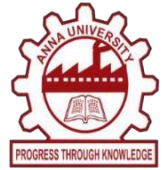




**MACHINE LEARNING BASED
CYBERBULLYING DETECTION
ON SOCIAL MEDIA TEXTS**



A PROJECT REPORT

Submitted by

JENIDA P (731620104025)

KAVISHNI S (731620104028)

SOWMIYA G (731620104051)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K S R INSTITUTE FOR ENGINEERING AND TECHNOLOGY

TIRUCHENGODE – 637 215

An Autonomous Institution

ANNA UNIVERSITY :: CHENNAI 600 025

MAY 2024

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report “**MACHINE LEARNING BASED CYBERBULLYING DETECTION ON SOCIAL MEDIA TEXTS**” is the bonafide work of “**JENIDA P, KAVISHNI S, SOWMIYA G**” who carried out the project work under my supervision.

SIGNATURE

Dr.MALATTHI SIVASUNDARAM,
M.E., Ph.D.,

HEAD OF THE DEPARTMENT

Associate Professor & Head
Department of Computer Science and
Engineering,
K S R Institute For Engineering and
Technology
Tiruchengode – 637215

SIGNATURE

Mr.KARTHIKEYAN C,
M.E.,

SUPERVISOR

Assistant Professor
Department of Computer
Science and Engineering,
K S R Institute For Engineering
and Technology
Tiruchengode – 637215

Submitted for the Project work Viva- Voce held on -----

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We express our deep sense of gratitude to our respected Founder of **Theivathiru Lion Dr. K. S. RANGASAMY, MJF.**, K S R Educational Institutions and chairman of K S R Educational Institution **Mr. R. SRINIVASAN, BBM.**, for their moral support throughout the project.

We wish to express our sincere thanks to respected principal **Dr. M. VENKATESAN, M.E., Ph.D.**, for allowing us to have the extensive use of college facilities to do this project effectively.

We would like to express our sincere gratitude to **Dr. MALATTHI SIVASUNDARAM, M.E., Ph.D.**, Head of the Department, Department of Computer Science and Engineering for her valuable guidance and constant motivation.

We express our extreme gratefulness to our project coordinator **Dr. D. SATHIYA, M.E., Ph.D.**, Associate Professor, Department of Computer Science and Engineering, for providing us kind advice during the development of the project.

We are also grateful to our guide **Mr. C. KARTHIKEYAN, M.E.**, Assistant Professor, Department of Computer Science and Engineering, for the continuous help over the period and creative ideas for this phase of project work.

We wish to extend our sincere thanks to all faculty and staff of our Computer Science and Engineering Department for their valuable suggestions, kinds, co-operation and constant encouragement for successful completion of this project.

ABSTRACT

Cyberbullying has emerged as a pervasive and concerning issue on social media platforms, impacting the mental health and well-being of individuals worldwide. To address this problem, this study proposes a cyberbullying detection system using the (K-SVM) algorithm. Leveraging the power of machine learning, the system aims to automatically identify and flag instances of cyberbullying in social media content. The development of the detection system begins with the collection and labelling of a comprehensive dataset containing examples of cyberbullying and non-cyberbullying posts or comments. After pre-processing the text data by removing irrelevant information, converting text to lowercase, and tokenizing it, meaningful features are extracted using the bag-of-words or TF-IDF techniques. These transformed feature vectors serve as inputs for training the K-SVM classifier, which seeks to find the optimal hyper plane for effectively distinguishing cyberbullying from non-cyberbullying content. The performance of the K-SVM model is evaluated using a separate testing dataset, with metrics such as accuracy, precision, recall, F1-score, and ROC-AUC analysed to assess its effectiveness in identifying cyberbullying instances.

TABLE OF CONTENT

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iv
	LIST OF TABLES	viii
	LIST OF FIGURES	ix
	LIST OF ABBREVIATIONS	x
1.	INTRODUCTION	1
	1.1 MACHINE LEARNING	1
	1.2 CYBERBULLYING DETECTION	1
	1.3 DISTIL BERT	2
	1.4 PRE-TRAINED LANGUAGE MODELS	3
	1.5 TRANSFER LEARNING	4
2	LITERATURE SURVEY	6
3	SYSTEM ANALYSIS	15
	3.1 EXISTING SYSTEM	15
	3.1.1 Drawbacks	17
	3.2 PROPOSED SYSTEM	18
	3.2.1 Advantages	19
	3.3 FEASIBILITY STUDY	20
	3.3.1 Technical Feasibility	20
	3.3.2 Operational Feasibility	21
	3.3.3 Economical Feasibility	22
4	SYSTEM SPECIFICATION	23
	4.1 HARDWARE CONFIGURATION	23

