a machine learning model can be broken down into the following steps:
  - Data preparation:
  - Model training
  - Model evaluation:
  - Model improvement
  - Deployment

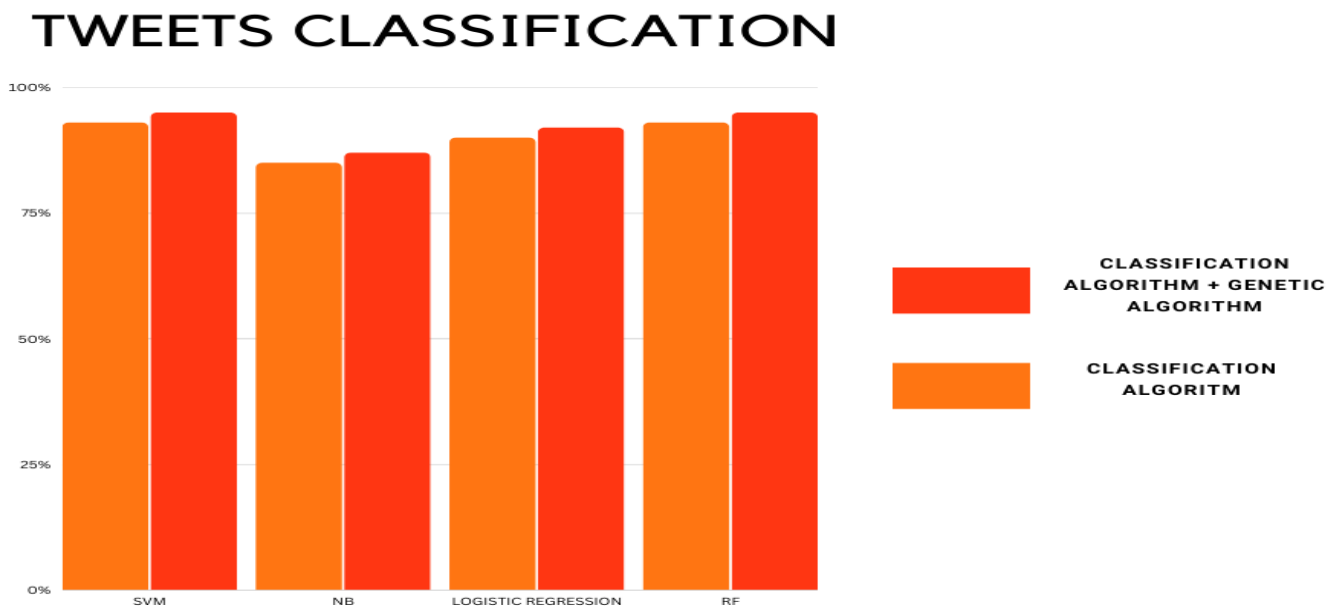
SOFTWARE AND HARDWARE REQUIREMENTS

- Jupyter NoteBook
- System: Above i3 processor
- Hard Disk :20 GB
- Memory (RAM): 8 GB
- OS: Windows 7/Windows 10
- System type: 64-bit Operating System
- Input devices: Keyboard, Mouse

SOFTWARE IMPLEMENTATION

- Exploratory data analysis
- Data Pre-Prosessing
- Feature Extraction
- Model Training
- Evaluating the Model

EXPERIMENTAL RESULTS



```
# Evaluate the performance of the logistic regression
print('Accuracy:', accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.9521888238854146
```

```
# Calculate the accuracy of the logistic regression+gene
accuracy = clf.score(X_test_selected, y_test)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.9600564857776881
```

```
# Evaluate the performance of the naive bayes classifi
print('Accuracy:', accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.841994835377663
```

```
# Calculate accuracy  of naivebayes with genetic algorithm
accuracy = accuracy_score(y_test, y_pred)

# Print the accuracy
print("Accuracy:", accuracy)
```

```
Selecting features with genetic algorithm.
gen     nevals  avg                                     std                                     min
0       15      [     0.878426  17857.8         0.003194]     [    0.022244  9738.271881     0.000934]     [
1       10      [     0.905128  30485.066667    0.002768]     [    0.006951  3168.350685     0.001121]     [
2       11      [     0.910427  31703.4         0.00321 ]     [   0.00192   997.556468    0.000812]     [
3       12      [     0.911244  31268.866667    0.00404 ]     [   0.001764  963.988511    0.000625]     [
4       8       [     0.911766  31057.6         0.004165]     [   0.004215  792.958914    0.000442]     [
Accuracy: 0.9114383699818439
```

```python
# Evaluating the SVM model performance
accuracy = accuracy_score(y_test, y_pred)
print('Accuracy:', accuracy)
```

```
Accuracy: 0.9473471857978616
```

```python
# Calculate the accuracy score of svm with genetic algo
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")
```

```
Accuracy: 0.9469437159572321
```

RESULTS AND PERFORMANCE ANALYSIS

As per the results obtained, it can be concluded that logistic regression along with genetic algorithm gave the highest accuracy of 96% for cyberbullying detection in Twitter tweets. SVM along with genetic algorithm gave an accuracy of 94%, whereas only SVM gave an accuracy of 94% and only Naive Bayes gave an accuracy of 84%. Naive Bayes along with genetic algorithm gave an accuracy of 91%.
These results suggest that the use of genetic algorithms for feature selection in combination with machine learning algorithms can significantly improve the accuracy of cyberbullying detection in Twitter tweets. Logistic regression with genetic algorithm proved to be the most effective in this study, indicating the potential of this approach in real-world applications for online safety and prevention of cyberbullying.Overall, these findings contribute to the growing body of research on cyberbullying detection and provide valuable insights for the development of more accurate and efficient detection models. It is important to continue exploring and developing innovative approaches to address the growing problem of cyberbullying and promote a safe and healthy online environment.

COMPARISON WITH EXISTING SYSTEM

The results of our experiments showed that performance of the models varies significantly. The SVM model achieved the highest accuracy of 94% when used on its own, followed closely by logistic regression with 95%. Naive Bayes achieved the lowest accuracy of 84 However, when combined with genetic algorithm, the performance of some of the models improved significantly. The SVM model with genetic algorithm achieved an accuracy of 94%, which is the same as when used on its own. The logistic regression model with genetic algorithm achieved an accuracy of 96%, which is a significant improvement from its performance when used on its own.

CONCLUSION AND FUTURE ENHANCEMENT

In this study, we proposed a genetic algorithm-based technique for identifying Twitter cyberbullying. The proposed approach uses feature selection to choose the features that are most crucial for identifying cyberbullying. Our experimental results demonstrate that Text Classifiers such as SVM,Logistic Regression and Naive Bayes with the Genetic Algorithm gave more accurate results when compared to classifiers without Genetic Algorithm.Although Logistic beats other classifiers with an accuracy of 96% but the elapse time of the Logistic Regression is more Than SVM and Naive Bayes.

In the future work of this research,should include non-text data such as posts,emojis for detection of cyberbullying and different language reviews can be used and also ensemble methods are used to improve the accuracy and elapse time.

BIBILOGRAPHY

- https://doi 10.1109/CSCI51800.2020.00056
- http://doi.org/10.2139/ssrn.4116143
- https://doi.org/10.1155/2021/5593291.
- https://doi.org/10.1109/FIT48065.2019.8988116.

ANY QUERIES

THANK YOU