"Cyberbullying Detection Using Machine Learning Techniques"

A Report submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNVERSITY ANANTAPUR.

In Partial Fulfillment of the Requirements for the Award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING
BY

V. NAGESH	**19121A05R0**
V.GANGA PRADEEP	**19121A05R3**
V. HEMANTH KUMAR REDDY	**19121A05R7**
V. SINDHURA	**19125A05R9**
T. CHIRUDEEP	**19121A05P3**

Under the Guidance of

Dr. M. Sakthivel
Professor
Dept of CSE, SVEC

Department of Computer Science and Engineering
SREE VIDYANIKETHAN ENGINEERING COLLEGE
(Affiliated to JNTUA, Anantapuramu)
Sree Sainath Nagar, Tirupathi – 517 102

2019-2023

SREE VIDYANIKETHANENGINEERINGCOLLEGE

(Affiliated to Jawaharlal Nehru Technological University Anantapur)
Sree Sainath Nagar, A. Rangampet, Tirupati – 517 102, Chittoor Dist., A.P.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project Work entitled

"Cyberbullying Detection Using Machine Learning Techniques "

is the bonafide work done by

V. NAGESH	**19121A05R0**
V.GANGA PRADEEP	**19121A05R3**
V. HEMANTH KUMAR REDDY	**19121A05R7**
V. SINDHURA	**19125A05R9**
T. CHIRUDEEP	**19121A05P3**

In the Department of Computer Science and Engineering, Sree Vidyanikethan Engineering College, A.Rangampet. is affiliated to JNTUA, Anantapuramu in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering during 2019-2023.

This is work has been carried out under my guidance and supervision.

The results embodied in this report have not been submitted in any University or Organization for the award of any degree or diploma.

Internal Guide

Dr. M. Sakthivel
Professor
Dept of CSE
Sree Vidyanikethan Engineering College
Tirupathi

Head

Dr. B. Narendra Kumar Rao
Prof & Head
Dept of CSE
Sree Vidyanikethan Engineering College
Tirupathi

INTERNAL EXAMINER

EXTERNAL EXAMINER

TABLE OF CONTENTS

ABSTRACT

Research on cyberbullying detection is gaining increasing attention in recent years as both individual victims and societies are greatly affected by it. Access to social media is very easy and the mistreatment of people through informal words and comments bullying, sexism, racism, aggressive content, harassment, toxic comment, etc has increased. Hence there needs to be a monitoring system established to control and reduce these types of mis behaviour and cyberbullying that spread around social networks which has inspired to development of this research to automate the detection process of cyberbullying. Our major motivation is to build a model that categorizesoffensive and non-offensive. For this process, we have developed a machine-learning algorithm that uses Random Forest and J48 for feature extraction anddetection analysis of Twitter data.

Keywords: Cyber Bullying, Sentiment Analysis, Behavioural Analysis.

INTRODUCTION

Cyberbullying is known as an unnoticed crime and there are many victims who havebeen bullied online through toxic comments. Some statistics show that as many as 50% of children have experienced cyberbullying. Victims of cyberbullying can experience wide-ranging effects, including mental health issues, poor academic performance, a desire to drop out of school, and even suicidal ideation. Even though many victims face this, many of them are not being registered because of social status and other reasons and are being unnoticed. Cyberbullying also tarnishes theimage of a person. It hampers their reputation with the false rumors spread about them. Nearly 8 out of 10 children are being subjected to these various cyberbullying.Cyberbullying detection using machine learning involves using algorithms and statistical models to identify and classify instances of online harassment, abuse, andbullying. This process typically involves training a model on a large dataset of examples of cyberbullying, along with examples of non-cyberbullying text. The model can then be used to make predictions about whether new instances of online text are likely to be cyberbullying. The features that the model considers when making these predictions can include word usage, sentiment, and other linguistic patterns. The goal of this approach is to provide a fast, scalable, and automated way to detect and address cyberbullying, so that online communities can be safer and more supportive for everyone.

Some common machine learning techniques used in cyberbullying detection include:

Supervised learning: This is the most commonly used approach, where the model is trained on a labeled dataset of examples of cyberbullying and non-cyberbullyingtext. The model uses this training data to learn patterns in the language that are associated with cyberbullying, and then uses these patterns to make predictions onnew, unseen text.

Unsupervised learning: This approach involves training the model on a large dataset of text without explicit labels, and then using clustering or other techniques to identify patterns and group similar instances of text together. This can be used to identify clusters of text that are likely to be cyberbullying, or to uncover other patterns in the data that may be useful for detecting cyberbullying.

Deep learning: Deep learning models, such as neural networks, can be used to automatically learn patterns in the text data, and to make predictions about whethera given instance of text is cyberbullying or not. This approach can be especially effective when combined with large amounts of training data and powerful computational resources.

The performance of machine learning models for cyberbullying detection can be evaluated in a number of ways, such as accuracy, precision, recall, and F1 score. The choice of evaluation metric will depend on the specific goals and requirements of the task, and may require careful consideration of trade-offs between false positive and false negative rates. In general, the use of machine learning for cyberbullying detection can be a powerfultool for making online communities safer and more supportive for everyone. However, it is important to keep in mind that machine learning models are not perfect, and that there may be biases and limitations in the data and algorithms that can affect the performance and fairness of these systems.

Additionally, it is important to note that the use of machine learning for cyberbullying detection should be part of a larger, comprehensive approach to addressing online harassment and abuse. Machine learning models can help to identify instances of cyberbullying and bring them to the attention of moderators, but they are not a complete solution in and of themselves.

Effective strategies for addressing cyberbullying also need to consider the social andemotional impact of online harassment on individuals and communities, as well as the broader cultural and legal context in which cyberbullying occurs. They may alsoneed to involve a range of stakeholders, including educators, parents, law enforcement, and technology companies, to create a safe and supportive online environment for everyone.

Machine learning can be a valuable tool for detecting and addressing cyberbullying,but it should be used in a responsible and thoughtful way, as part of a larger and more comprehensive approach to making online communities safer and more supportive.

It is also important to consider ethical and privacy implications when using machine learning for cyberbullying detection. For example, personal data of users may be collected and processed in order to train and evaluate the models, and it is crucial to ensure that this data is handled in a secure and confidential manner, inaccordance with relevant privacy regulations and ethical principles.

Furthermore, there is also a risk that machine learning models may be used to unfairly target and penalize certain individuals or groups, for example, by misidentifying innocent speech as cyberbullying or by disproportionately affecting marginalized or minority communities. To address these concerns, it is important todevelop and implement robust fairness and accountability measures, such as regular audits and evaluations of the models, to ensure that they are operating in afair and ethical manner.