	4.2 SOFTWARE SPECIFICATION	23
5	SOFTWARE DESCRIPTION	24
	5.1 FRONT END: JAVA	24
	5.1.1 Features of Java	24
	5.1.2 Socket overview	25
	5.1.3 Client/Server	25
	5.1.4 Reserved Sockets	26
	5.1.5 Java and the Net	26
	5.1.6 INET Address	26
	5.1.7 Factory Methods	27
	5.1.8 Instance methods	27
	5.1.9 TCP/IP Client Sockets	28
	5.1.10 TCP/IP Server Sockets	29
6	PROJECT DESCRIPTION	31
	6.1 PROBLEM DEFINITION	31
	6.2 MODULE DESCRIPTION	31
	6.2.1 Load Data	31
	6.2.2 Data Pre-Processing	32
	6.2.3 Feature selection	32
	6.2.4 Training and Testing	32
	6.2.5 Evaluation and Performance	32
	6.3 SYSTEM FLOW DIAGRAM	33
	6.4 INPUT DESIGN	33
	6.5 OUTPUT DESIGN	34
7	SYSTEM TESTING AND IMPLEMENTATION	36
	7.1 SYSTEM TESTING	36

	7.2 SYSTEM IMPLEMENTATION	37
8	SYSTEM MAINTENANCE	39
	8.1 TYPES OF MAINTENANCE	40
	8.1 Corrective Maintenance	40
	8.2 Adaptive Maintenance	40
	8.3 Perfective Maintenance	41
	8.4 Preventive Maintenance	41
9	CONCLUSION AND FUTURE WORK	42
10	APPENDICES	43
	APPENDIX 1	43
	APPENDIX 2	49
11	REFERENCES	52

LIST OF TABLES

TABLE NO	TABLE NAME	PAGE NO
4.1	Hardware Requirement	23
4.2	Software Requirement	23

LIST OF FIGURES

FIGURE NO.	CONTENTS	PAGE NO.
1.1	Cyberbullying detection	2
1.2	Distil BERT	3
6.3	System flow diagram	33

LIST OF ABBREVIATIONS

TERM	ABBREVIATIONS
BERT	BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS
CNNs	CONVOLUTIONAL NEUTRAL NETWORKS
DEA	DOLPHIN ECHOLOCATION ALGORITHM
DL	DEEP LEARNING
EDL-TSGSO	ENSEMBLE DEEP LEARNING WITH TOURNAMENT-SELECTED GLOWWORM SWARM OPTIMIZATION
ELSTM	ENSEMBLE LONG SHORT-TERM MEMORY
LSTM	LONG SHORT-TERM MEMORY
ML	MACHINE LEARNING
MNB	MULTINOMIAL NAIVE BAYES
NIC	NATIONAL INFORMATICS CENTER
NLP	NATURAL LANGUAGE PROCESSING
OSNs	ONLINE SOCIAL NETWORKS
RNN	RECURRENT NEURAL NETWORKS
TF-IDF	TERM FREQUENCY – INVERSE DOCUMENT FREQUENCY

CHAPTER 1

INTRODUCTION

1.1 MACHINE LEARNING

Machine Learning (ML) is a cutting-edge field of artificial intelligence that empowers computers to learn from data and improve their performance over time without being explicitly programmed. By mimicking the way humans learn, ML algorithms enable machines to recognize patterns, make decisions, and solve complex problems across various domains. Supervised learning involves training the model on labelled data, where input-output pairs are provided. The model learns to map inputs to corresponding outputs and can then predict outputs for new inputs it has not seen before. This technique is widely used for tasks such as image recognition, natural language processing, and spam detection. Unsupervised learning, on the other hand, deals with unlabelled data, seeking to find patterns, groupings, or hidden structures within the data without explicit guidance. This is particularly useful for tasks like clustering similar data points, dimensionality reduction, and anomaly detection.

1.2 CYBERBULLYING DETECTION

In recent years, the widespread adoption of social media platforms has revolutionized the way people communicate and interact online. While these platforms offer tremendous opportunities for connectivity and expression, they have also given rise to a dark side - cyberbullying. Cyberbullying refers to the use of digital communication tools, such as social media, text messages, or online forums, to harass, intimidate, or

demean individuals. The prevalence of cyberbullying has become a growing concern as it can have severe and lasting effects on the victims, leading to emotional distress, social isolation, and, in some tragic cases, even suicide. Identifying and combating cyberbullying has, therefore, become a critical priority for creating safe and inclusive online spaces. Traditional manual methods for detecting and preventing cyberbullying are often insufficient to address the sheer volume of content generated on social media platforms. Fortunately, advances in Machine Learning (ML) and Natural Language Processing (NLP) have opened new possibilities for automating cyberbullying detection. By leveraging these technologies, we can develop intelligent systems capable of identifying potentially harmful content in real-time, allowing for timely interventions and fostering a healthier online environment.

