Automatic Detection of Cyber Bullying Using Data Science & Machine Learning

A PROJECT REPORT

Submitted by

ANIKET DESAI

2101202003

Under the Guidance of

Dr. Nandini Chaudhari

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Information Technology

Krishna School of Emerging Technology & Applied Research

Drs. Kiran & Pallavi Patel Global University, Vadodara

May, 2025

Automatic Detection of Cyber Bullying Using Data Science & Machine Learning

Major Project

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology in Information Technology

By

[Aniket Desai]: 2101202003

Training At

LearnAtRISE -RISHABH SOFTWARE

Atladara, VADODARA-390012 Gujarat.

Internal Guide Name :
Dr.Nandani Chaudhari

External Guide Name :
Mr. Ketan Rathod

Krishna School of Emerging Technology & Applied Research

DECLARATION

We hereby declare that the Project report submitted along with the Project entitled **Automatic Detection of Cyber Bullying Using Data Science & Machine Learning** submitted in partial fulfillment for the degree of Bachelor of Technology in **Information Technology** to Drs. Kiran & Pallavi Patel Global University, Vadodara, is a bonafide record of original project work carried out by me at Drs. Kiran & Pallavi Patel Global University under the supervision of **Dr. Nandani chaudhari**, and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

Name of Student	Sign of Student
1. Aniket Desai	____________

Krishna School of Emerging Technology & Applied Research

CERTIFICATE

This is to certify that the project report submitted, along with the project entitled **"AUTOMATIC DETECTION OF CYBER BULLYING USING DATA SCIENCE & MACHINE LEARNING"** has been carried out by **ANIKET DESAI(2101202003)** under my guidance, in partial fulfillment of the requirements for the degree of **Bachelor of Technology in Information Technology**, 8th Semester, at **Drs. Kiran & Pallavi Patel Global University** during the academic year **2024–25**.

Ms. Monali Parikh	Dr. Nandini Chaudhary
Internal Guide	Head of the Department

RSPL/RISE/24-25/024 Date: 11^{th} April, 2025

To,

Aniket Desai

KPGU

Vadodara, Gujarat

Internship Duration Certificate

Dear Aniket,

We are pleased to certify that you have been selected for an Internship in Data Analytics at Rishabh Integrated Skill Enhancement Centre (RISE), a skill development initiative by Rishabh Software Pvt. Ltd from 23rd January to 22^{nd} April, 2025 and the same will be continued till June, 2025.

This opportunity is based on your expressed interest in enhancing your skills and gaining practical experience. Please note that this internship is focused on learning and skill development, and therefore does not include payment or a stipend.

During the internship, you have undergone of comprehensive training with a strong focus on acquiring new skills, deepening your understanding of core concepts, and applying your knowledge through hands-on experience.

We look forward to supporting your learning journey.

Best regards,

For LearnAtRISE

Aparna Shah Singhal
Aparna Shah Singhal (Apr 15, 2025 13:38 GMT+5.5)

Ms. Aparna Singhal

Team LearnAtRISE

ACKNOWLEDGEMENT

We would like to express sincere appreciation and gratitude to **Dr.Nandani Chaudhari** esteemed teacher, for her invaluable guidance, unwavering support, and mentorship throughout our college project. Her expertise, dedication, and commitment to our academic growth have played a pivotal role in the successful completion of our project. We are immensely grateful for her constant encouragement, insightful feedback, and constructive criticism, which have significantly enriched our learning experience.

Furthermore, we extend our heartfelt thanks to KPGU (Kiran and Pallavi Global University) for providing us with the platform and resources to undertake and accomplish our project goals. The college's support and commitment to our educational endeavours have been instrumental in our growth and success. We deeply appreciate the opportunities, infrastructure, and academic environment that KPGU has provided, enabling us to excel in our project and expand our knowledge in the field.

We would also like to acknowledge the administration and faculty of KPGU for their continuous support, guidance, and encouragement. Their expertise and willingness to go above and beyond have greatly contributed to the development and realization of our project. We are grateful for their dedication to fostering an environment of excellence and for their commitment to nurturing our academic journey.

Lastly, we extend our appreciation to all individuals who have assisted us in various capacities, whether through their expertise, time, or resources. Your contributions have been invaluable, and we are deeply thankful for your support.

ABSTRACT

In terms of modern culture, social media has emerged as the most potent forum in recent memory. A growth in online harassment, cyberbullying, and cybercrime has occurred with the widespread use of social media, which has had a beneficial effect on people's outlook on the world. People's mental health be negatively impacted by cyberbullying, and in some situations, it even be a direct cause of mental health issues.

Sexually explicit comments and rumours that are disseminated by several users are two examples of the kinds of things that are badly impacting the social media ecosystem. There has been a rise in recent years in the number of researchers interested in identifying indicators of online harassment. One of our goals is to employ Natural Language Processing (NLP) and Random Forest regression to create a system that identify instances of online abuse

The rapid spread of the COVID-19 disease has altered cultural norms, leading to a rise in cyberbullying, especially among young people. Younger folks are the most likely to follow this tendency. The number of reported incidents of cyberbullying has risen in tandem with the dramatic rise in popularity of several platforms that promote online engagement.

These word characteristics might be made up of terms that stand in for the context of the various words in our lexicon; this is aid in creating vectors that link words with comparable meanings.

List of Figures

List of Tables

Abbreviations

ALU	Arithmetical & Logical Unit
SDLC	Software Development Life Cycle
NLP	Natural Language Processing
IT	Information Technology
LSTM	Long-Short-Term Memory
AI	Artificial Intelligence
SVM	Support Vector Machines
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks
UML	Unified Modeling Language
GPU	Graphics Processing Units
CPU	Central Processing Units
SSD	Solid-State Drive
AWS	Amazon Web Services
TF-IDF	Term Frequency-Inverse Document Frequency
UX	User Experience
QA	Quality Assurance
KPI	Key Performance Indicators

Table of Contents

1. Introduction to Project

1.1 Project / Internship Summary

It should not come as much of a surprise that young people are utilizing various forms of social media to harass and hurt one another because of the growth of the internet and other forms of social media. Nevertheless, there is a dearth of study that has been conducted in Arab communities to investigate the myriad manifestations and dimensions of cyberbullying. Because of the constraints imposed by society and culture, cyberbullying is also a problem in many other civilizations.

When it comes to the society of today, social media has evolved into the most influential platform in the annals of recent history. The proliferation of social media has not only had a positive influence on society's vision of the world, but it has also contributed to an increase in instances of online harassment, cyberbullying, and cybercrime. Cyberbullying has a detrimental impact not only on the physical health of individuals, but also on their mental health; in some situations, it even be the direct cause of mental health issues in those who are victimized. The environment of social media is being negatively impacted by a number of factors, including sexually explicit comments and rumors that are spread by multiple users. Recent years have seen an uptick in the number of scholars interested in determining ways to spot instances of online harassment. In order to develop a system that is capable of locating instances of online harassment, one of our goals is to make use of Natural Language Processing (NLP) as well as Random Forest regression.

Due to the quick spread of the COVID-19 disease, which has led to a shift in cultural standards, people, especially young people, are facing an upsurge in cyberbullying. This trend is especially prevalent among younger people. Because of the meteoric rise in popularity of numerous programmed that facilitate online contact, the number of reported cases of cyberbullying has increased. Not only has the COVID-19 epidemic caused a shift in social contacts to take place on the internet, but it has also helped contribute to the ongoing digitization of cyberbullying. When people started working from home, it wasn't long before bullying followed.

1.2 Purpose

The purpose of automatic detection of cyberbullying using data science and machine learning is to develop effective methods to identify and prevent online harassment and abusive behavior. Cyberbullying is a serious problem that affects many people, particularly children and teenagers. It led to emotional distress, mental health issues, and even suicide.

By using data science and machine learning techniques, we build models that automatically identify cyberbullying behavior in online communications. These models analyze various forms of online communication, such as text messages, social media posts, and emails, to detect patterns and characteristics of abusive behavior.

The benefits of automatic detection of cyberbullying using data science and machine learning are manifold. Firstly, it is help in early identification and intervention, preventing further harm to the victim. Secondly, it is reducing the burden on human moderators who are responsible for monitoring online communication for cyberbullying. Thirdly, it is help in creating a safer and more supportive online environment for everyone.

Overall, the goal of automatic detection of cyberbullying using data science and machine learning is to create a more positive and inclusive online space where people communicate freely without fear of harassment or abuse.

1.3 Objective

The following list provides the primary goals that is accomplished in order to construct the suggested model.

To develop a word tokenization for sentence or paragraph that converts the text to separate words in a list data structure so that manipulated.

To build and comapare the results the word Embedding Techniques of TF-IDF and Word2Vec.

To construct a model for detecting cyberbullying or inappropriate language using naive bayes, a family of classification algorithms founded on Bayes' theorem, and to evaluate its efficacy against other approaches. It's not just one algorithm, but a group of related algorithms that all work on the same premise: that separating each pair of features being classed is a separate and distinct task.

To use bagging technique to improve model performance as it diversifies models as each individual tree which allows randomly sample from the dataset with replacement.

To perform evaluation of model using comparative study between various Supervised algorithms.

1.4 Scope (what it do and ’t do)

What it do:

Analyze large volumes of data: Machine learning algorithms process huge amounts of data quickly and accurately, allowing for the analysis of vast amounts of online content.

Identify patterns and trends: By analyzing patterns and trends in online conversations, machine learning algorithms identify instances of cyberbullying.

Identify specific types of cyberbullying: Different types of cyberbullying identified and categorized by analyzing the content of online interactions.

Flag potentially problematic content: Machine learning algorithms flag content that considered cyberbullying, so that human moderators review it and take appropriate action.

What it 't do:

Understand context: Machine learning algorithms not always understand the nuances of online conversations and miss the context of a particular message.

Detect sarcasm and irony: Machine learning algorithms are not always capable of detecting sarcasm and irony, which make it difficult to identify cyberbullying in some cases.

Identify all forms of cyberbullying: Some forms of cyberbullying, such as subtle forms of harassment or manipulation, difficult for machine learning algorithms to identify.

Take action on its own: While machine learning algorithms flag potentially problematic content, they not take action on their own to address cyberbullying. Human moderators are still needed to review flagged content and take appropriate action.