Finally, it is also important to recognize that cyberbullying is a complex and evolving problem that is influenced by a wide range of social, cultural, and technological factors. Machine learning models for cyberbullying detection are just one aspect ofa larger and more comprehensive approach to addressing this issue, and they needto be used in a way that is aligned with broader goals and strategies for creating asafe and supportive online environment. Previous methods for detecting cyberbullying mainly relied on human moderators or rules-based systems. For example, moderators would manually review flagged posts or comments and determine whether they constituted cyberbullying, based on predefined criteria. Rules-based systems, on the other hand, would use a set ofpredefined rules or keywords to automatically identify instances of cyberbullying.

However, these methods had several limitations, including high levels of subjectivity, the need for significant human effort and resources, and the inability to effectively keep up with the large volume and speed of online content.

In recent years, machine learning has emerged as a more effective and scalable approach to detecting cyberbullying. Machine learning models can automatically learn patterns in the text data that are indicative of cyberbullying, and they can process vast amounts of data quickly and efficiently, allowing for near real-time detection of cyberbullying incidents.

Current methods for using machine learning for cyberbullying detection typically involve training supervised machine learning models on large datasets of labeled examples of cyberbullying and non-cyberbullying text, and then using these models to make predictions about new, unseen instances of online text.

These models can be trained using a variety of techniques, such as logisticregression, decision trees, and neural networks.

Additionally, some current methods also incorporate additional information, such as user demographics and network structure, to enhance the performance of the models and to provide a more comprehensive understanding of cyberbullying incidents. For example, some studies have shown that incorporating network information, such as the relationships between users, can improve the accuracy of cyberbullying detection models.

The relevance of the project to the field of machine learning lies in the use of advanced techniques and models to analyze and classify online text data, in order to detect instances of cyberbullying. Machine learning algorithms have the ability to learn from large amounts of data and make predictions based on patterns and relationships in that data, making them well-suited for the task of detecting cyberbullying.

In recent years, there has been a growing interest in the use of machine learning for the detection of online harassment, and cyberbullying in particular. This is due to the increasing prevalence of cyberbullying and the need for more effective methods to address this issue. The use of machine learning can help to overcome some of the challenges associated with traditional methods of cyberbullying detection, such as the lack of scalability and the difficulty of accurately identifying instances of abuse in unstructured text data.

The project is also relevant to the field of online harassment because it seeks to develop and evaluate systems for detecting and preventing cyberbullying. The results of the project have the potential to contribute to a better understanding of the dynamics and patterns of online harassment and abuse, and to provide a framework for developing more effective and scalable methods for addressing theseissues. The use of machine learning has the potential to play a key role in addressingthe problem of cyberbullying, and the project has the potential to make a valuablecontribution to the field.

STATEMENT OF PROBLEM

Cyber bullying, defined as the use of digital technology to harass, humiliate, orthreaten someone, is a growing problem in today's digital age. With the increasing popularity of social media, instant messaging, and other online platforms, cyber bullying has become more prevalent and can have severe consequences for the victims, including low self-esteem, depression, and even suicide. The traditional methods of detecting and addressing cyber bullying, such as manual moderation by platform administrators or reporting by users, are time-consuming and often ineffective. They also rely heavily on human judgment, which can be prone to biases and errors. The problem of cyber bullying detection using machine learning is to develop a system that can accurately and efficiently identify instances of cyber bullying in online content, such as social media posts and comments, and provide relevant information tosupport the prevention and intervention of such incidents. The challenge is to develop a system that can distinguish between instances of cyber bullying and other forms of online expression, such as humor, sarcasm, and disagreement, while also considering the cultural, linguistic, and contextual differences of the content. The system must also be able to handle the large volume and velocityof online content, operate in real-time, and maintain privacy and ethicalstandards.

OBJECTIVES

- Accurately identifying instances of cyberbullying in online communication.
- Classifying the type of cyberbullying being used (e.g., harassment, threats,spreading rumors).
- Determining the severity of cyberbullying to prioritize interventions andresponses.
- Predicting the likelihood that an individual will become a victim ofcyberbullying to intervene and prevent bullying before it occurs.
- Identifying the individuals involved in cyberbullying incidents, in order tohold them accountable and provide support to victims.

SCOPE

The scope of the project includes the following key areas:

- Data Collection and Pre-processing: Gathering a dataset of online content and preparing it for analysis, including cleaning, tokenizing, andnormalizing the text.
- Feature Engineering: Extracting relevant features from the text data, such as sentiment, keywords, and entities.
- Model Training and Evaluation: Selecting and training machine learning algorithms, such as Convolutional Neural Networks, Naive Bayes, and Support Vector Machines, to detect cyber bullying and evaluating their performance using metrics such as accuracy, precision, recall, and F1 score.
- Deployment and Integration: Implementing the solution in a real-world environment and integrating it into existing systems, such as social media platforms, to detect cyber bullying in real-time.
- User Interaction: Designing an interface for users, such as administratorsand moderators, to interact with the system, view outputs, and make decisions based on the results.

SOFTWARE AND HARDWARE REQUIREMENTS

Software Requirements	:	Keras, TensorFlow, Jupyter Notebook, Visualstudio
Technology Implemented	:	JUPYTER NOTEBOOK
Language Used	:	PYTHON 3.9.3
User Interface Design	:	Html, css, javascriptWeb
Browser	:	Google Chrome
Hardware Requirements	:	Intel i3 processor and above 4GB RAM and 256GBHDD

EXISTING SYSTEM

Even though many systems have been developed till now using different machine learning algorithms like Random Forest, Naive Bayes, and Artificial Neural network, the accuracy of those models is low and the work using those classificationtechniques is done with the mindset of detecting Cyber Bullying. In the existing System Support Vector Machine (SVM) is used in which accuracy decreases as the amount of data increases. SVM's are not ideal for dealing with frequent language ambiguities typical for cyberbullying. Only keywords or bag words are used for detection. Personality and Emotions are not considered. Limited data availability, false positives and false negatives, contextual ambiguity, bias in training data, scalability, and computational complexity are some of the limitations in the existingsystem.

Cyber bullying detection using machine learning is a crucial task in the modern digital world, as it helps to identify and prevent bullying behaviour in online communication platforms. The following is an overview of the existing systems thatuse machine learning to detect cyber bullying.