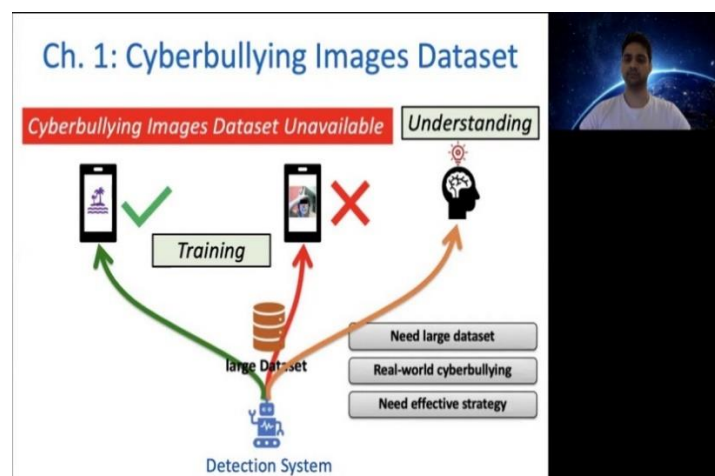


Figure 1.1 Cyberbullying detection

1.3 DISTIL BERT

In the era of modern Natural Language Processing (NLP), the development of powerful language models has revolutionized the way machines understand and process human language. Among these ground-

breaking models, Distil BERT has emerged as a prominent contender, offering remarkable efficiency and performance in various NLP tasks.

Distil BERT is a distilled version of the revolutionary BERT (Bidirectional Encoder Representations from Transformers) model, which was introduced by Google in 2018. BERT's ability to learn context and meaning from both left and right contexts of a word was a significant advancement in NLP. However, its sheer size made it computationally expensive and challenging to deploy in resource-constrained environments. As a result, Distil BERT is significantly smaller and faster, making it more practical for real-world applications without compromising performance.

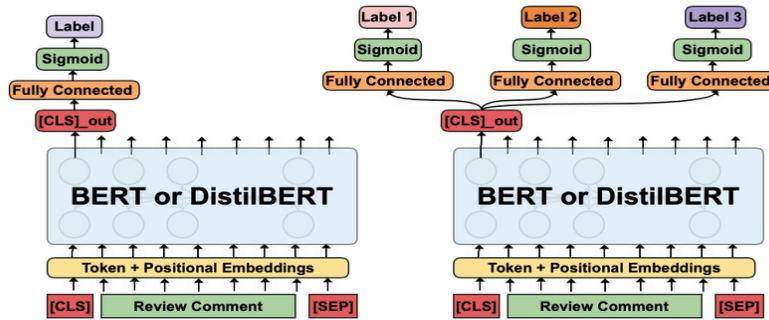


Figure 1.2 Distil BERT

1.4 PRE-TRAINED LANGUAGE MODELS

Pre-trained language models have become the cornerstone of cutting-edge Natural Language Processing (NLP) and have ushered in a new era of language understanding and generation. These models, driven by the power of deep learning and massive amounts of data, possess the remarkable ability to learn intricate patterns and structures from vast corpora of text. By pre-training on diverse and extensive datasets, they

acquire a broad understanding of language that can then be fine-tuned for specific NLP tasks.

The advent of pre-trained language models represents a paradigm shift in NLP, eliminating the need for building task-specific models from scratch. Traditionally, NLP tasks required painstaking feature engineering and domain-specific knowledge, making the development of language-based applications time-consuming and challenging. Pre-trained language models, however, offer a more efficient and effective approach by providing a solid foundation of language understanding that can be leveraged across various tasks.

One of the most ground-breaking pre-trained language models is BERT (Bidirectional Encoder Representations from Transformers), introduced by Google in 2018. This bidirectional understanding revolutionized the way models comprehend language, leading to state-of-the-art performance on a wide range of NLP tasks. BERT's innovation lies in its bidirectional attention mechanism, enabling it to capture contextual information from both left and right contexts of a word. benchmarks.

1.5 TRANSFER LEARNING

Transfer learning, a groundbreaking concept in the field of machine learning, has revolutionized the way models are developed, enabling significant advancements in various domains. Traditional machine learning approaches required models to be trained from scratch for each specific task, which was time-consuming and computationally expensive. Transfer learning, however, provides a more efficient and effective solution by allowing knowledge learned from one task to be leveraged and transferred to related tasks.

At the core of transfer learning lies the idea that models can acquire general knowledge from a large dataset and then fine-tune that knowledge for specific tasks with smaller datasets. This approach takes advantage of the fact that many tasks share underlying patterns and features, making it possible to transfer knowledge from one domain to another. One of the most remarkable applications of transfer learning can be observed in the domain of computer vision.