1.5 Technology and Literature Review

Technology Review:

Automatic detection of cyberbullying involves using machine learning algorithms to analyze patterns of text or other data to identify potentially harmful or abusive content. Some common approaches to detecting cyberbullying using machine learning include:

Text classification: This involves training a machine learning model to classify text messages as either cyberbullying or not. The model trained using labeled data, such as text messages that have already been identified as cyberbullying, or it use unsupervised learning techniques to identify patterns in the data.

Sentiment analysis: This involves analyzing the sentiment of text messages to identify those that are negative or abusive. Machine learning models trained to recognize patterns in language that indicate hostility, aggression, or other negative emotions.

Natural Language Processing (NLP): This is the use of machine learning models to process and analyze human language, including identifying cyberbullying content. NLP techniques used to extract relevant features from text, such as the use of profanity or specific topics related to bullying.

Social network analysis: This involves analyzing social media networks to identify patterns of behavior that indicate cyberbullying. Machine learning models trained to identify clusters of users who engage in abusive behavior or to detect changes in the volume or tone of conversations on social media.

Overall, automatic detection of cyberbullying using data science and machine learning is a complex and ongoing research area, with many different approaches being developed and tested.

Literature Review:

In recent times, the number of people using social media has been increasing at an exponential rate, making it the most prominent online forum of the 21st century. Growth of the social media also creates a negative impact on the thought of the society as well as various problems such as the online abuse, cyberbullying as well as cybercrime increases drastically (Campbell, 2012). Cyberbullying also impacts on the mental state of the people as well as creates a mental stress (Campbell, 2012). Many incidents such as the rumours as well as the sexual remark creates a negative impact on the social media. Identification of the online harassment gained attention of the various researchers in recent time. The aim to develop the model is to design an effective system for the detection of the online abuse with the help of the Natural Language processing as well as the Random Forest regression technique.

Others are free to publish anything they choose on social media platforms, including photographs, videos, and documents, and they interact with people from all over the globe in real time. People connect to social media using either their laptops or their mobile phones. Important social media networks including Facebook, Twitter, Instagram, and TikTok, in addition to other major social media platforms, are included in this category. Social media is currently having an impact on a diverse variety of industries, including education, business, and charitable giving, amongst others. The expansion of social networking platforms, which has resulted in an increase in the number of available jobs, is helping to prop up the economy on a global scale (Karjaluoto, 2015).

Even if social media has numerous benefits, there are also some drawbacks. For those focused on harming the feelings of others and damage their reputations, this media is used by those with malevolent purpose. In recent years, cyberbullying has become one of the most pressing issues on social media. A kind of bullying or harassment that takes place via the Internet is referred to as "cyberbullying" or "cyberharassment." Online harassment and cyberbullying are synonyms for one another. Cyberbullying has become a widespread issue, particularly among teens, as a result of the growth of the digital domain and the ongoing development of technology.

Several research initiatives have focused on developing methods to identify instances of cyberbullying via the use of machine learning. In a supervised machine learning method, the bag-of-words approach used to analyse a phrase in order to identify the sentimental and contextual characteristics of that phrase. This technique has just a 61.9 percent success rate in terms of accuracy. Support vector machines were used by the Ruminati project at the Massachusetts Institute of Technology in order to detect instances of cyberbullying within YouTube comments. The researcher combined the processes of detection and common sense thinking by making use of social traits. The accuracy of this project was improved to a level of 66.7 percent through the use of probabilistic modelling. A language-based approach that has a success rate of 78.5 percent has been devised by the authors as a method for identifying instances of cyberbullying. Through the use of the decision tree and the instance-based trainer, accuracy was accomplished. The researcher who wrote the study used personality, emotion, and mood as a feature to improve the accuracy of identifying instances of cyberbullying (Vandebosch, 2014).

Several models based on deep learning were able to uncover instances of cyberbullying as well. A model that is built on a Deep Neural Network used to detect instances of cyberbullying in real time by analysing data from the actual world. After performing an in-depth investigation of the topic, the authors used transfer learning in order to recognise instances of cyberbullying. Using the designs of deep neural networks that were suggested by the authors, it is possible to identify hate speech. A model of a convolutional neural network used to identify instances of cyberbullying. They used a strategy known as word embedding, which involves the repetitive embedding of words that are connected to one another. (Agrawal, 2018) study the one-of-a-kind subject of identifying cyberbullying using a collaborative process that makes use of data from social media. This challenge is tough to conquer because of the interaction between cross-modal connections across different techniques and structural correlations across social media sessions. As a consequence of this interplay, it is difficult to solve this obstacle. They propose XBully as a solution to these problems as a novel way for identifying instances of cyberbullying. This method begins by reformulating multi-modal social media data as a heterogeneous network and then makes an effort to construct node-bedding representations on the network (Ajlan, 2018).

Textual analysis has been the subject of a number of research on cyberbullying that have been conducted over the last couple of decades. The issue of cyberbullying is becoming much more convoluted as time goes on. The enormous variety of bullying data that found on social networking platforms is beyond the capabilities of conventional text analysis methods (Wang, 2020).

The authors proposed a multi-modal identification system as a means of coping with the most recent form of cyberbullying. This system include multi-modal information from social media, such as photographs, videos, comments, and timestamps. This type of information used to track the target of the bullying. They use hierarchical attention networks, for instance, to capture the social network session function and encode a broad variety of information from various types of media, such as video and image. One example of this found in the above sentence. In order to combat the most common kind of cyberbullying, a multi-modal cyberbullying detection system has been developed on the basis of these qualities. (Ramachandra, 2020).

In recent years, there has been a rise in the use of neural networks as an aid in the identification of cyberbullying that occurs online. Layers of long-short-term memory also be found in these neural networks, either alone or in conjunction with other kinds of layers. A new Neural Network model has been created by (Huang, 2018), and it has the potential to be used to identify instances of cyberbullying in written material. This novel design is a product of the combination of the convolutional architecture with the Long-Short-Term Memory (LSTM) architecture. In addition, they make use of stacked core layers in their design, which is more evidence of how their study increases the performance of Neural Networks. In addition, a mechanism that operates in a manner similar to that of a Support Vector Machine has been added into the design. It is possible to achieve a "Support Vector Machine-like activation" by employing L2 weight regularisation and a linear activation function in the activation layer, in conjunction with a Hinge loss function. This will allow for the achievement of the "Support Vector Machine-like activation." Because of this, a "Support Vector Machine-like activation" will occur as a result.

By constructing a machine learning system with three distinct qualities, the authors were able to circumvent the computational challenges that were involved with the identification of harassment in social networks. A minimal amount of monitoring is employed in the form of key phrases supplied by an expert that accurately forecast whether or not a student will engage in bullying behaviour. One student investigates the linguistic content of the text, while the other learner examines the social structure of the situation being discussed. This reveals instances of bullying. Through the use of nonlinear deep models for training, this blends decentralised word and graph-node representations. Training the model involves optimising an objective function that comprises a weak-supervision loss as well as a co-training loss (Al-garadi, 2016).

The creation of an effective detection model is of vital scientific interest since cyberbullying is a big problem that affects public health. Elements obtained from Twitter, such as user behaviour, content of tweets, and tweets themselves, have been included by the writers. They have created a supervised machine learning technique in order to detect instances of cyberbullying that occur on Twitter. An examination found that their newly created detection system had outcomes with a region under the receiver-operating characteristic curve of 0.943 percent and a F measure of 0.936 percent based on the features they gave. These figures are taken from the evaluation (Ghosh, 2017).

1.6 Project / Internship Planning

1.6.1 Project / Internship Development Approach and Justification

Approach:

The development approach for an Automatic Detection of Cyberbullying project using Data Science and Machine Learning broken down into the following steps:

Data collection: Collect a large dataset of text messages, social media posts, or other online content that has been labeled as either cyberbullying or not. This dataset diverse and representative of different types of cyberbullying behaviors, as well as different demographics of users.

Data preprocessing: Clean and preprocess the collected data by removing irrelevant or redundant information, correcting misspellings, and standardizing the format of the data. This step also involve feature engineering, which involves extracting relevant features from the data that used to train a machine learning model.

Model selection: Choose an appropriate machine learning algorithm and architecture to use for the project. This involve testing and comparing different algorithms to find the one that performs best on the collected dataset.

Model training: Train the selected machine learning model using the preprocessed data. This step involves feeding the data into the model and adjusting its parameters to optimize its performance.

Model evaluation: Evaluate the performance of the trained model using various metrics such as accuracy, precision, recall, and F1 score. This step involves testing the model on a separate dataset of labeled data to measure how well it is detecting cyberbullying.

Model deployment: Deploy the trained model into a production environment where it is used to automatically detect cyberbullying in real-time. This step involves integrating the model with other software tools or platforms.

Justification:

Cyberbullying has become a signific problem in today's digital age, affecting millions of people worldwide. It has severe consequences, including depression, anxiety, and even suicide. Detecting and preventing cyberbullying is crucial to creating a safe and supportive online community.

Automatic detection of cyberbullying using data science and machine learning offers a promising solution to this problem. Machine learning algorithms analyze large amounts of online data and identify patterns that indicate cyberbullying behavior. This approach more effective and efficient than manual monitoring, which time-consuming and error-prone.

Moreover, the use of machine learning algorithms in automatic detection of cyberbullying lead to the development of proactive measures, such as alerts or warnings to users who at risk of cyberbullying. This help prevent cyberbullying before it escalates and causes harm.

In conclusion, developing an Automatic Detection of Cyberbullying project using Data Science and Machine Learning is a vital and necessary step towards creating a safer and more supportive online community. It is help identify and prevent cyberbullying, which is crucial to ensuring the well-being of online users, especially children and teenagers who are the most vulnerable to this kind of abuse.