1. **Supervised Machine Learning**: In this type of machine learning, a model istrained on a labelled dataset to identify the type of speech that constitutes bullying. The model then predicts the type of speech in new, unseen data. This approach is often used in sentiment analysis and text classification tasks.Some of the commonly used algorithms for this approach include Naive Bayes, Support Vector Machines (SVM), and Logistic Regression.
2. **Unsupervised Machine Learning**: This approach does not use labelled datato train the model, but instead, the model is trained on the patterns and relationships within the data. Some of the commonly used algorithms for thisapproach include clustering algorithms such as K-Means and hierarchical clustering, and anomaly detection algorithms such as One-Class SVM.
3. **Deep Learning**: Deep learning algorithms, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), are used to detect cyber bullying based on the semantic meaning of the text. These algorithms are designed to handle large amounts of data and complex relationships between the data. They can learn features from the text that are important for detecting cyber bullying.
4. **Transfer Learning**: Transfer learning is a deep learning technique in which a model trained on a large dataset is fine-tuned for a specific task. In the context of cyber bullying detection, a pre-trained language model, such as BERT, can be fine-tuned for the task of cyber bullying detection. This approachcan improve the performance of the model, as it leverages the knowledge learned from the large pre-trained dataset.
5. **Hybrid Approaches**: Some existing systems use a combination of supervised and unsupervised machine learning, or deep learning and transfer learning, to detect cyber bullying. This approach can improve the performance of the model, as it leverages the strengths of multiple machine learning techniques.

In cyberbullying detection, there are several existing systems for cyber bullying detection using machine learning. Each approach has its strengths and weaknesses, and the choice of which approach to use depends on the specific requirements of the task. However, regardless of the approach used, it is important to note that the performance of these systems can be improved through the use of appropriate feature engineering, data pre-processing, and model fine-tuning techniques.

Techniques like unsupervised labelling methods which use N-gram, TF-IDF methods to detect cyberbullying are used which use the youtube dataset to detect attacks. Another key aspect of existing cyber bullying detection systems is the dataset used for training and evaluating the models. The quality and diversity of the dataset play a crucial role in determining the performance of the models. Some existing systems use annotated datasets, which contain examples of bullying and non-bullying behaviour, to train and evaluate the models. Other systems use unannotated datasets, such as social media posts, and apply unsupervised machine learning techniques to detect bullying behaviour.

One of the challenges faced by existing systems is the changing nature of cyber bullying behaviour. Cyber bullies often use creative and evolving methods to harass their victims, making it difficult for machine learning models to keep up. Additionally, the context in which the bullying behaviour occurs is important in determining whether it is bullying or not. For example, a joke between friends may be misinterpreted as bullying by a machine learning model. To address these challenges, it is important to regularly update the dataset used to train and evaluate the models, as well as to fine-tune the models based on the changing nature of cyber bullying behaviour.

Another challenge faced by existing cyber bullying detection systems is the need for privacy and security. Many cyber bullying detection systems use social media posts as their source of data, which raises privacy and security concerns. To address these concerns, it is important to use appropriate data anonymization techniques and secure data storage methods.

Finally, it is important to note that cyber bullying detection using machine learning is only one aspect of preventing cyber bullying. The most effective approach is to take a multi-disciplinary approach that combines technological solutions with education, awareness, and policy initiatives.

In conclusion, existing systems for cyber bullying detection using machine learning have made significant advances in identifying and preventing cyber bullying. However, there are still challenges that need to be addressed, such as the changing nature of cyber bullying behaviour, privacy and security concerns, and the need for a multi-disciplinary approach. Nevertheless, the use of machine learning to detect cyber bullying is a promising and effective approach, and further research and development in this area is expected to lead to even better results.

It is also important to note the ethical considerations when using machine learning for cyber bullying detection. One of the main concerns is the potential for false positives, where a machine learning model may incorrectly label a statement as bullying. This can have serious consequences for the person wrongly accused of bullying, including social and professional consequences. To minimize the risk of false positives, it is important to use robust evaluation methods, such as cross- validation and test-time augmentation, and to have human annotators validate theresults of the models.

Another ethical concern is the potential for bias in the models. Machine learning models are only as good as the data they are trained on, and if the training data contains biases, these biases will be reflected in the models. This can result in unfairtreatment of certain groups, such as minority groups or those with different politicalviews. To address this, it is important to use diverse and representative datasets and to monitor the models for potential biases.

Finally, the use of machine learning for cyber bullying detection raises privacy concerns. Social media platforms, in particular, collect a vast amount of personal information, which can be used to detect bullying behaviour. However, this raises concerns about data privacy and data protection. To address these concerns, it is important to use secure data storage and transfer methods, and to use data anonymization techniques where appropriate.

In conclusion, the use of machine learning for cyber bullying detection is a complexissue that requires a thorough understanding of the technical, ethical, and legal considerations. While machine learning has the potential to greatly improve the ability to detect and prevent cyber bullying, it is important to approach this issue with caution and to carefully consider the potential consequences of using these technologies.

SOFTWARE REQUIREMENTS SPECIFICATION

Introduction:

1. Purpose: Explain the purpose and scope of the project.
2. Objectives: Describe the main objectives and goals of the project.
3. Overview: Give a brief overview of the system and its functionalities.Functional

Requirements:

1. Input: Specify the types of inputs that the system should accept.
2. Processing: Define the machine learning algorithms to be used for detectionof cyber bullying.
3. Output: Outline the types of outputs that the system should provide.Non-Functional

Requirements:

1. Performance: Specify the performance requirements such as response timeand accuracy.
2. Security: Discuss the security measures to be implemented to protectsensitive data.
3. Usability: Describe the user-friendliness of the system and ease of use.Technical

Requirements:

1. Hardware: List the hardware requirements for the system, including serverspecifications and storage requirements.
2. Software: Specify the software requirements, including programminglanguages and libraries.
3. Infrastructure: Describe the network infrastructure and connectivity requirements.

User Requirements:

1. User roles: Define the different user roles and their access levels.
2. User interface: Describe the user interface, including the layout and design.
3. User interactions: Specify the ways in which users will interact with thesystem.

Acceptance Criteria:

1. Define the criteria that must be met for the system to be consideredcomplete and ready for release.
2. Include functional, performance, and security requirements.