Models like VGG, ResNet, and Inception, pre-trained on massive image datasets like ImageNet, have demonstrated the ability to extract general visual features. Similarly, transfer learning has made significant strides in natural language processing. Pre-trained language models like BERT, GPT, and ROBERTa, trained on vast amounts of text data, have acquired a deep understanding of language structures and semantics. By fine-tuning these models on specific NLP tasks like sentiment analysis, question-answering, and text classification, they can achieve state-of-the-art performance with minimal labelled data.

CHAPTER 2

LITERATURE SURVEY

Ensemble Learning With Tournament Selected Glowworm Swarm

Optimization algorithm for cyberbullying detection on social media

Ravuri Daniel et.al.[8] (2023) proposed in this paper that Online social networks (OSNs) are vital for fostering social connections, but they also unfortunately, contribute to antisocial behaviors such as trolling, cyberbullying, and hate speech. Cyberbullying, particularly harmful to children and women, has led to severe physical and mental distress and, tragically, even suicides. Traditional methods of detecting cyberbullying, like relying on user reports, are often insufficient. To address this, researchers are turning to deep learning (DL) and machine learning (ML) techniques to automatically recognize and highlight possible cases of cyberbullying, as well as recognize behavior patterns indicative of such behavior. This study focuses on developing an algorithm called Ensemble Deep Learning with Tournament-Selected Glowworm Swarm Optimization (EDL-TSGSO) for cyberbullying detection and classification using Twitter data. The goal is to leverage natural language processing (NLP) and ensemble learning to analyze social media data. The EDL-TSGSO algorithm preprocesses tweets and utilizes Glove word embedding, along with ensemble Long Short-Term Memory (ELSTM) with Adaboost (AB) model, to effectively detect and classify cyberbullying. By integrating the predictions of LSTM and Adaboost models, the ensemble ELSTM-AB classifier enhances overall classification performance. Additionally, the TSGSO algorithm is

employed as a hyperparameter optimizer to improve cyberbullying detection.

A multi-stage machine learning and fuzzy approach to cyber-hate detection

Liba Kestsbain et.al.[5] (2023) has argued in this study that although social media's rise has revolutionized international communication and information exchange, it has also raised worries about the growth of cyber-hatred. Many approaches, such as machine learning and deep learning methods including recurrent neural networks, logistic regression, convolutional neural networks, and naive bayes, have been put forth by researchers. These techniques use mathematical algorithms to differentiate between several classifications. However, a more sophisticated approach—one that takes into account how people really understand online messages—is required for successful categorization when working with sentiment-driven data. This work used four datasets connected to online hatred using Multinomial Naive Bayes and Logistic Regression classifiers, based on a survey of relevant literature. Particle Swarm Optimization and other bio-inspired optimization methods can improve classifier performance.

Approaches to automated detection of cyberbullying: A survey

Semiu Salawu et.al.[9] (2020), here the Automated methods of cyberbullying detection are becoming more popular, as this framework suggests. These methods use machine learning and natural language processing techniques to identify the characteristics of cyberbullying interactions and automatically flag instances of cyberbullying by

examining text data. Using data from the bibliographic databases IEEE Xplore, ACM, and Scopus, this report provides a comprehensive analysis of published research on methods for detecting cyberbullying. We divide the current methods into four primary groups based on our extensive analysis of the literature: supervised learning, lexicon-based, rule-based, and mixed-initiative techniques. To construct predictive models for cyberbullying detection, supervised learning techniques often use classifiers like SVM and Naïve Bayes. Cyberbullying is detected by lexicon-based systems by looking for certain terms in word lists.

DEA-RNN: A hybrid deep learning approach for cyberbullying detection in twitter social media platform

Belal Abdullah Hezam Murshed et.al.[1] (2022) has proposed in this system that this methodology has shown that the ubiquity of cyberbullying on social media platforms has sparked worries about user safety. In order to detect cyberbullying on the Twitter network, this study presents DEA-RNN, a hybrid deep learning model. To optimize the Dolphin Echolocation Algorithm (DEA) and minimize training time, DEA-RNN blends Elman-type Recurrent Neural Networks (RNN) with an enhanced DEA. A dataset of 10,000 tweets was used for a thorough assessment of DEA-RNN, with its performance compared to that of cutting-edge algorithms such as Random Forests (RF), RNN, SVM, Multinomial Naive Bayes (MNB), and Bi-directional Long Short-Term Memory (Bi-LSTM). According to the results, DEA-RNN routinely beats other methods for identifying cyberbullying on Twitter. Situation 3 shows very impressive performance, with an average accuracy of 90.45%.