1.6.2 Project / Internship Effort and Time, Cost Estimation

Time Estimation:

Table 1.6.2: Project/ Internship Effort and Time

Activity	Estimated Duration
Choosing & Supervisor	3 days
Project Initiation	1 weeks
Initial Proposal	2 weeks
Final Proposal	1 week and 2 days
Submission of Client Consent Form	1 week and 3 days
Submission of Ethical Form	2 days
Building a Project Plan	1 week and 2 days
Presentation, Submission & Delivery	1 week and 4 days
Literature Review	2 days
Methodology	1 week
System Analysis	1 week and 2 days
Project Design	6 days
Project Development	6 days
Testing	6 days
Critical Analysis	4 days
Evaluation	1 days
Report Writing	6 days
Total Effort	12 week and 4 days (Approx. 3 months)

Cost Estimation:

Data Collection: The need to collect a large dataset of text messages, social media posts, emails, or other forms of digital communication for training and testing the Machine Learning models. The cost of data collection varies depending on the source and size of the data. The either collect the data manually or use web scraping tools.

Data Preprocessing: The collected data need to be preprocessed to remove any irrelevant or redundant information and convert the data into a suitable format for Machine Learning algorithms. This involves data cleaning, text normalization, tokenization, stopwords removal, and feature extraction. The cost of data preprocessing varies depending on the complexity of the data and the preprocessing techniques used.

Machine Learning Models: The need to build and train various Machine Learning models such as Natural Language Processing (NLP) models, deep learning models, or ensemble models for detecting cyberbullying. The cost of building and training Machine Learning models vary depending on the complexity of the models and the size of the dataset.

Infrastructure: The need to set up a robust infrastructure for running Machine Learning models, such as high-performance computing systems, cloud-based platforms, or dedicated servers. The cost of infrastructure varies depending on the size and complexity of the project.

Human Resources: The need to hire Data Scientists, Machine Learning Engineers, NLP Experts, and other professionals for developing and implementing the project. The cost of human resources varies depending on the skillset and experience of the professionals.

Testing and Evaluation: The need to perform testing and evaluation of the developed models to measure the accuracy, precision, recall, and F1 score of the models. The cost of testing and evaluation vary depending on the size and complexity of the dataset and the testing methods used.

1.6.3 Roles and Responsibilities

1.7 Project / Internship Scheduling (Gantt Chart/PERT/Network Chart)

	Feb-23																														
Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
Requirement Gathering and Building Flow																															
Identification of Problem and Literature Survey																															
Installing Anaconda distribution and understanding how to use																															
Importing the Dataset for cyberbulling																															
Importing required libraries and installing them																															
Understanding NLP concepts																															

	Mar-23																														
Task	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
Tokenization																															
Data cleaning: Removing urls and punctuations																															
removing Stopwords																															
Replacing Contractions																															
Extracting Cleaned Text																															
Research on Sentiment analysis																															
implementation of sentiment Identification																															
Feature Selection																															
Understanding polarity scores																															
Building pie chart for sentiment scores																															
Research on Selection of classification																															

	Apr-23																														
Task	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
Implementation of Naive bayes Algorithm																															
Training and Testing Split data																															
Defining the hyperparameters																															
Implementaion of Classification Report																															
Improving the Accuracy and precision by using hyperparameters																															
Comparision of Training and Testing Accuracy																															
Testing the Algorithm for different data inputs																															

2.System Analysis

2.1 Study of Current System

The current system of automatic detection of cyberbullying using Data Science and Machine Learning involves the use of various techniques and algorithms to analyze and classify text data. Here are some of the techniques and algorithms used in the current system:

Natural Language Processing (NLP): NLP is a subfield of Artificial Intelligence (AI) that deals with the processing and understanding of natural language text. NLP techniques are used in the automatic detection of cyberbullying to analyze and classify text data based on sentiment, tone, and context.

Machine Learning Algorithms: Machine Learning algorithms such as Naive Bayes, Support Vector Machines (SVM), Random Forest, and Logistic Regression are used in the automatic detection of cyberbullying to train and test the models on the dataset. These algorithms use various features such as bag of words, n-grams, and word embeddings to represent the text data.

Deep Learning: Deep Learning algorithms such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) are used in the automatic detection of cyberbullying to learn the features and patterns of the text data. These algorithms are used in the detection of cyberbullying in social media platforms such as Twitter and Facebook.

Text Preprocessing: Text preprocessing techniques such as text cleaning, tokenization, stopwords removal, stemming, and lemmatization are used in the automatic detection of cyberbullying to remove noise and irrelevant information from the text data. These techniques are used to convert the raw text data into a suitable format for analysis and classification.

Evaluation Metrics: Evaluation metrics such as accuracy, precision, recall, and F1 score are used in the automatic detection of cyberbullying to measure the performance of the models. These metrics are used to evaluate the effectiveness of the models in detecting cyberbullying accurately.

The current system of automatic detection of cyberbullying using Data Science and Machine Learning is still evolving, and researchers are constantly working on improving the accuracy and efficiency of the models. The development of more advanced algorithms and techniques such as deep learning and ensemble learning has shown promising results in the detection of cyberbullying in online platforms. However, there are still challenges in the automatic detection of cyberbullying, such as the diversity of the language used in online communication and the adaptation of the models to new forms of cyberbullying.

2.2 Problem and Weaknesses of Current System

Lack of high-quality labeled data: One of the biggest challenges in building an effective cyberbullying detection system is the lack of high-quality labeled data. Labeled data is necessary for supervised learning algorithms, which are widely used in cyberbullying detection. The lack of labeled data makes it difficult to train accurate models.

Difficulty in detecting sarcasm and irony: Cyberbullies often use sarcasm and irony in their messages, which difficult for machine learning algorithms to detect. These types of language nuances are difficult to understand even for humans and so it's even more difficult for machines.

High false positive rates: Automatic detection of cyberbullying have high false positive rates, which means that the system identify a message as cyberbullying even if it is not. This lead to incorrect accusations and cause harm to innocent individuals.

Limited language coverage: Most current automatic detection systems are trained on English language data and may not work well for other languages. Developing models that work well across multiple languages is still an area of active research.

Limited context awareness: Machine learning models have limited context awareness and may not be able to take into account the complex social and cultural factors that influence whether a message constitutes cyberbullying.

Lack of transparency: Machine learning models opaque and difficult to interpret. This lead to challenges in explaining how the model is making decisions and what factors it is taking into account.

Overall, while there have been significant advances in automatic detection of cyberbullying using data science and machine learning, there are still many challenges that need to be addressed to build more effective and reliable systems.

2.3 Requirements of New System

Following are the requirements for the new system.

Data Gathering: There are many different ways that occurrences of cyberbullying might take place. For instance, it is taking the form of sharing or posting inappropriate video content; sharing their hashtag with abusive language; or sharing their hashtag without the owner's consent, etc. However, the most common kind of cyberbullying involves the transmission of textual content. Therefore, the Twitter Api utilized to collect the textual information that is available from social media in real time.

For this purpose, Tweepy, which is a library written in Python, will extract the textual information from the posts containing the relevant hashtags.
Importing Libraries for Proposed System

Listed and imported here are all of the libraries that will be helpful in the identification of cyberbullying, including the following.

Tweepy: It is a Python package that makes gaining access to the Twitter API very simple to do.

Vadersentiment: It is called the Valence Aware Dictionary and Sentiment Reasoner, and it is a vocabulary and rule-based sentiment analysis tool that is specifically attuned to the sentiments that are conveyed in social media.

Nltk: Utilizing the tools and techniques provided by natural language processing in order to conduct an analysis of sentiment.

Sklearn: It offers a variety of effective tools for machine learning and statistical modelling, including classification, all of which are going to be utilized in the system that we have presented.

Data Cleaning: This process is useful for the detection as well as the correction of the corrupt data and also identify the inaccurate data from the given dataset. This process also replaces, modify as well as delete the coarse data available in the dataset.

Tokenization: With the help of the tokenization process model is able to divide the given text into the various segments and convert it into the form of the tokens which are meaningful word.

Lemmatization: Lemmatization process is a linguistic based process which is used for the grouping of an inflected words for the analysis as a single block and identify the word's lemma which is a dictionary form.

TF-IDF Vectorizer: The method of Term Frequency-Inverse Document Frequency is a type of text vectorizer that is utilized for the purpose of transforming the text into a vector that utilized.

Term frequency is used for the identification of how important a specific term in the document and it represents the matrix form of the text in which row is used for the representation of the number of documents while column is used for the representation of the distinct terms available in the documents.

The number of papers that contain a particular term is referred to as its document frequency. The number of documents that use a term provides an indication of how prevalent it is.

Naïve Bayes Classifier: Bayes' Theorem-based classification algorithms further broken down into subsets, one of which is called Naive Bayes classifiers. There is not just one algorithm; rather, there is a family of algorithms that all adhere to the same notion, which is that each pair of features being categorized is independent of the others. There is not just one algorithm; rather, there is a family of algorithms that all adhere to this idea.

2.4 System Feasibility

4.4.1 Does the system contribute to the overall objectives of the organization?

The system of automatic detection of cyberbullying using data science and machine learning contribute to the overall objectives of an organization. Cyberbullying has a significant impact on individuals, communities, and organizations. It is result in decreased productivity, increased absenteeism, decreased job satisfaction, and decreased morale. Addressing cyberbullying is not only the right thing to do, but it also makes good business sense.

Implementing a system that automatically detect cyberbullying provide several benefits to an organization. Firstly, it is help protect the well-being of employees and ensure a safe and healthy work environment. Secondly, it is help protect the reputation of the organization by demonstrating a commitment to addressing cyberbullying. Thirdly, it is help prevent legal liability by addressing cyberbullying before it escalates to a level that requires legal action.

Furthermore, the implementation of such a system also contribute to the overall objectives of an organization in terms of efficiency and effectiveness. By automating the process of cyberbullying detection, organizations save time and resources that have otherwise been used to manually monitor and address cyberbullying. This allows organizations to focus on their core business objectives while also ensuring a safe and healthy work environment.

In conclusion, the implementation of an automatic detection system for cyberbullying using data science and machine learning contribute to the overall objectives of an organization by protecting the well-being of employees, safeguarding the reputation of the organization, preventing legal liability, and improving efficiency and effectiveness.

2.4.2 the system be implemented using the current technology and withinthe given cost and schedule constraints

The project divided into several phases, including data collection, data preprocessing, feature extraction, model training, and model evaluation. The cost and schedule constraints will depend on the specific requirements of the project, such as the amount of data, the complexity of the model, and the performance metrics.