Deployment:

1. Explain the steps to deploy the system in a production environment.
2. Include any necessary training or support for users.
3. Maintenance and Support:
4. Discuss the maintenance and support requirements, including bug fixingand updating.
5. Include a plan for future development and enhancements

PROPOSED SYSTEM

Working Methodology:

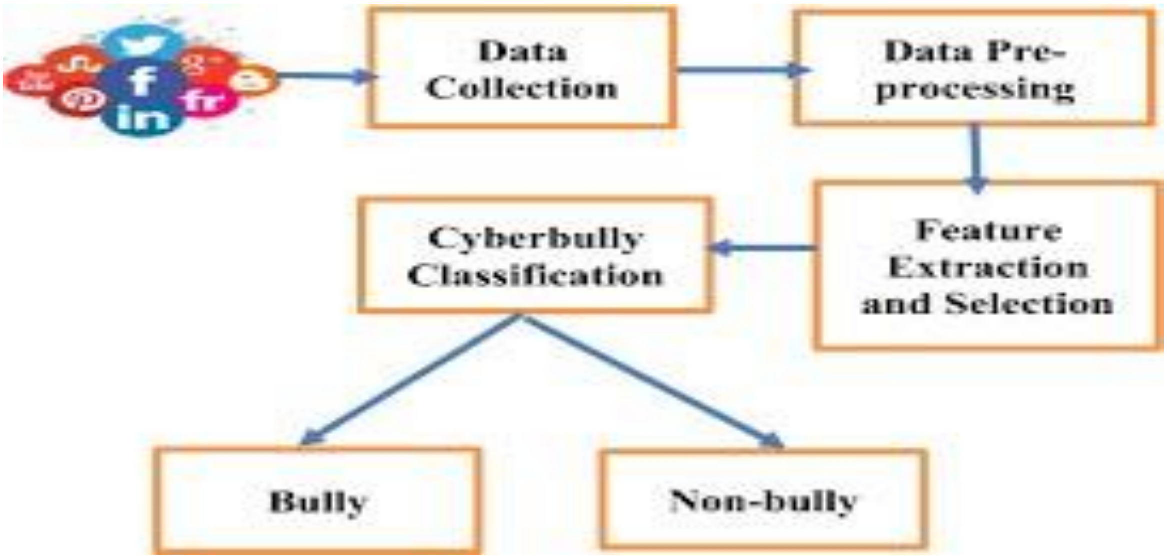
Building a machine learning model to identify cyber bullyingrequires several steps:

Data Collection: Collect a large and diverse dataset of text (such as social media posts or online comments) that contain instances of cyber bullying and non- cyber bullying examples.

Data Preprocessing: Clean the data by removing any irrelevant information, such as URLs, and perform any necessary text normalization, such as convertingall text to lowercase.

Feature Engineering: Decide on the features that the model will use to make itspredictions. In the case of text classification, this usually involves creating numerical representations of the text, such as using term frequency-inverse document frequency (TF-IDF) or word embeddings.

Model Selection: Choose an appropriate machine learning algorithm for the task of text classification,



such as a support vector machine (SVM), a neural network,or a Logistic Regression.

Model Training: Train the model on the preprocessed and feature-engineered data. This involves providing the model with the features and corresponding labels (e.g., cyber bullying or non-cyber bullying) for each instance in the training data.

Model Evaluation: Evaluate the performance of the model by testing it on a held-out test set and computing metrics such as accuracy, precision, recall, and F1 score.

Model Fine-tuning: If necessary, fine-tune the model by adjusting its parametersor trying a different algorithm, and then repeat the evaluation step.

ALGORITHM USED

The Algorithm that will be implemented for working of this proposed system isas follows:

Step 1: Import all the modules required such as numpy, pandas, matplotlib, sklearn, Tensorflow.

Step 2: Load the cyberbullying dataset and extract the attributes.

Step 3: Perform the necessary preprocessing techniques and analyze the data.

Step 4: Split the dataset into training and testing data.

Step 5: Perform training the machine learning model using Ensemble learning,Naïve Bayes, SVM, K Nearest Neighbors, Logistic Regression.

Step 6: Compare the accuracies of the algorithms and use the best algorithm forprediction.

Step 7: Predict the new data using the algorithm which obtained better accuracy.

PROJECT DESIGN

Use case diagram:

A use case diagram is a type of Unified Modeling Language (UML) diagram that represents the functionality offered by a system to its users. It is used to mode the relationships between the system and its actors (users or external systems) and to describe the interactions between the actors and the system. A use case diagram provides a high-level view of the functional requirements of a system and helps to communicate the system's capabilities to stakeholders.

A use case diagram consists of:

Actors: Represented as stick figures, actors are the entities that interact with the system, such as a human user or another system.

Use cases: Represented as oval shapes, use cases describe the functional requirements of the system and what the system can do for its actors.

Associations: Represented as lines connecting actors and use cases, associations indicate the relationships between the actors and the use cases they participate in.

System boundary: Represented as a rectangle, the system boundary encloses the use cases and actors and defines the scope of the system.

The use case diagram helps to identify the functional requirements of a system and to ensure that the system meets the needs of its users. It is used during the requirements gathering and analysis phase of the software development process to provide a visual representation of the system's capabilities and to facilitate communication between stakeholders.