Cyberbullying detection and severity determination model

Mohammed Hussein Obain et.al.[6] (2023) has suggested in the research that some teenagers participate in cyberbullying, a negative online activity that targets other people. Many teenagers are not aware that cyberbullying can have serious consequences, such as despair, self-harm, and even suicide. It is imperative to combat cyberbullying due to its substantial influence on mental health. The purpose of this research was to create a technique that would use fuzzy logic and deep learning algorithms to evaluate the severity of cyberbullying. Cyberbullying incidents were found by processing and analyzing 47,733 comments from Kaggle's Twitter data. Using Keras, the comments were embedded, and a four-layer long short-term memory network was fed the data for classification. Fuzzy logic was then applied to assess the seriousness of the remarks. Results from experiments show that the suggested framework provides a useful remedy for cyberbullying detection, achieving accuracy, F1- score, and recall values of 93.67%, 93.64%, and 93.62%, respectively.

Offensive Language Detection in spanish social media: Testing from bag-of-words to transformers models

Jose Maria Molero et.al.[3] (2023) proposed the challenge of identifying offensive language in social media, particularly in languages rich in popular sayings, colloquial expressions, and idioms, such as Mexican Spanish. The authors define the main linguistic features of aggressive, offensive, and vulgar language in social networks to establish linguistic-based criteria to facilitate the identification of abusive language. They compile and analyze a Mexican Spanish Twitter corpus

to define linguistic criteria for determining whether a message is offensive. The study highlights the importance of linguistic-based criteria in identifying offensive language in social media and provides a valuable resource for researchers working on detecting abusive language in Spanish. The authors hope that their work will contribute to the development of more sophisticated and effective methods for detecting and mitigating offensive language in social media.

Predicting Cyberbullying on Social Media in the Big Data Era Using Machine Learning Algorithms: Review of Literature and Open Challenges

Murtaza et.al.[7] (2019) proposed the Prior to the innovation of information communication technologies (ICT), social interactions evolved within small cultural boundaries such as geo spatial locations. The recent developments of communication technologies have considerably transcended the temporal and spatial limitations of traditional communications. These social technologies have created a revolution in user-generated information, online human networks, and rich human behavior-related data. However, the misuse of social technologies such as social media (SM) platforms, has introduced a new form of aggression and violence that occurs exclusively online. A new means of demonstrating aggressive behavior in SM websites are highlighted in this paper. The motivations for the construction of prediction models to fight aggressive behavior in SM are also outlined. We comprehensively review cyberbullying prediction models and identify the main issues related to the construction of cyberbullying prediction models in SM. This paper provides insights on the overall process for cyberbullying detection and most importantly overviews the

methodology. Though data collection and feature engineering process has been elaborated, yet most of the emphasis is on feature selection algorithms and then using various machine learning algorithms for prediction of cyberbullying behaviors. Finally, the issues and challenges have been highlighted as well, which present new research directions for researchers to explore.

When the timeline meets the pipeline: A survey on automated cyberbullying detection

Fatma Elsafoury et.al.[2] (2023) proposed that the Web 2.0 helped user-generated platforms to spread widely. Unfortunately, it also allowed for cyberbullying to spread. Cyberbullying has negative effects that could lead to cases of depression and low self-esteem. It has become crucial to develop tools for automated cyberbullying detection. The research on developing these tools has been growing over the last decade, especially with the recent advances in machine learning and natural language processing. Given the large body of work on this topic, it is vital to critically review the literature on cyberbullying within the context of these latest advances. In this paper, we survey the automated detection of cyberbullying. Our survey sheds light on some challenges and limitations for the field. The challenges range from defining cyberbullying, data collection, and feature representation to model selection, training, and evaluation. We also provide some suggestions for improving the task of cyberbullying detection. In addition to the survey, we propose to improve the task of cyberbullying detection by addressing some of the raised limitations: 1) Using recent contextual language models like BERT for the detection of cyberbullying; 2) Using slang-

based word embeddings to generate better representations of the cyberbullying-related datasets. Our results show that BERT outperforms state-of-the-art cyberbullying detection models and deep learning models. The results also show that deep learning models initialized with slang-based word embeddings outperform deep learning models initialized with traditional word embeddings.