Data collection involve web scraping, API calls, or other methods to gather social media data that contains instances of cyberbullying. The data will need to be preprocessed to remove irrelevant information, handle missing data, and convert the text into a format that used for feature extraction. Feature extraction involve techniques such as word embeddings, sentiment analysis, and topic modeling to capture relevant information from the text.

Model training involve supervised learning techniques such as logistic regression, support vector machines, or neural networks. The performance of the model evaluated using metrics such as accuracy, precision, recall, and F1 score.

The cost and schedule constraints managed by carefully selecting the tools and technologies used for each phase of the project, as well as optimizing the data processing and model training pipelines. Additionally, the project broken down into smaller milestones to allow for iterative development and testing.

Overall, with proper planning and execution, an automatic detection system for cyberbullying using data science and machine learning implemented within the given cost and schedule constraints.

4.4.3 the system be integrated with other systems which are already inplace?

the system is being developed for a social media platform; it is integrated with the existing moderation tools to flag instances of cyberbullying for review by human moderators. The system also be integrated with reporting tools, allowing users to report instances of cyberbullying to the platform for further investigation.

The integration of the automatic detection system with other systems will require careful consideration of the data flow, APIs, and interfaces between the different systems. The system will need to be designed to work seamlessly with the existing systems, without causing any disruption to the user experience or performance of the platform.

In addition, the system needs to comply with legal and regulatory requirements related to data privacy and security, especially if it involves sharing user data with other systems. Appropriate measures such as data encryption, access controls, and data anonymization need to be implemented to ensure that user data is protected and used only for legitimate purposes.

Overall, the integration of the automatic detection system for cyberbullying with other systems will require careful planning, collaboration, and technical expertise to ensure that the system functions effectively and seamlessly within the existing ecosystem.

2.5 Activity / Process in New System / Proposed System

The activity / process in the proposed system of the Cyber Bullying project using Data Science & Machine Learning divided into several stages:

Data Collection: Data is collected from various sources such as social media platforms, online forums, chat rooms, and blogs. The collected data includes text, images, videos, and other multimedia content that used to train the machine learning models.

Data Preprocessing: Once the data is collected, it needs to be preprocessed to remove irrelevant information, duplicate content, and data noise. This stage involves tasks such as cleaning, filtering, and formatting the data.

Data Labeling: In this stage, the data is labeled based on whether it contains cyberbullying content or not. This is done either manually or through automated labeling techniques.

Feature Extraction: Feature extraction involves identifying relevant features from the labeled data. This stage helps in identifying patterns and trends in the data that is used to train the machine learning models.

Model Selection: Based on the features extracted from the data, various machine learning models are selected and trained. This stage involves tasks such as choosing the appropriate algorithms, setting parameters, and tuning the models.

Model Evaluation: In this stage, the trained models are evaluated based on their accuracy, precision, recall, and other metrics. The evaluation helps in identifying the best performing model that is used in the final system.

System Development: Based on the selected machine learning model, the final system is developed. This system a web-based application, mobile app, or any other platform that allows users to detect and report cyberbullying content.

Deployment and Monitoring: Once the system is developed, it is deployed and monitored to ensure that it is performing as expected. This stage involves tasks such as testing, bug fixing, and system maintenance.

Continuous Improvement: As new data is collected and more insights are gained, the system continuously improved to enhance its performance and accuracy. This involves updating the machine learning models, retraining the models, and improving the system's features and functionalities.

2.6 Features of New System / Proposed System

The proposed system for the Cyberbullying project using Data Science and Machine Learning will have several features. Here are some of the key features:

Data Collection: The system will collect data from various sources, such as social media platforms, messaging apps, and other online communication channels. This data will include text messages, comments, posts, and other forms of online content.

Preprocessing: Once the data is collected, the system will preprocess it to remove any noise or irrelevant information. This will involve text normalization, stemming, and stop-word removal, among other techniques.

Sentiment Analysis: The system will perform sentiment analysis on the preprocessed data to determine the overall sentiment of each message or post. This will help identify instances of cyberbullying, as negative or aggressive sentiment a key indicator.

Machine Learning Algorithms: The system will use various machine learning algorithms to analyze the data and identify patterns of cyberbullying behavior. This will include techniques such as clustering, classification, and regression analysis.

User Profiling: The system will build profiles of users based on their online behavior and communication patterns. This will help identify individuals who are more likely to engage in cyberbullying behavior and those who are more likely to be targets.

Reporting: The system will generate reports on cyberbullying incidents to help identify trends and patterns over time. This information used to develop targeted interventions to prevent cyberbullying and support victims.

Overall, the proposed system for the Cyberbullying project using Data Science and Machine Learning will leverage advanced technologies to detect and prevent instances of cyberbullying. By analyzing large amounts of online data, the system will be able to identify patterns of behavior and provide actionable insights to help prevent cyberbullying and support victims.

2.7 List Main Modules / Components / Processes / Techniques of New System /Proposed System

The main modules/components/processes/techniques of a proposed system for automatic detection of cyberbullying using Data Science and Machine Learning include:

Data Collection and Pre-processing: Collecting relevant data from different sources, pre-processing the data by removing irrelevant data, removing stop words, and transforming data into a standardized format.

Feature Extraction: Extracting relevant features from pre-processed data that used for classification.

Machine Learning Algorithms: Applying supervised machine learning algorithms such as Naive Bayes, Random Forest, Decision Trees, Support Vector Machines (SVM), etc. to classify the data as bullying or not.

Model Evaluation: Evaluating the performance of the machine learning models using metrics such as accuracy, precision, recall, F1 score, etc.

Deployment: Deploying the machine learning models to a web application or API that automatically detect cyberbullying.

Continuous Learning: Updating the machine learning models periodically to keep them up-to-date with the latest trends and patterns in cyberbullying.

Natural Language Processing (NLP): Using NLP techniques to analyze and understand the meaning of the text, and to identify the sentiment and emotions of the text.

Social Network Analysis: Analyzing the social networks to detect the patterns of cyberbullying, such as identifying the bullies, victims, and bystanders.

Data Visualization: Visualizing the results of the machine learning models using graphs, charts, and other visualization techniques to help users understand the patterns and trends in cyberbullying.

2.8 Selection of Hardware / Software / Algorithms / Methodology / Techniques /Approaches and Justification

Hardware:

The hardware selection for a cyberbullying detection project primarily depends on the data processing requirements, data storage capacity, and the machine learning algorithm used in the project. Here are some hardware recommendations that considered for this project:

CPU: Cyberbullying detection requires extensive data processing, and a powerful CPU handle complex algorithms efficiently. Intel Core i7 or i9 processors considered for this project.

GPU: Machine learning algorithms such as deep learning require large amounts of data processing. Using a GPU significantly reduce the time required for processing the data.

RAM: The size of the RAM determines the amount of data that processed at a time. At least 16GB or 32GB of RAM is recommended for this project.

Storage: The storage requirement for this project depends on the size of the dataset used. An SSD with a storage capacity of at least 1TB used for this project.

Networking: Cyberbullying detection projects require a stable and reliable internet connection to collect data from various sources. A high-speed internet connection with low latency is recommended.

Sensors: Depending on the project's scope, sensors like microphones and cameras required to collect data for analysis.

In summary, a cyberbullying detection project requires a powerful CPU and GPU, sufficient RAM, ample storage capacity, and a reliable internet connection. The use of cloud services and sensors also be required depending on the project's scope.

Software:

Programming Language: Python is one of the most widely used programming languages for data science and machine learning projects. Therefore, it is a good choice for this project. Additionally, R is another option that is used for data analysis and visualization.

Machine Learning Libraries: Several machine learning libraries used for this project, including TensorFlow, PyTorch, Scikit-learn, Keras, and NLTK. These libraries provide useful tools for data pre-processing, feature engineering, model training, and evaluation.

Natural Language Processing Tools: Natural Language Processing (NLP) is a crucial component of this project. Some popular NLP tools include SpaCy, Stanford NLP, and Gensim. These tools offer features such as text pre-processing, entity recognition, sentiment analysis, and language translation.

Visualization Tools: Data visualization tools like Matplotlib, and Seaborn be used to create visualizations that help to understand patterns and trends in the data.

Development Environment: A code editor such as Jupyter be used as a development environment to write and debug code.

Version Control: Version control software like Git and GitHub help manage and track changes to project code.

Algorithms:

Random Forest: Random Forest is a type of ensemble learning algorithm that combines multiple decision trees to make a more accurate prediction. It be applied to classification tasks, which is what cyberbullying detection fall under. By training a Random Forest model on a dataset of labelled examples of cyberbullying and non-cyberbullying text, the model learns to classify new, unseen text as either cyberbullying or not.

Natural Language Processing: Natural Language Processing, is a field of study that focuses on enabling computers to understand and process human language. NLP techniques be used to analyze text data and identify patterns and features that be used for classification. For example, sentiment analysis be used to identify the overall tone of a message, which be useful in detecting cyberbullying.

Count Vectorizer: Count Vectorizer is a technique for converting text data into numerical vectors that be used as input for machine learning algorithms. It works by counting the frequency of each word in a document and creating a vector with these counts. This be useful for cyberbullying detection as it helps to identify patterns and features in the text that are indicative of cyberbullying.

To use these techniques for automatic detection of cyberbullying, a dataset of labelled examples of cyberbullying and non-cyberbullying text need to be collected and pre-processed. The text data need to be cleaned, tokenized, and transformed using techniques such as Count Vectorizer. Then, a Random Forest model trained on the transformed data, using NLP techniques to extract useful features and patterns from the text. The resulting model then be used to automatically classify new text as either cyberbullying or not.

Methodology:

Requirements gathering: This phase involves gathering information about the problem of cyberbullying, the available data sources, and the desired outcomes of the detection system.

Data collection and preprocessing: In this phase, relevant data related to cyberbullying will be collected from various sources such as social media platforms, online forums, and messaging applications. The collected data will then be preprocessed by removing noise, irrelevant information, and performing feature engineering.

Model selection and training: In this phase, appropriate machine learning algorithms such as Support Vector Machines (SVM), Random Forests, or Deep Learning models will be selected and trained using the preprocessed data. The models will be evaluated using cross-validation techniques and hyperparameter tuning to achieve optimal performance.