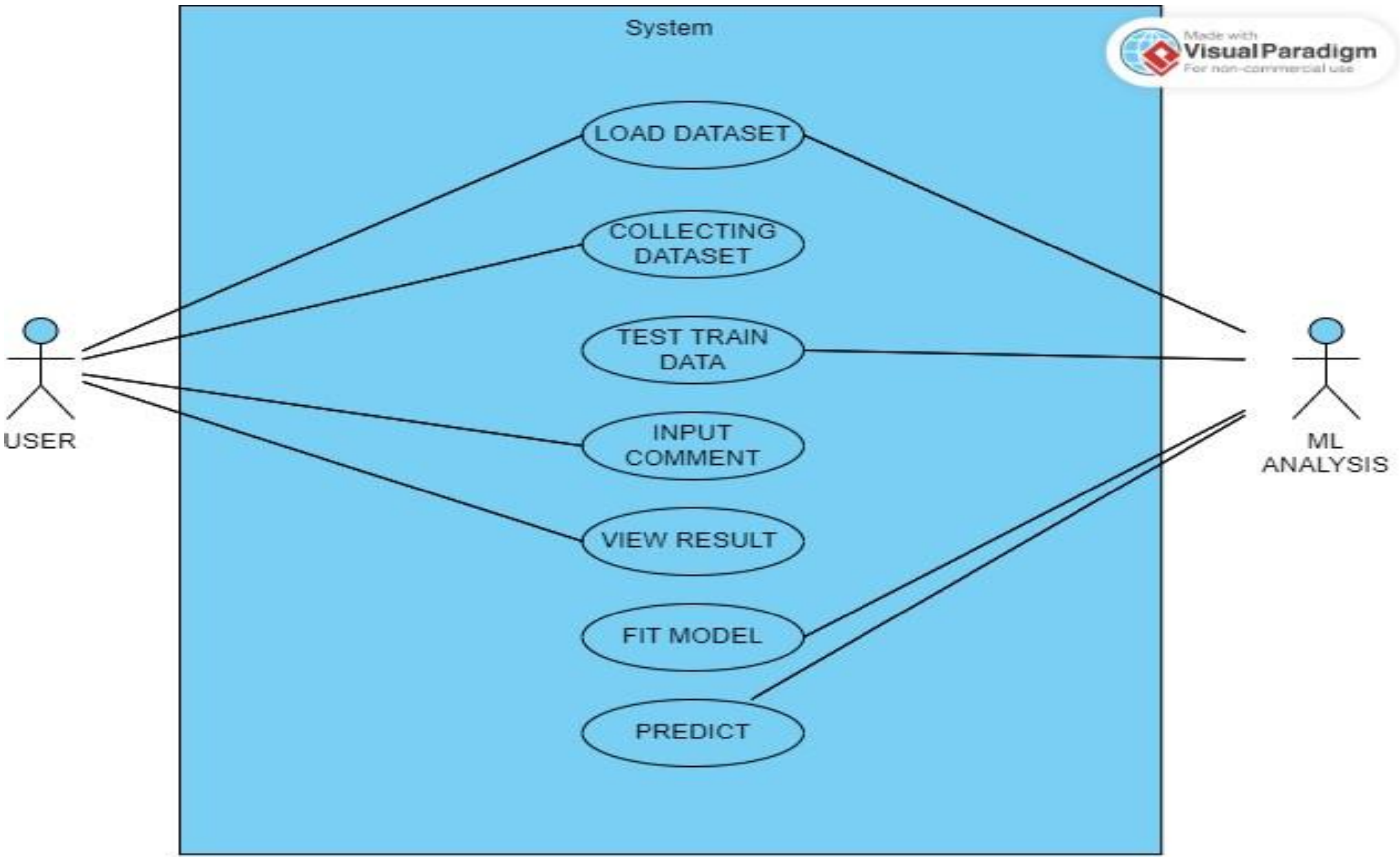


Fig. Use Case Diagram

Sequence diagram:

A sequence diagram is a type of Unified Modeling Language (UML) diagram that represents the interactions between objects or components in a system over time. It provides a visual representation of the interactions between objects and the orderin which they occur, helping to describe the behavior of a system and to communicate it to stakeholders.

A sequence diagram consists of:

Objects: Represented as rectangles, objects represent instances of classes and represent the entities that interact with each other in the system.

Lifelines: Represented as vertical lines, lifelines represent the existence of objects over time and the interactions they participate in.

Messages: Represented as arrows, messages represent the interactions between objects and describe the behavior of the system.

Activations: Represented as rectangles on lifelines, activations represent the processing time of an object during an interaction.

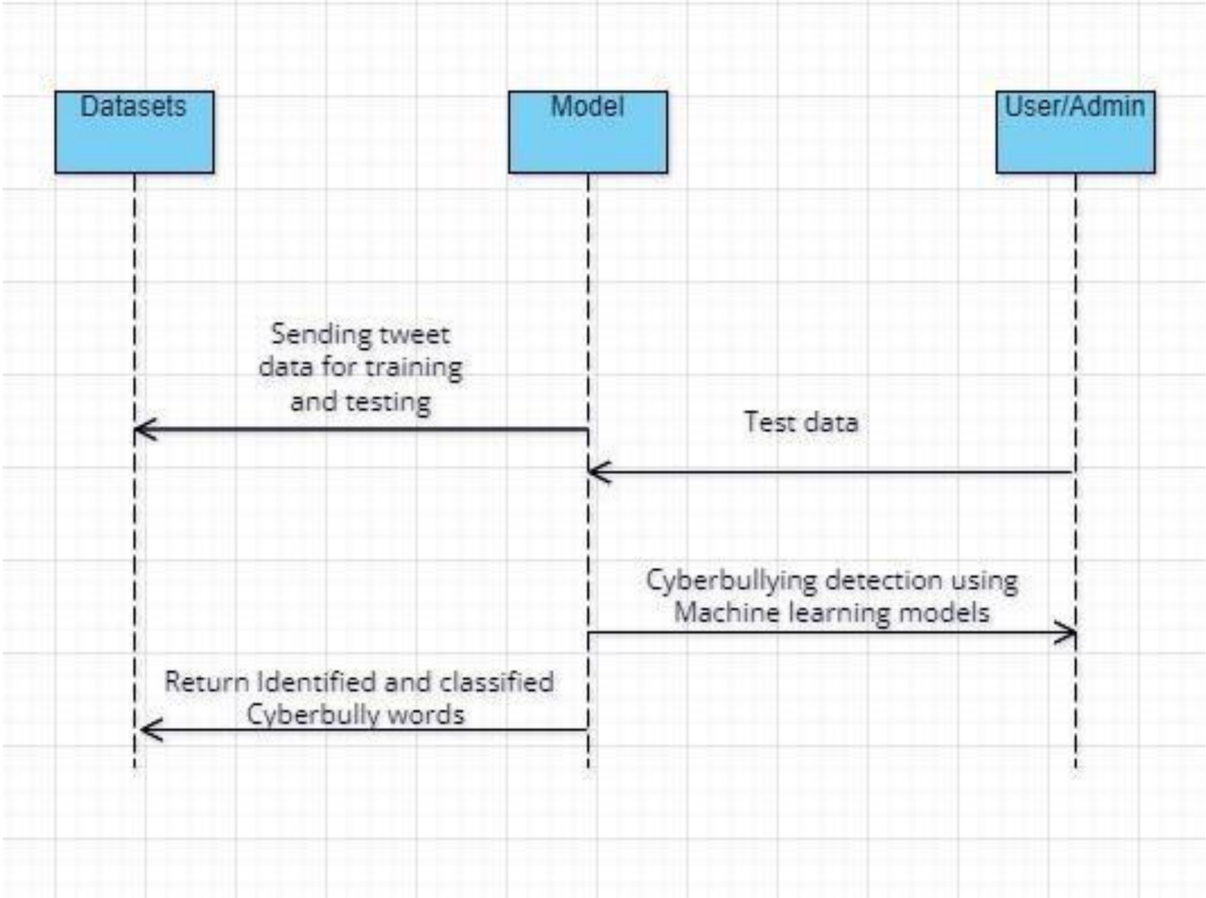


Fig. Sequence Diagram

State Chart Diagram:

A statechart diagram, also known as a state machine diagram or state diagram, is a type of Unified Modeling Language (UML) diagram that represents the behavior of a system as a series of states and the events that cause the systemto transition from one state to another. It provides a visual representation of the behavior of a system and helps to communicate the behavior to stakeholders.