Explainable cyberbullying detection in hinglish: A generative approach

Krishanu Maity et.al.[4] (2023) proposed that the escalating prevalence of online cyberbullying and trolling across various social media platforms has become a pressing concern. Extensive research demonstrates the detrimental impact of cyberbullying on the mental well-being of its victims. Given the sheer volume of online content, manual identification of cyberbullying instances proves unfeasible, necessitating the development of automated cyberbullying detection methods. This challenge has attracted considerable attention within the natural language processing (NLP) community, owing to advancements in machine learning techniques. However, most of the methods fail to provide reasoning for their decisions which warrants the use of interpretable models that can explain the model’s output in real-time. Interpretable models rather than black-box models with high performance are becoming popular adhering the “right to explanations” laws. Motivated by this, we create a cyberbullying corpus BullyExplain in code-mixed language, where a post has been annotated with four labels, i.e., bully, sentiment, target, and rationales (explainability). Current work addresses the task of explainable cyberbully detection and

proposes a unified generative framework, BullyGen by redefining this multitask problem as a text-to-text generation task. Our framework is capable of not only detecting whether the text is a cyberbully or not but also provides reasoning by predicting rationale, target group and sentiment of the text. Experimental results illustrate the efficacy of our proposed model by outperforming the state-of-the-art and several baselines by a significant margin and conclude that text-to-text generation model could be a good alternative for multitask classification problems.

Cyberbullying Detection in Social Networks: A Comparison Between Machine Learning and Transfer Learning Approaches

Teoh Hwai Teng et.al.[10] (2023) proposed that the Information and Communication Technologies fueled social networking and facilitated communication. The user-dependent mechanisms like reporting, blocking, and removing bullying posts online is manual and ineffective. Bag-of-words text representation without metadata limited cyberbullying post text classification. This research developed an automatic system for cyberbullying detection with two approaches: Conventional Machine Learning and Transfer Learning. Textual, sentiment and emotional, static and contextual word embeddings, psycholinguistics, term lists, and toxicity features were used in the conventional Machine Learning approach. This study was the first to use toxicity features to detect cyberbullying. This study is also the first to use the latest psycholinguistics features from the Linguistic Inquiry and Word (LIWC) 2022 tool, as well as Empath's lexicon, to detect cyberbullying. The contextual embeddings of ggeluBert, tnBert, and

DistilBert have alike performance, however DistilBert embeddings were elected for higher F-measure. Textual features, DistilBert embeddings, and toxicity features that struck new benchmark were the top three unique features when fed individually. The model's performance was boosted to F-measure of 64.8% after feeding with a combination of textual, sentiment, Distil Bert embeddings, psycholinguistics, and toxicity features to the Logistic Regression model that outperforms Linear SVC with faster training time and efficient handling of high-dimensionality features. Transfer Learning approach was by fine-tuning optimized version Pre-trained Language Models namely, DistilBert, Distil RoBerta, and Electra-small which were found to have speedier training computation than their base form. The fine-tuned DistilBert resulted with the highest F-measure of 72.42%, surpassing CML. Our research concluded that Transfer Learning was the best for uplifted performance and lesser effort as feature engineering and resampling was omitted.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

As social media users keep on increasing it has attracted the attention of researchers in examining a novel kind of creative language utilized over the Internet to best search the depth of communication and human thoughts. One most popular social media is Twitter, a micro-blogging site that permits users to write up to 280 text characters simply called tweets. Developments in Twitter have changed the way individuals share their views and feelings with a large audience because of its easy accessibility and free format messages. Twitter was a real-time information platform that collects the global opinions of the public and Twitter has been considered an outstanding channel to examine peoples' opinions and social interactions. Cyberbullying refers to the use of electronic communication, such as social media platforms, to harass, intimidate, or harm others. On Twitter, this could manifest in the form of abusive tweets, hate speech, or targeted harassment directed at specific individuals or groups.

Certainly, the students show symptoms of anxiety and depression, internalizing problems, and negative social relationships, with a risk of suicidal ideas as a function of the frequency of aggressions. Given the significance of cyberbullying and bullying in society, many researchers have examined what can act as protective factors or risks in the involvement of phenomena, addressing the significance of implementing an ecological structure. Cyberbullying through Twitter has gained attention in some years as its leads to several tragic, high-profile

suicides. A conventional system was implemented for managing the problem of cyberbullying in Social Media platforms, with companies including guidelines that their users should follow along with using editors to check manually for bullying behaviour. Moreover, the significant growth in cyberbullying cases has emphasized the danger of cyberbullying, predominantly among adolescents and children, who can be juvenile and inconsiderate.

Adolescents consider bullying as a serious problem without knowing how to handle social problems; this made them share their feelings on social networking sites in a way that could hurt others. Many researchers have exposed that bullies undergo psychological states, which leads them to bully and inflict suffering on other people. Therefore, cyberbullying was the same epidemic, and can result in a violent society, predominantly considering hightech university and school students. Thus, most of the global initiatives were modelled to tackle the issue of cyberbullying. Detection of cyberbullying in social networking sites is highly essential and must be paid higher attention to so that society and children were protected from all those threats.