Model integration and testing: In this phase, the trained models will be integrated into a detection system that automatically detect cyberbullying in real-time. The system will be tested using a dataset containing cyberbullying instances to assess its accuracy, precision, and recall.

Deployment and maintenance: Once the detection system is deployed, it will be continuously monitored to ensure its effectiveness in detecting cyberbullying instances. The system will also be updated regularly to adapt to new trends and patterns of cyberbullying.

Justification:

The proposed methodology is based on the Waterfall model, which is a traditional software development process that involves sequential phases from requirements gathering to deployment. This model provides a clear roadmap for the development of the detection system, ensuring that all the necessary steps are taken to create a reliable and accurate system.

Data Science and Machine Learning techniques are ideal for detecting cyberbullying because they analyze large amounts of data and identify patterns that not be visible to human observers. By using machine learning models, the system learn to identify cyberbullying instances based on various features such as text, images, and audio.

Moreover, the proposed methodology involves continuous monitoring and maintenance of the detection system, ensuring that it remains effective in detecting cyberbullying instances even as trends and patterns change over time. This approach ensures that the system remains up-to-date and relevant in the face of evolving cyberbullying tactics.

Overall, the proposed methodology offers a comprehensive approach to automatic detection of cyberbullying, leveraging the power of data science and machine learning techniques to develop a reliable and accurate detection system.

3. System Design

3.1 System Design & Methodology

Use case Diagram

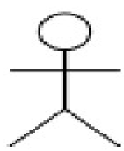
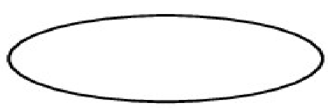
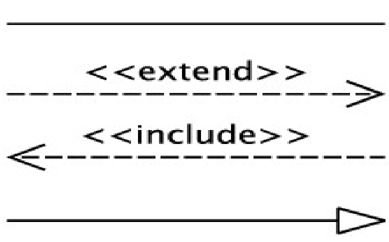
A use case diagram is a type of UML diagram that represents the interactions between actors and a system, showing the various ways in which, the system be used.

In a use case diagram, actors are represented as stick figures, and use cases are represented as ovals. The actors represent individuals or systems that interact with the system being modeled, while the use cases represent specific tasks or functions that the system performs.

Use case diagrams are useful for visualizing the functionality of a system and for communicating the requirements of the system to stakeholders. They help identify the actors and use cases that are most critical to the system, and help in the design and development of the system by providing a clear understanding of how the system behave in different scenarios.

Use case diagrams also be used to identify potential issues or problems with a system before it is built, and to help ensure that the system is meeting the needs of the stakeholders. By providing a visual representation of the system's functionality and interactions, use case diagrams be a valuable tool in the development of complex software systems.

Table 3.1.1 Symbols and components of use case Diagram

Symbol	Reference Name
	Actor
	Use case
	Relationship

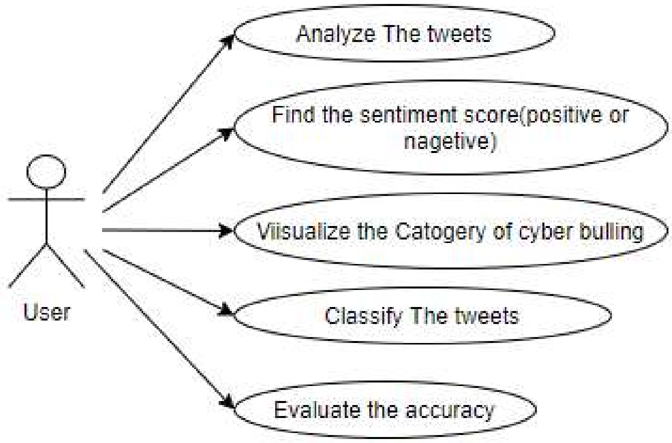


Fig 3.1.1: User Use case Diagram

System Flow chart

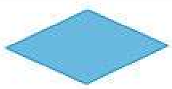
A flowchart is a graphical representation of a process or system that uses symbols and arrows to show the flow of information or steps in a process. It is a visual tool that helps to illustrate the sequence of events or decision points in a process or system.

In a flowchart, different shapes are used to represent different types of information or actions. For example, rectangles represent processes or actions, diamonds represent decision points or branches, and arrows show the flow of information or steps between different shapes.

Flowcharts are commonly used in software engineering, business analysis, and project management to document processes and procedures, and to help identify potential problems or improvements in a system. They also be used to explain complex processes to stakeholders and team members, and to help with process improvement efforts.

Flowcharts range from simple diagrams that show a few steps in a process to complex diagrams that show multiple branches and decision points. They created using specialized software or by hand, and customized to suit the needs of the specific process or system being documented.

Table 3.1.2: Symbols and components of Flow chart

Symbol	Name	Function
	Start/end	An oval represents a start or end point
	Arrows	A line is a connector that shows relationships between the representative shapes
	Input/Output	A parallelogram represents input or output
	Process	A rectagle represents a process
	Decision	A diamond indicates a decision

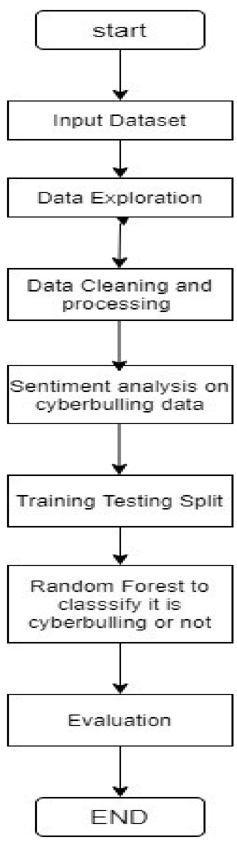


Fig 3.1.2: Flow chart

4. Implementation

4.1 Implementation Platform / Environment

Lemmatization:

Lemmatization is a natural language processing technique that involves reducing words to their base or root form, known as the lemma, in order to normalize and standardize words. The lemma is the canonical or dictionary form of a word, and it represents the common, uninflected or unconjugated form of a word that serves as a base for generating different inflected or conjugated forms.

Lemmatization is commonly used in various NLP tasks, such as text analysis, information retrieval, and machine translation, to reduce words to their basic form and consolidate different forms of the same word into a single representation. By doing so, it helps in reducing vocabulary size, simplifying text analysis, and improving the accuracy of text processing tasks.

For example, the lemma of the words "running," "ran," and "runs" would be "run." Similarly, the lemma of "better" and "best" would be "good." Lemmatization takes into account the morphological rules of a language and uses dictionaries or linguistic rules to map words to their corresponding lemmas. It is different from stemming, which involves removing suffixes from words to obtain their base form without considering linguistic rules or the actual meaning of the word.

Lemmatization is typically performed using specialized software libraries or tools in programming languages such as Python, NLTK, spaCy, and Stanford NLP, among others. It is an important preprocessing step in many NLP applications that require word normalization and standardization for effective text analysis and processing.

Stop Word Removal:

Stop word removal is a common text preprocessing technique in natural language processing (NLP) that involves removing common words, known as stop words, from a piece of text. Stop words are typically very common words that do not carry much meaning and are often used to connect or structure sentences, but are generally considered to be of low importance in text analysis tasks.

The rationale behind stop word removal is to reduce the noise and unnecessary information in a text, and to focus on the more meaningful content words that carry important semantic meaning. By removing stop words, the remaining words in the text are usually more relevant to the analysis or processing being performed, and the resulting text may be more concise and meaningful for downstream tasks such as sentiment analysis, text classification, or information retrieval.

Examples of stop words in English include "the," "and," "is," "in," "of," "to," "that," and "it," among others. However, the specific set of stop words may vary depending on the context, language, and application being used.

Stop word removal can be done using various approaches, including rule-based methods, dictionary-based methods, or using pre-built stop word lists. Some popular programming libraries for NLP, such as NLTK, spaCy, and scikit-learn in Python, provide built-in stop word lists and functions for stop word removal. However, it's important to note that stop word removal may not always be beneficial in every NLP task, as stop words can sometimes carry important information, especially in certain contexts, and the decision to remove them should be made carefully based on the specific requirements of the task at hand.

Naïve Bayes Classifier:

Naive Bayes Classifier is a simple probabilistic machine learning algorithm that is commonly used for classification tasks, such as text classification, spam detection, and sentiment analysis. It is based on the probabilistic principles of Bayes' theorem, which is a mathematical formula that calculates the conditional probability of an event based on the prior probability and the likelihood of the event.

The "naive" aspect of the Naive Bayes Classifier refers to the assumption that the features used for classification are conditionally independent, meaning that the occurrence of one feature does not depend on the occurrence of another feature. This simplifying assumption allows the algorithm to be computationally efficient and makes it easy to implement, but it may not always hold true in real-world data.

The Naive Bayes Classifier works by building a model based on a labeled training dataset, where the input data is represented as feature vectors and the corresponding class labels are known. The algorithm estimates the probabilities of each feature occurring in each class, as well as the prior probabilities of each class, from the training data. Then, for a new, unseen input, it calculates the posterior probability of each class given the input features using Bayes' theorem, and assigns the input to the class with the highest posterior probability.

One of the key strengths of the Naive Bayes Classifier is its ability to handle high-dimensional data with a relatively small amount of training data. It is also fast and scalable, making it suitable for large datasets. Additionally, Naive Bayes Classifier is robust to irrelevant features and can handle categorical and continuous features.

However, the Naive Bayes Classifier has some limitations. The assumption of feature independence may not always hold true in real-world data, which can lead to suboptimal performance. It is also a "naive" classifier, meaning it may not capture complex relationships between features, and may not perform well in scenarios where feature interactions are important. Despite these limitations, Naive Bayes Classifier is a popular and widely used algorithm in many classification tasks due to its simplicity, efficiency, and good performance in many practical scenarios.

4.2 Process / Program / Technology / Modules Specification(s)

To achieve the objectives of the proposed system, model is integrated with the following modules and the algorithms, illustrated in the below figure.