A statechart diagram consists of:

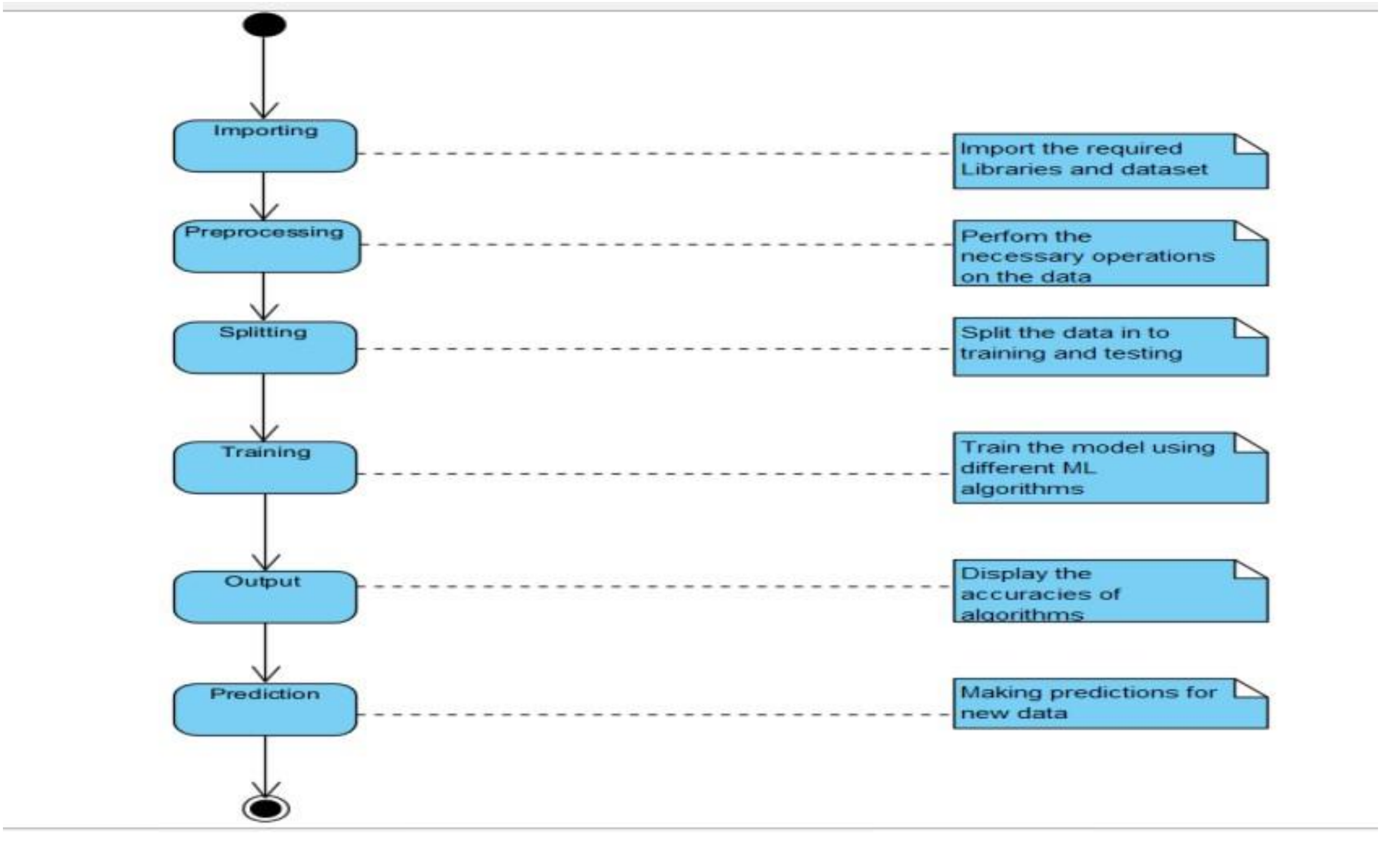
States: Represented as rectangles, states represent the different conditions orphases that a system can be in.

Transitions: Represented as arrows, transitions represent the events that causea system to change from one state to another.

Events: Represented as ovals, events represent the external or internal triggers that cause transitions to occur.

Actions: Represented as short horizontal lines with text, actions describe theactivities that take place during a transition.

A statechart diagram helps to describe the behavior of a system and tounderstand how a system reacts to events. It is used to model the dynamic behavior of a system and to communicate the behavior to stakeholders. The diagram is particularly useful in describing complex behavior and can be used to model the behavior of real-time systems, embedded systems, and user interfaces.



Data Flow Diagram:

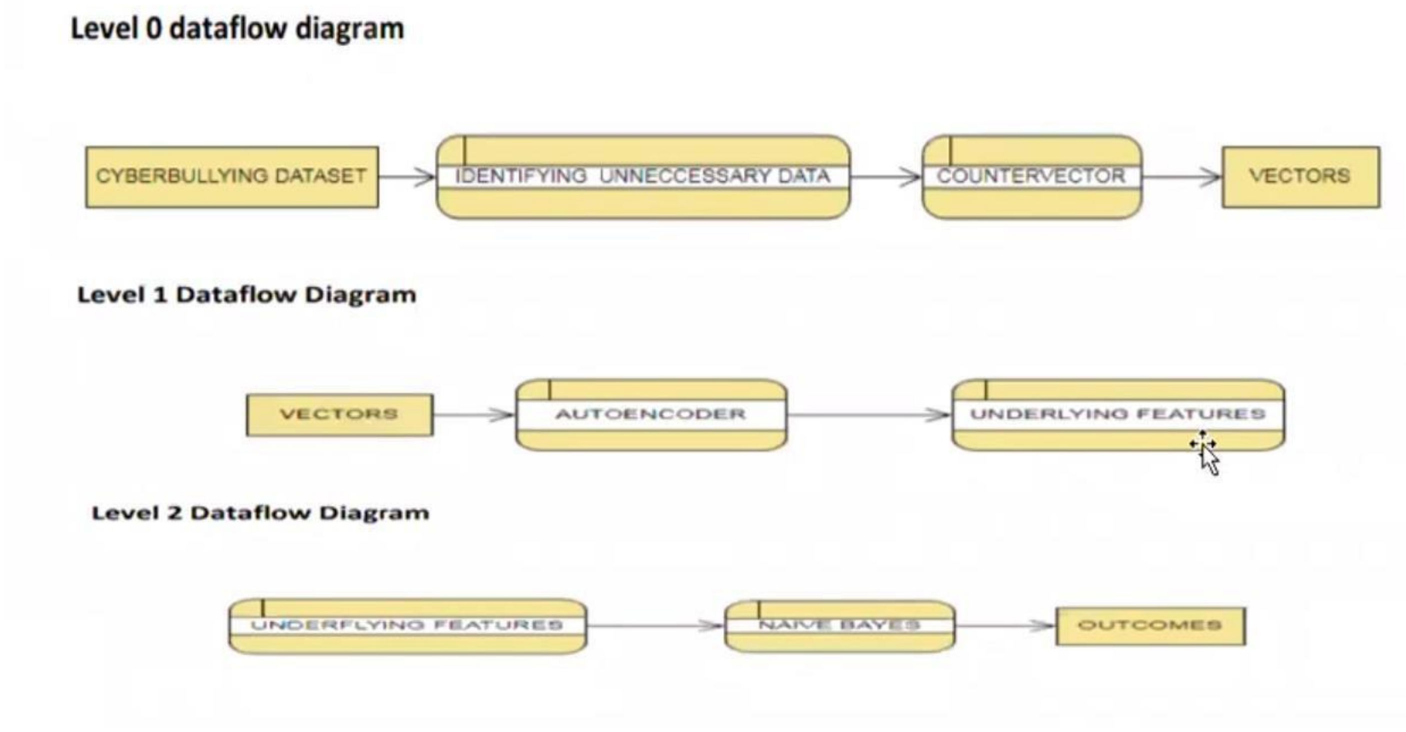
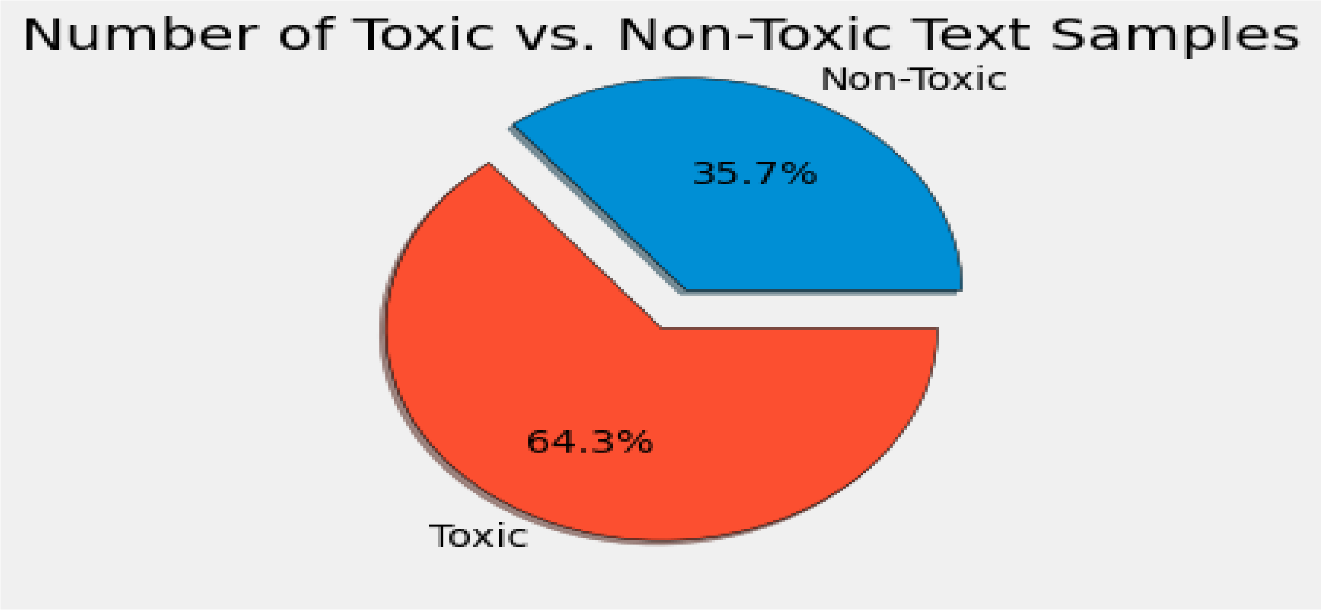


Fig. Data Flow Diagram

PROJECT CODE

DATASET DESCRIPTION:

The dataset is the snapshot of the twitter information during a particular period of time .
It consists of tweets from all kinds of users .It has seven columns na--mely 'Unnamed,'count', 'hate_speech' , 'offensive_language ', 'neither', 'class', 'tweet '.
The class is our dependent variable and remaining
are the independent varibles .The class consists of numeric values so we would be converting them into discrete values by labelling them.The dataset have nearly 24000 rows .we would be partitioning it for testing and training .
It does not have any missing values or inconsistent
values .
The tweets is the most important variable as we are going to use it for classifying the comments .The class is our target variable one which will be predicted.



IMPORTING LIBRARIES:

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score, f1_score, confusion_matrix, classification_report
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import RandomizedSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.svm import SVC,LinearSVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.pipeline import Pipeline
import pickle
```

IMPORTING AND DOWNLOADING NLP TOOLS:

```
import re
import nltk
nltk.download('stopwords')
from nltk.util import pr
stemmer = nltk.SnowballStemmer("english")
from nltk.corpus import stopwords
import string
stopword=set(stopwords.words("english"))
```

LOADING DATASET:

```
ds=pd.read_csv("twitter_data.csv")
print(ds.head())
ds.describe()
ds.columns
len(ds.index)
```

```
ds['labels']=ds['class'].map({0:"Hate Speech Detected", 1:"cyber Bullying detected", 2:"no hate and offensive speech"})
print(ds.head())
```

Here, we would be converting the target variable into a discrete variable as we would be working with decision tree classifier.