Cyberbullying is hot a research topic among research communities aimed at deducting, controlling, and reducing cyberbullying on social networking sites. One direction in this field was to find the intention of users to post aggressive content by examining offensive language related to different features, such as the unique content and structure, and the writing style of the users. Another direction of cyberbullying research was to identify text content utilizing ML for offensive language classification and detection. This study concentrates on the design and

development of ensemble deep learning with tournament-selected glowworm swarm optimization (EDL-TSGSO) algorithm for cyberbullying detection and classification on Twitter data. This EDL-TSGSO technique preprocesses the raw tweets and then employs the Glove word embedding technique. In addition, the presented EDL-TSGSO technique utilizes ensemble long short-term memory with AdaBoost (ELSTM-AB) model for effective cyberbullying detection and classification. To further improve the cyberbullying detection performance of the EDL-TSGSO algorithm, the TSGSO algorithm is applied as a hyperparameter optimizer. The experimental validation of the EDL-TSGSO algorithm on the Twitter dataset demonstrates its promising performance over other existing systems in terms of different measures.

3.1.1 Drawbacks

- ML models are only as good as the data they are trained on. If the training data contains biases or is not representative of the diverse ways cyberbullying can manifest, the model might perform poorly on real-world scenarios or unintentionally reinforce existing biases.
- Cyberbullying detection models may produce false positives (flagging harmless content as cyberbullying) and false negatives (failing to detect actual cyberbullying). Achieving a balance between minimizing false positives and false negatives can be challenging.

- Social media platforms constantly evolve, and cyberbullying tactics change accordingly. An ML model trained on historical data might struggle to detect newer forms of cyberbullying or adapt to changing language trends.
- Cyberbullying instances might be relatively rare compared to non-cyberbullying instances, leading to class imbalance. The model may become biased towards the majority class, resulting in lower sensitivity to cyberbullying detection.

3.2 PROPOSED SYSTEM

The proposed system aims to develop an efficient and accurate cyberbullying detection solution for social media platforms. Leveraging the power of machine learning, the system will employ the (K-SVM) algorithm to automatically identify instances of cyberbullying in r social media content. The process will begin with the collection and labeling of a diverse dataset containing either cyberbullying and non-cyberbullying posts or comments. Preprocessing techniques, including text cleaning, lowercasing, and tokenization, will be applied to transform the raw text data into a suitable format for feature extraction. The bag-of-words or TF-IDF techniques will then be employed to extract meaningful features from the pre-processed text data.

These features will serve as inputs for training the K-SVM classifier, which will learn to distinguish between cyberbullying and non-cyberbullying content by finding an optimal hyperplane in the feature space. The system's performance will be rigorously evaluated using various metrics, and fine-tuning will be performed to optimize its

efficiency. Once trained and evaluated, the K-SVM -based cyberbullying detection system will be deployed to operate in on social media platforms, providing timely alerts and support to users facing potential cyberbullying incidents. By ensuring continuous monitoring and updating, the proposed system aims to adapt to evolving cyberbullying patterns, fostering a safer and more respectful online environment for all users.

3.2.1 Advantages

- The system utilizes the K-SVM algorithm to detect cyberbullying in, allowing for swift identification and intervention in potential cyberbullying incidents, thereby promoting a safer online environment.
- Leveraging the power of K-SVM and feature extraction techniques, the system achieves high accuracy in distinguishing between cyberbullying and non-cyberbullying content, reducing false positives and negatives.
- The proposed system's machine learning-based approach can easily scale to handle large volumes of social media data, making it suitable for deployment on various social media platforms with diverse user bases.
- Regular monitoring and model updates enable the system to adapt to emerging cyberbullying behaviours, ensuring its effectiveness in combating ever-changing online threats.

3.3 FEASIBILITY STUDY

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operation Feasibility
- Economical Feasibility

3.3.1 Technical feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipments have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at DB2 Database. Thus it provides an easy access to the users. The

database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified.

Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hardware requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

3.3.2 Operation feasibility

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

- Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question

of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

3.3.3 Economic feasibility

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

CHAPTER 4

SYSTEM SPECIFICATION

4.1 HARDWARE REQUIREMENTS

Table 4.1 Hardware Requirements

CPU type	Intel core i3 processor
Clock speed	3.0 GHz
RAM size	8 GB
Hard disk capacity	500 GB
Keyboard type	Internet Keyboard
CD-drive type	52xmax

4.2 SOFTWARE REQUIREMENTS

Table 4.2 Software Requirements

Operating System	Windows 11
Front End	JAVA

CHAPTER 5

SOFTWARE DESCRIPTION

5.1 FRONT END: JAVA

The software requirement specification is created at the end of the analysis task. The function and performance allocated to software as part of system engineering are developed by establishing a complete information report as functional representation, a representation of system behavior, an indication of performance requirements and design constraints, appropriate validation criteria.