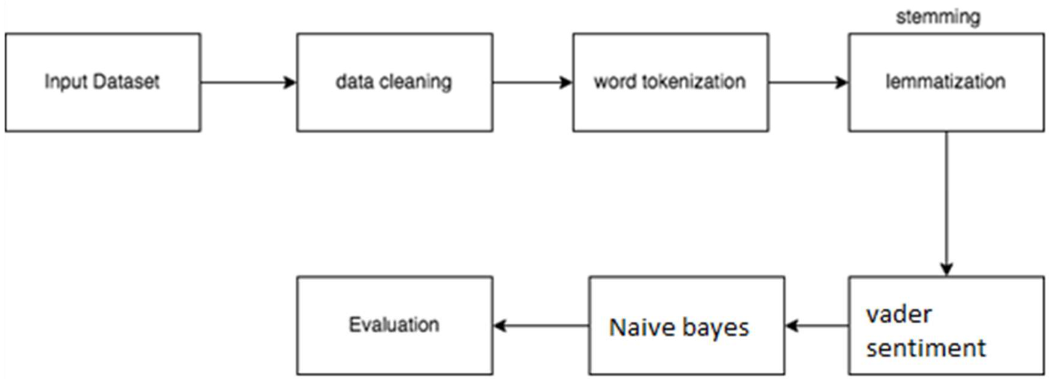


Fig 6.2: Flow of the proposed system

Data Cleaning:

This process is useful for the detection as well as the correction of the corrupt data and also identify the inaccurate data from the given dataset. This process also replaces, modify as well as delete the coarse data available in the dataset.

Tokenization:

With the help of the tokenization process model is able to divide the given text into the various segments and convert it into the form of the tokens which are meaningful word.

Lemmatization:

Lemmatization process is a linguistic based process which is used for the grouping of an inflected words for the analysis as a single block and identify the word's lemma which is a dictionary form.

TF-IDF Vectorizer:

Term Frequency-Inverse Document Frequency is a text vectorizer process which is used for the transformation of the text into the usable vector.

Term frequency is used for the identification of how important a specific term in the document and it represents the matrix form of the text in which row is used for the representation of the number of documents while column is used for the representation of the distinct terms available in the documents.

Document frequency is the number of documents containing a specific term. Document frequency indicates how common the term is.

Naïve Bayes Classifier:

Naive Bayes classifiers are a subset of Bayes' Theorem-based classification methods. There isn't just one algorithm; rather, there's a family of algorithms that all adhere to the same idea, which is that each pair of features being classed is independent of the others.

4.3 Finding / Results / Outcomes

4.3.1 Technical Requirements

Proposed system employed the following techniques and library resources.

Nltk: This library is in charge of carrying out a wide range of natural language processing tasks, such as bag of words, tokenization, and analysis of different data-related issues.

Sklearn: To offer support for both supervised and unsupervised learning techniques.

Vader Sentiment: This method, when combined with natural language processing, allows for the seamless integration of APIs to carry out a wide range of tasks, such as sentiment analysis.

4.3.2 Data Gathering

Data Exploration

As social media usage becomes increasingly prevalent in every age group, a vast majority of citizens rely on this essential medium for day-to-day communication. Social media's ubiquity means that cyberbullying can effectively impact anyone at any time or anywhere, and the relative anonymity of the internet makes such personal attacks more difficult to stop than traditional bullying.

On April 15th, 2020, UNICEF issued a warning in response to the increased risk of cyberbullying during the COVID-19 pandemic due to widespread school closures, increased screen time, and decreased face-to-face social interaction. The statistics of cyberbullying are outright alarming: 36.5% of middle and high school students have felt cyberbullied and 87% have observed cyberbullying, with effects ranging from decreased academic performance to depression to suicidal thoughts.

In light of all of this, this dataset contains more than **47000** tweets labelled according to the class of cyberbullying:

- Age;
- Ethnicity;
- Gender;
- Religion;
- Other type of cyberbullying;
- Not cyberbullying

The data has been balanced in order to contain ~8000 of each class.

6.3.3 Data preparation

Data in Data Frame

	tweet_text	cyberbullying_type
0	In other words #katandandre, your food was cra...	not_cyberbullying
1	Why is #aussietv so white? #MKR #theblock #ImA...	not_cyberbullying
2	@XochitlSuckkks a classy whore? Or more red ve...	not_cyberbullying
3	@Jason_Gio meh. :P thanks for the heads up, b...	not_cyberbullying
4	@RudhoeEnglish This is an ISIS account pretend...	not_cyberbullying

Fig 4.3.1: Data scrapped from twitter

The read csv () function uses the Pandas library to read the data, which is then stored in the data frame.

4.3.4 Converting categories into numbers

In dataset there are 5 different categories that needs to be converted from string to number to label the category for classification.

Syntax :

```
df["sentiment"].replace({"religion": 1, "age": 2, "gender": 3, "ethnicity": 4, "not_cyberbullyin
g": 5}, inplace=True)
```

4.3.5 Data-Pre-processing and cleaning

Tweets with a large number of characters that have no meaning are gathered and processed once the data is collected.
Since sentiment analysis cannot be performed in this scenario, several strategies such as word contractions, emotion emojis, and word cleaning are employed to remove or substitute the offending data.

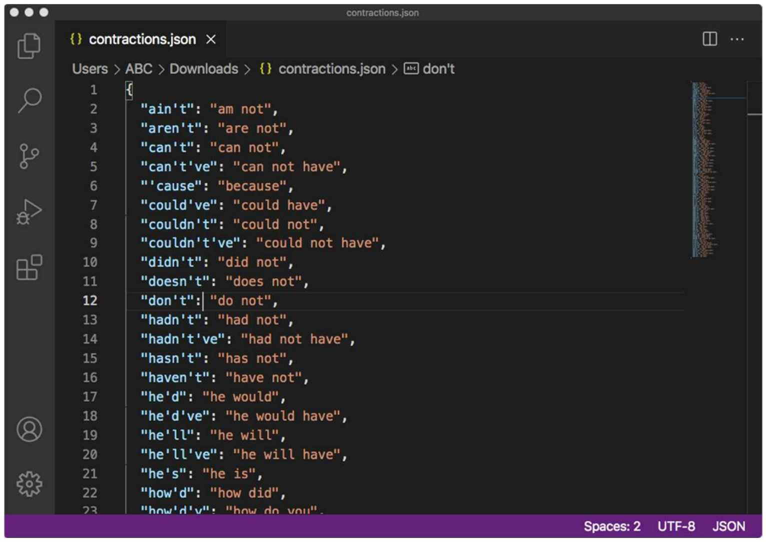


Fig 4.3.2: Converting Contractions words

Syntax:

```
with open ('contractions.json', 'r') as f:
    contractions_dict = json.load (f)contractions = contractions_dict
```

Emoji Conversion

In the context of social media, rather than omitting the emoji altogether, it is possible to determine the sentiment behind the expression by use the regular expression module of Python.

Syntax:

```
import re
def emoji(tweet):
    # Smile -- :), : ), :-), (:, ( :, (-:, :') , :O
    tweet = re.sub(r'(:\s?\)|:-\)|\(\s?:|\(-:|:\'\)|:O)', ' positiveemoji ', tweet)
    # Laugh -- :D, : D, :-D, xD, x-D, XD, X-D
    tweet = re.sub(r'(:\s?D|:-D|x-?D|X-?D)', ' positiveemoji ', tweet)
    # Love -- <3, :*
    tweet = re.sub(r'(<3|:\*)', ' positiveemoji ', tweet)
    # Wink -- ;-), ;), ;-D, ;D, (;,  (-; , @-)
    tweet = re.sub(r'(;-?\)|;-?D|\(-?;|@-\))', ' positiveemoji ', tweet)
    # Sad -- :-(, : (, :(, ):, )-:, :-/ , :-|
    tweet = re.sub(r'(:\s?\(|:-\(|\)\s?:|\)-:|:-/|:-\|)', ' negetiveemoji ', tweet)
    # Cry -- :,(, :'(, :"(
    tweet = re.sub(r'(:,\(|:\'\(|:"\()', ' negetiveemoji ', tweet)
    return tweet
```

Processing and cleaning the tweets

Because of this, pre-processing and post-processing are utilized in order to clean up the Twitter data, eliminate or replace the words, and extract the emotion from the data that was collected from Twitter.

	text	sentiment	text_clean
0	In other words #katandandre, your food was cra...	5	word katandandr food crapilici mkr
1	Why is #aussietv so white? #MKR #theblock #ImA...	5	aussietv white mkr theblock today sunris studi...
2	@XochitlSuckkks a classy whore? Or more red ve...	5	classi whore red velvet cupcak
3	@Jason_Gio meh. :P thanks for the heads up, b...	5	meh p thank head concern anoth angri dude twitter
4	@RudhoeEnglish This is an ISIS account pretend...	5	isi account pretend kurdish account like islam...

Fig 4.3.3: Processed and clean tweets

In this part, the re.i.e., regular expression library is imported, and the function process tweet is used to pre-process each tweet. In the tweets represented by the syntax shown above, any words or symbols that aren't necessary have been removed or replaced. The preceding function starts by lowering the string using the lower () method, which converts the tweet to a lower string. The sub technique is then used to strip the tweet of any usernames, numerals, URLs, emoticons, and single characters. Finally, the tweet is checked to ensure that it does not contain any emotion. After that, any punctuation marks, such as ([], are removed, because the processed tweet should only contain the words that convey meaning and express emotion.

From the Figure 4.4.2 it is obvious that hashtags are taken from the text, and it is also clear that all tweets are handled when the process tweet function is executed.

It will be used to categorise abusive phrases and positive terms in order to collect the user's opinion regarding a particular subject in preparation for the following step, which will involve the creation of a visual depiction of that feeling.

Lemmatization

After the data has been pre-processed, understanding the meaning of the terms used in tweets as well as the words' roots is necessary before modelling the data. Lemmatization, which is a rule-based technique, is utilised because of this reason.

Lemma is the name given to the root word in the process of lemmatization. The canonical form, dictionary form, or citation form of a group of words is referred to as a lemma (sometimes written as lemmas or lemmata). As an illustration, runs, running, and ran are all variations of the word run; hence, run is the lemma of all of these words.

Syntax:

from nltk.stem.wordnet import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

tokenized_tweet = tokenized_tweet.apply(lambda x: [lemmatizer.lemmatize(i) for i in x]) .

Here every processed tweet goes through a lemmatization and it convert it into a root word wherever required.