```
ds=ds[['tweet','labels']]
ds.head()
```

Here, we would be visualizing the changes we have made.

USER DEFINED FUNCTIONS:

```
def clean(text):
    text=str(text).lower()
    text=re.sub('\[.*?\]','',text)
    text=re.sub('https?://\S+|www\.\S+','',text)
    text=re.sub('<.*?>+','',text)
    text=re.sub('[%s]'% re.escape(string.punctuation),'',text)
    text=re.sub('\n','',text)
    text=re.sub('\w*\d\w*','',text)
    text=[word for word in text.split(' ')if word not in stopword]
    text=" ".join(text)
    text=[stemmer.stem(word) for word in text.split(' ')]
    text=" ".join(text)
    return text
ds["tweet"]=ds["tweet"].apply(clean)
print(ds.head())
```

The inbuilt function clean is used to remove the unwanted gibberish or symbols from the tweets that are in the dataset.

Function to remove emoji's in text:

```
In [ ]: def strip_emoji(text):
            return emoji.replace_emoji(text,replace="")
```

SPLITTING THE DATASET:

```
x=np.array(ds["tweet"])
y=np.array(ds["labels"])
cv=CountVectorizer()
x=cv.fit_transform(x)
x_train, x_test, y_train, y_test=train_test_split(x,y, train_size=0.73, random_state=0)
```

CREATION AND TRAINING OF DIFFERENT MODELS:

Decision Tree Classifier:

```
In [37]: dtc = DecisionTreeClassifier()
         dtc.fit(X_over, y_over)
         y_pred = dtc.predict(X_test)
         print("Accuracy: ",metrics.accuracy_score(y_test, y_pred))
         print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
         getStatsFromModel(dtc)
```

```
Accuracy:  0.8455386153461635
Confusion Matrix: 
 [[1883  546]
 [  72 1500]]
              precision    recall  f1-score   support

           0       0.96      0.78      0.86      2429
           1       0.73      0.95      0.83      1572

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

Support Vector Machine:

In [69]:
```
lin_svc = LinearSVC()
```

In [70]:
```
lin_svc_cv_score = cross_val_score(lin_svc,X_train_tf,y_train,cv=5,scoring='f1_macro',n_jobs=-1)
mean_lin_svc_cv = np.mean(lin_svc_cv_score)
mean_lin_svc_cv
```

Out[70]: 0.8220066371295554

Naïve Bayes:

In [32]:
```
gnb = GaussianNB()
gnbmodel = gnb.fit(X_over, y_over)
y_pred = gnbmodel.predict(X_test)
print ("Score:", gnbmodel.score(X_test, y_test))
print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
getStatsFromModel(gnb)
```

```
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
```

Logistic Regression:

```
In [34]:  lgr = LogisticRegression()
          lgr.fit(X_over, y_over)
          y_pred = lgr.predict(X_test)
          print("Accuracy: ",metrics.accuracy_score(y_test, y_pred))
          print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
          getStatsFromModel(lgr)
```

Logistic Regression Confusion matrix:

```
Accuracy:  0.8007998000499875
Confusion Matrix:
 [[1907  522]
 [ 275 1297]]
              precision    recall  f1-score   support

           0       0.87      0.79      0.83      2429
           1       0.71      0.83      0.76      1572

    accuracy                           0.80      4001
   macro avg       0.79      0.81      0.80      4001
weighted avg       0.81      0.80      0.80      4001
```

TESTING THE MODEL:

```
test_data="your work sucks"
ds=cv.transform([test_data]).toarray()
print(clf.predict(ds))
```

RESULT

```
from sklearn import metrics
y_pred = clf.predict(x_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.8734309623430963
```

```
clf=DecisionTreeClassifier()
clf.fit(x_train,y_train)
```

```
DecisionTreeClassifier()
```

```
test_data="your work sucks"
ds=cv.transform([test_data]).toarray()
print(clf.predict(ds))
```

```
['cyber Bullying detected']
```

```
from sklearn import metrics
y_pred = clf.predict(x_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.8734309623430963
```

CONCLUSION AND DISCUSSION:

CHALLENGES FACED:

1) Long Execution time: Running all of the 450 experiments in the first set was challenging since it required a huge amount of time to complete. To overcome the slow execution time of the models, the experiments were conducted in parallel on high-performance compute nodes in Compute Canada clusters.

2) Random under sampling: The use of random under sampling had a negative effect on model's performance. This can be due to the huge loss of information that happens when using this under sampling method. It is possible to experiment with over sampling techniques in the future to overcome this issue.

FUTURE WORK:

Along with the laws that are used to punish those people who cause cyber violence having an system that automatically detects the context of cyberbullying and changes it into a positive comment will be of great help as the saying goes prevention is the best .Using this will prevent lot of people from depression ,low self-esteem and also suicides.it also never let the users to use the social media as a tool to humiliate or bully others .The cyberbullying detecting system will lead to healthy environment on social media. It can be embedded into all social media and messaging apps.

REFERENCES

BASE PAPER:

- Improving Cyberbullying Detection using Twitter Users' Psychological Featuresand Machine Learning
- Date Added to ELSEVIER: 31 December 2019
- Publisher: ELSEVIER

Authors: Vimala Balakrishnan Ph.D , Shahzaib Khan , Hamid R. Arabnia Ph.D Cyberbullying is the act of using technology, such as the internet and smartphones, to harass, humiliate, threaten, or intimidate someone. It can take many forms, including mean and hurtful messages, spreading rumors or lies, sharing personal information or photos without consent, and creating fake socialmedia profiles. The impact of cyberbullying can be severe, leading to emotional distress and even mental health problems.

LIST OF REFERENCE PAPERS

- ALBayari, Reem, Sharif Abdullah, and Said A. Salloum. "Cyberbullying classification methods for Arabic: A systematic review." *Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2021)*. Cham: Springer International Publishing, 2021.

- Gencoglu, Oguzhan. "Cyberbullying detection with fairness constraints." *IEEE Internet Computing* 25.1 (2020): 20-29.

- Ali, Wan Noor Hamiza Wan, Masnizah Mohd, and Fariza Fauzi. "Cyberbullying detection: an overview." *2018 Cyber Resilience Conference (CRC)*. IEEE, 2018.

- Zhao, Rui, Anna Zhou, and Kezhi Mao. "Automatic detection of cyberbullying on social networks based on bullying features." *Proceedings of the 17th international conference on distributed computing and networking*. 2016

- Dadvar, Maral, et al. "Improving cyberbullying detection with user context." *Advances in Information Retrieval: 35th European Conference on IR Research, ECIR 2013, Moscow, Russia, March 24-27, 2013. Proceedings 35*. Springer Berlin Heidelberg, 2013.