Removing Stopwords

Some words remain unemotional even after lemmatization; these are termed stopwords, and they can be eliminated from a phrase by first breaking it down into individual words and then checking each one against a pre-defined list of stopwords provided by NLTK.

Below is an example of using the nltk. corpus module to import a collection of stopwords into the syntax.

Syntax:

```
stopwords = nltk.corpus. Stopwords.words('english')
for i in range(len(tokenized_tweet)):
    tokenized_tweet[i] = ' '.join(tokenized_tweet[i])
data['processed_tweet'] = tokenized_tweet
```

4.3.6 Vader Sentiment Analysis .

In analyses text and determine polarity (positive or negative) and emotion strength, the suggested system employs a model called Valence Aware Dictionary for the Sentiment Reasoning (VADER). The VADER sentiment analyzer's capabilities are imported into the NTLK package model so it may be used to unlabeled textual input.

The key to the success of this approach to sentiment analysis is the dictionary's ability to convert the lexical data into an emotion intensity metric. To arrive at the overall sentiment score, we add together the weights assigned to each word in the text.

SentimentIntensityAnalyzer (), which is imported from the Vader module, is used to determine each tweet's Polarity Score; this function returns the tweet's negative, positive, and compound score.

Syntax:

```
analyzer = SentimentIntensityAnalyzer ()
def print_sentiment_scores(sentence):
snt = analyser.polarity_scores(sentence)
data['neg']= data['processed_tweet'].apply(lambda x:analyzer.polarity_scores(x)['neg'])
data['neu']= data['processed_tweet'].apply(lambda x:analyzer.polarity_scores(x)['neu'])
data['pos'] = data['processed_tweet'].apply(lambda x:analyzer.polarity_scores(x)['pos'])
data['compound']=data['processed_tweet'].apply(lambda x:analyzer.polarity_scores(x)['compound'])
```

	processed_tweet	neg	pos	compound	sentiment
0	can fathom how tiktok work am officially old	0.000	0.000	0.0000	neutral
1	nomore snapchat no tiktok ig business twitter ...	0.172	0.203	0.1027	positive
2	from this video we can know yeri is such mood ...	0.084	0.000	-0.2960	negative
3	tiktok tiktok update with jake and jungwon enh...	0.000	0.000	0.0000	neutral
4	tiktok tiktok update with jake and jungwon enh...	0.000	0.000	0.0000	neutral

	text	sentiment
23285	U can't speak who was given a BIG land and 65 ...	religion
7670	I need to get this college shit going I'm bore...	not_cyberbullying
16317	I think if I want to shiplap every wall in my ...	religion
17509	As opposed to what you doing. It's amazing how...	religion
603	Which was my first choice.	not_cyberbullying
20159	Wow @chancetherapper did you read Ye's intervi...	religion
44689	I'm ashamed I'm half white ! I don't want to b...	ethnicity
45473	UPGRADES PEOPLE UPGRADES! ur first drawing was...	ethnicity
14596	@FuzzyMooseBaby I expect nothing advanced from...	gender
21846	Why US? Ask you king first to ban India you id...	religion

Fig 4.3.4: Sentiment Polarity Scores and classification categories

It is now quite easy to design the bar chart according to the categories that were received after label distribution of the category.

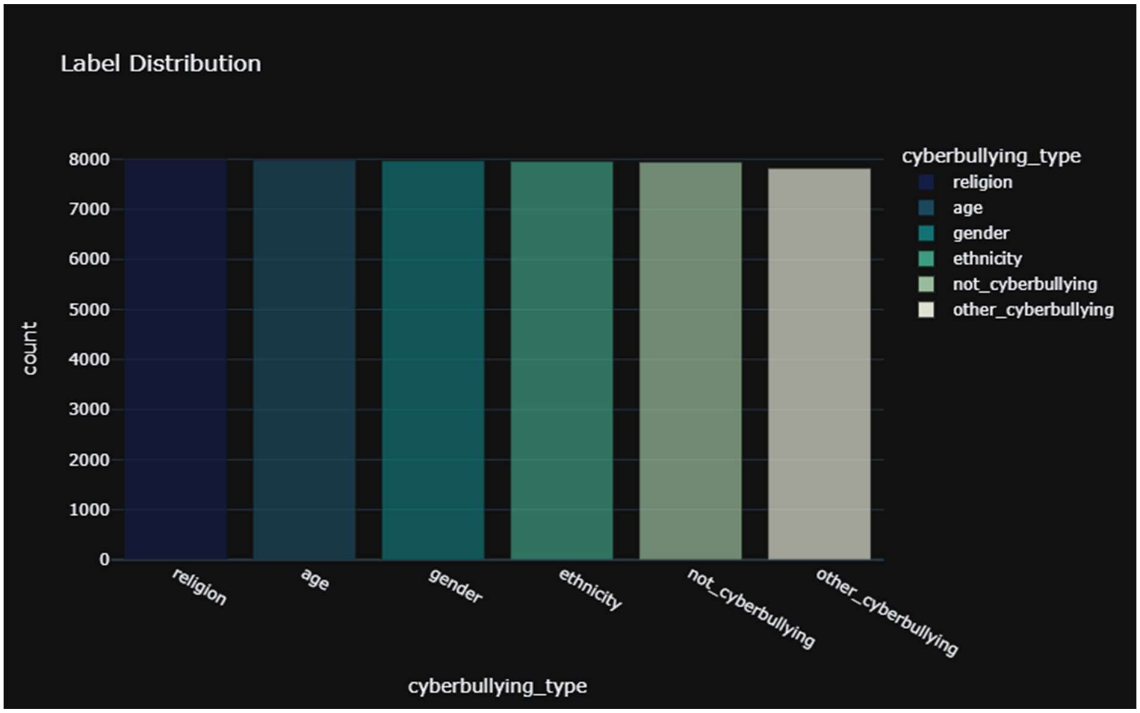


Fig 4.3.5: Graphical Representation of cyberbullying type

6.3.7 Probabilistic Classifier Naive Bayes

A naive bayes classifier, using the python language's built-in nltk package, that predicts a sentiment score between -1 and 1, by performing the following steps.

- Takes out the duplicate feature from the options
- Apply a method known as Stop words to generate a feature vector.
- Create a data set containing the feature vector.

In order for the model to be in a position to determine the F1 score, the activities that were outlined up top need to be carried out. After that, the naive bayes algorithm needs to be applied in order to separate the different classes based on the attributes that are currently available by computing the probabilities of the attributes that belong to each class. This is done in order to separate the classes based on the attributes that are currently available. In this scenario, the naive bayes classifier operates under the presumption that each characteristic can be understood independently of the other attributes. Using the conditional probability, which is based on the idea that the likelihood of any characteristic is independent from the likely of any other feature, we are able to successfully complete this task. This makes it easier for the model to compute and accurately identify the feature, which ultimately leads to an improvement in the model's overall performance. The implementation of the naive bayes classifier is provided for your review further down in this article.

	text	sentiments	rf_predicted
0	"@2015seniorprobs: I probably would not mind s...	age	age
1	CauseWereGuys: On my way to fuck yo bitch me ...	age	age
2	#HonestyHour in middle school hoes used to cal...	age	age
3	#InMiddleSchool i was fat as hell and was a ba...	age	not_cyberbullying
4	#southcarolina has a high school graduation ra...	age	age
...	...	...	...
95	time to eat with my bae swalscha Ã°ÅŸÂ¯Â≣Ã¢ÅœÅ...	not_cyberbullying	not_cyberbullying
96	ilovethesecret #lawofattraction #quiz #love ...	not_cyberbullying	not_cyberbullying
97	it seems like the only place with action here ...	not_cyberbullying	not_cyberbullying
98	@user brilliant service at your kettering bran...	not_cyberbullying	not_cyberbullying
99	i am thankful for now. #thankful #positive	not_cyberbullying	not_cyberbullying

Fig 4.3.6: Processed tweet with cyberbullying prediction

Here, following categorization, with original sentiment and predicted sentiment is shown in above figure.

5. Testing

5.1 Testing Plan / Strategy

7.1.1 Evaluation

A scoring function is used to evaluate the model. This function will compare the original result with the anticipated result, taking into account both right and wrong tags. The goal of this comparison is to determine the exact accuracy of the proposed model.

Syntax:

```
def score (self, feature, target):

    compare = []

    for i in range (0, len(feature)):

        if feature[i] == target[i]:

            tmp ='correct'

            compare.append (tmp)

        else:

            tmp ='incorrect'

            compare.append(tmp)

    r = Counter(compare)

    accuracy = r['correct']/(r['correct']+r['incorrect'])

    return accuracy
```

Result Obtained:

Syntax:

```
tnb = TweetNBClassifier(df_train)
tnb = tnb.fit ()
predict = tnb.predict(df_test)
score = tnb.score(predict,df_test.sentiment.tolist())
print(score)
```

The evaluation of the model is done by invoking various functions, such a TweetNBClassifier, predict, and score.

Accuracy: Proposed model achieves accuracy of 77 %.

7.1.2 Accuracy Comparison

Accuracy is helpful for determining whether the prediction of the data is accurate or whether one make a note and calculate using the next formula.

$$Accuracy = \frac{TrueNegatives + TruePositive}{TruePositive + FalsePositive + TrueNegative + FalseNegative}$$

Table 5.1: Comparison Table

Algorithm Used	Support Vector Machine	Naive Bayes
Feature Extraction method	TextBlob	Vader sentiment
Accuracy	72 %	77%

The table above provides an illustration of the accuracy that attained with the assistance of the SVM and the Naive Bayes algorithm by utilizing various methods for Feature Extraction.

Findings of proposed model

Due to the time-sensitive nature of the data, proposed model used Gaussian Naive Bayes to train the algorithm how to autonomously retrieve the desired output in response to a predefined set of input variables. Because model used Gaussian Naïve Bayes on an input variable and divide the data into classes, then each class's set of observations will have the same probability distribution.

The model begins by classifying a wide variety of data into subsets, or "classes," before calculating statistical measures for each subset. Then it uses the class values to save the statistics as tuples on a dictionary.

Here, the model first determines whether or not a piece of new information fits into the first class, and then, using conditional probability, determines whether or not the information fits into the second class. This is repeated for each of the classes in the model.

The following formula be used to determine the conditional probability that data belongs to a specific class:

P (class | data) = P (X | class) * P(class)

The Bayes Theorem presented above is different, as have noticed.

In contrast to the nave bayes theorem, division is not used in this computation for the sake of simplification, and the outcome is not strictly dependent on the condition indicates that the model takes the conclusion with the highest probability.

6. Conclusion and Discussion

6.1 Overall Analysis of Internship / Project Viabilities

The victim of cyberbullying experience a variety of mental health problems, including despair, anxiety, rage, fear, concerns related to trusting others, and low self-esteem. Therefore, detection of cyberbullying in the enormous social media network has become immensely vital. In conclusion, we conducted research on the identification of cyberbullying in tweets in order to identify cyberbullying language and actors on the discord Platform. The Naive Bayes algorithm for classification and Vader Sentiment for Feature Extraction with suitable Data Preprocessing Techniques were used in this work, and it was successful in identifying tweets that were engaging in cyberbullying.

The moderation improved, and they able to react more rapidly when it was essential thanks to the automatic detection of signals of cyberbullying. On the other hand, these tweets very well be an indication that the target is being bullied online. This project's primary objective is to offer a system that is able to automatically detect signals of cyberbullying on the social media platform Twitter. This system able to distinguish between various forms of cyberbullying and cover postings made by bullies, victims, and onlookers.

The result of the proposed solution was accomplished by performing a comparative analysis of various machine learning approaches, such as SVM, as well as the Naive bayes methodology, with the Vader sentiment serving as the feature vector rather than the Text blob. Extraction as TextBlob ignores terms that it does not know, and it takes into account words and phrases that it assign polarity to in order to calculate an average score; in contrast, Vader sentiment does not need any training data in order to function properly. It is capable of comprehending the meaning of a text quite well, even if it contains emoticons, slang, conjunctions, capital words, punctuation, and a great deal more besides. It works extremely well on social media text, and the results demonstrate that naive bayes with Vader sentiment performs appropriately in comparison to the other methods.

I conduct, one of my goals is to enhance the accuracy of the classifier by using additional datasets. Due to the expansive nature of social media and the fact that it is not limited to a single language, I also like to implement the proposed method to identify cyberbullying in several languages. Even though tremendous progress has been made in the detection of cyberbullying based on text, there is still much need for improvement in the detection of cyberbullying based on audio or video.

6.2 Problem Encountered and Possible Solutions

Problem Encountered:

Lack of labeled data: One of the main challenges in implementing an automatic cyberbullying detection system is the availability of labeled data. It be difficult to obtain a sufficient amount of labeled data to train a machine learning model to detect cyberbullying effectively.

Class imbalance: The distribution of cyberbullying instances and non-cyberbullying instances be imbalanced, making it difficult for the machine learning algorithm to learn the patterns and accurately classify the instances.

Language variability: The language used in cyberbullying vary widely and include slang, sarcasm, and irony, which make it difficult for machine learning models to accurately detect it.

Context dependency: Cyberbullying be dependent on the context and the relationship between the parties involved, which make it challenging to develop a model that accurately detect it.

Possible Solutions:

Data augmentation: One solution to the lack of labeled data is to use data augmentation techniques, such as synthetic data generation, to increase the size of the dataset.

Sampling techniques: To address the class imbalance issue, sampling techniques such as oversampling or under sampling be used to balance the number of instances in each class.

Text preprocessing: Preprocessing techniques such as stemming, lemmatization, and removal of stop words be used to standardize the text and reduce the variability in the language used in cyberbullying.

Contextual features: Including contextual features, such as the relationship between the parties involved and the context in which the communication occurred, improve the accuracy of the model in detecting cyberbullying.

Ensemble learning: Using multiple machine learning algorithms in an ensemble improve the overall performance of the system in detecting cyberbullying.

6.3 Summary of Internship / Project work

Automatic detection of cyberbullying is a project that aims to use data science and machine learning to identify and flag instances of cyberbullying in online conversations. The project involves several steps, including data collection, preprocessing, feature engineering, model selection, and evaluation.

Data collection involves gathering online conversations and social media posts that are suspected to contain instances of cyberbullying. Preprocessing involves cleaning and transforming the data into a format that is suitable for analysis. Feature engineering involves selecting and creating features that be used to train machine learning models. This includes features such as the use of profanity, negative sentiment, or the frequency of certain words or phrases.

Model selection involves choosing a machine learning algorithm that is well-suited to the task of cyberbullying detection. Common models used for this task include support vector machines (SVMs), decision trees, and neural networks. Evaluation involves testing the accuracy of the chosen model using a validation set of labeled data.

Once a model has been trained and evaluated, it be used to automatically detect instances of cyberbullying in real-time. This be achieved by integrating the model into a chatbot or social media platform, which flag potentially harmful messages and alert moderators or users to take appropriate action. Overall, automatic detection of cyberbullying using data science and machine learning is an important tool for improving online safety and reducing the prevalence of harmful behavior in online communities.

6.4 Limitation and Future Enhancement

Limitations:

Lack of labeled data: One of the primary challenges in developing an effective cyberbullying detection system is the lack of annotated data. It challenges to acquire a large enough dataset that is labeled with accurate information about cyberbullying instances. This led to biased or inaccurate models.

Contextual understanding: Cyberbullying detection models often struggle with contextual understanding. It is difficult for algorithms to differentiate between playful teasing and actual bullying, sarcasm, or satire.

Adaptation to new forms of bullying: Cyberbullying take many forms and be constantly evolving, making it challenging for a detection system to keep up with new trends and identify new types of bullying.

Future Enhancements:

Multimodal data analysis: Cyberbullying involve different forms of data such as text, images, and videos. Developing models that analyze multiple types of data enhance detection accuracy.

Incorporating social network analysis: Cyberbullying often happens in social media platforms, and it be useful to analyze the social network to understand the relationship between the bully and the victim. Incorporating social network analysis techniques enhance the detection of cyberbullying instances.

Active learning: Developing models that learn from their mistakes and incorporate new labeled data enhance the model's performance over time. Active learning is an approach that involves selecting the most informative data points to label and adding them to the training set to improve model accuracy.

7. References

Adnan, 2019. An analytical study of information extraction from unstructured and multidimensional big data. Journal of Big Data , pp. 85-91.

Agarwal, 2015. Research on Data Preprocessing and Categorization Technique for Smartphone Review Analysis. International Journal of Computer Applications , pp. 30-36.

Agrawal, 2018. Deep learning for detecting cyberbullyingacross multiple social media platforms. s.l., Springer, pp. 141-153.

Ajlan, 2018. Deep learning algorithm for cyberbul-lying detection. International Journal of Advanced Computer Scienceand Applications, pp. 1-9.

Al-garadi, 2016. Cybercrimedetection in online communications: The experimental case of cyberbul-lying detection in the twitter network. Computers in Human Behavior, pp. 433-443.

Anon, 2012. Stemming Algorithms: A Comparative Study and their Analysis.. International Journal of Applied Information Systems,, pp. 7-12.

Breiman, 2001. RANDOM FORESTS,, Berkeley, CA: Statistics Department University of California..

Campbell, 2012. Online social networking and the experience of cyber-bullying.. Studies in Health Technology and Informatics, pp. 212-217.

Ghosh, 2017. Toward multimodal cyberbullying detection. s.l., s.n., pp. 2090-2099.

Huang, 2018. Weakly supervised cyberbullying detectionusing co-trained ensembles of embedding models. s.l., IEEE, pp. 479-486.

Jabbar, 2020. Empirical evaluation and study of text stemming algorithms. Springer, pp. 5559-5588.

Karjaluoto, 2015. An-tecedents of social media b2b use in industrial marketing context:customers' view. Journal of Business & Industrial Marketing, pp. 1-8.

Khyani, 2021. An Interpretation of Lemmatization and Stemming in Natural Language Processing. Journal of University of Shanghai for Science and Technology, pp. 350-357.

Koto, 2015. A Comparative Study on Twitter Sentiment Analysis: Which Features are Good?. Passau, Germany, s.n., pp. 453-457.

Peng, 2002. An Introduction to Logistic Regression Analysis and Reporting. The Journal of Educational Research, pp. 3-14.

Peng, 2002. An Introduction to Logistic Regression Analysis and Reporting.. The Journal of Educational Research, Volume 96, pp. 3-14.

Petcharat, 2017. A Corpus-Based Study of English Synonyms. Language Education and Acquisition Research Network Journal, pp. 1-15.

Type your text

Qaiser, 2018. Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents.. International Journal of Computer Applications , pp. 1-8.

Qaiser, 2018. Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents.. International Journal of Computer Applications, Volume 181, pp. 1-8.

Rajasekaran, 2018. Review on automatic text summarization. International Journal of Engineering & Technology, pp. 456-460.

Ramachandra, 2020. Automated cyberbullying detection insocial media using an svm activated stacked convolution lstm network. s.l., s.n., pp. 170-174.

Singh, 2015. VECTOR SPACE MODEL: AN INFORMATION RETRIEVAL. International Journal of Advanced Engineering Research and Studies, pp. 1-3.

Skorkovská, 2012. Application of Lemmatization and Summarization Methods in Topic Identification Module for Large Scale Language Modeling Data Filtering.. Brno, Czech Republic, s.n.

Skorkovská, 2012. Application of Lemmatization and Summarization Methods in Topic Identification Module for Large Scale Language Modeling Data Filtering.. Brno, Czech Republic, s.n.

Talib, 2016. Text Mining: Techniques, Applications and Issues. International Journal of Advanced Computer Science and Applications, pp. 1-5.

Vandebosch, 2014. Cyberbullying on social network sites. anexperimental study into bystanders' behavioural intentions to help thevictim or reinforce the bully. Computers in Human Behavior, pp. 259-271.

Wang, 2020. Multi-modal cy-berbullying detection on social networks. s.l., IEEE, pp. 1-8.

Webster, 1992. Tokenization as the initial phase in NLP. s.l., DBLP, pp. 1-8.

Zhang, 2015. Using Word2Vec to process big text data. s.l., s.n.

Zhang, 2015. Using Word2Vec to process big text data.. San Jose, CA, IEEE International Conference..