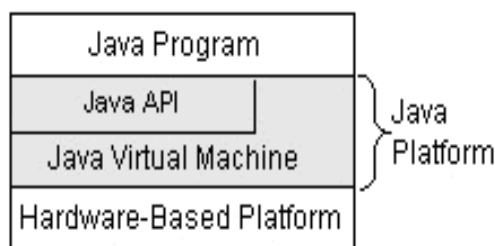
5.1.1 Features of java:

Java platform has two components:

- The Java Virtual Machine (Java VM)
- The Java Application Programming Interface (Java API)

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries (*packages*) of related components.

The following figure depicts a Java program, such as an application or applet, that's running on the Java platform. As the figure shows, the Java API and Virtual Machine insulates the Java program from hardware dependencies.



As a platform-independent environment, Java can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring Java's performance close to that of native code without threatening portability.

5.1.2 Socket overview:

A network socket is a lot like an electrical socket. Various plugs around the network have a standard way of delivering their payload. Anything that understands the standard protocol can “plug in” to the socket and communicate.

Internet protocol (IP) is a low-level routing protocol that breaks data into small packets and sends them to an address across a network, which does not guarantee to deliver said packets to the destination.

Transmission Control Protocol (TCP) is a higher-level protocol that manages to reliably transmit data. A third protocol, User Datagram Protocol (UDP), sits next to TCP and can be used directly to support fast, connectionless, unreliable transport of packets.

5.1.3 Client/server:

A server is anything that has some resource that can be shared. There are compute servers, which provide computing power; print servers, which manage a collection of printers; disk servers, which provide networked disk space; and web servers, which store web pages. A client is simply any other entity that wants to gain access to a particular server.

A server process is said to “listen” to a port until a client connects to it. A server is allowed to accept multiple clients connected to the same

port number, although each session is unique. To manage multiple client connections, a server process must be multithreaded or have some other means of multiplexing the simultaneous I/O.

5.1.4 Reserved sockets:

Once connected, a higher-level protocol ensues, which is dependent on which port user are using. TCP/IP reserves the lower, 1,024 ports for specific protocols. Port number 21 is for FTP, 23 is for Telnet, 25 is for e-mail, 79 is for finger, 80 is for HTTP, 119 is for Netnews-and the list goes on. It is up to each protocol to determine how a client should interact with the port.

5.1.5 Java and the net:

Java supports TCP/IP both by extending the already established stream I/O interface. Java supports both the TCP and UDP protocol families. TCP is used for reliable stream-based I/O across the network. UDP supports a simpler, hence faster, point-to-point datagram-oriented model.

5.1.6 INET address:

The Inet Address class is used to encapsulate both the numerical IP address and the domain name for that address. User interacts with this class by using the name of an IP host, which is more convenient and understandable than its IP address. The Inet Address class hides the number inside. As of Java 2, version 1.4, Inet Address can handle both IPv4 and IPv6 addresses.

5.1.7 Factory methods:

The Inet Address class has no visible constructors. To create an InetAddress object, user use one of the available factory methods. Factory methods are merely a convention whereby static methods in a class return an instance of that class. This is done in lieu of overloading a constructor with various parameter lists when having unique method names makes the results much clearer.

Three commonly used InetAddress factory methods are:

- Static InetAddressgetLocalHost () throws
UnknownHostException
- Static InetAddressgetByName (String hostName)
throwsUnknownHostException
- Static InetAddress [] getAllByName (String hostName) throws
UnknownHostException

5.1.8 Instance methods:

The Inet Address class also has several other methods, which can be used on the objects returned by the methods just discussed. Here are some of the most commonly used.

- Boolean equals (Object other) -Returns true if this object has the same Internet address as other.
- byte [] get Address ()-Returns a byte array that represents the object's Internet address in network byte order
- String getHostAddress () - Returns a string that represents the host address associated with the InetAddress object.

- `String get Hostname ()` - Returns a string that represents the host name associated with the `InetAddress` object.
- `boolean isMulticastAddress ()` - Returns true if this Internet address is a multicast address. Otherwise, it returns false.
- `String toString ()` - Returns a string that lists the host name and the IP address for convenience.

5.1.9 TCP/IP client sockets :

TCP/IP sockets are used to implement reliable, bidirectional, persistent, point-to-point and stream-based connections between hosts on the Internet. A socket can be used to connect Java's I/O system to other programs that may reside either on the local machine or on any other machine on the Internet.

The creation of a `Socket` object implicitly establishes a connection between the client and server. There are no methods or constructors that explicitly expose the details of establishing that connection. Here are two constructors used to create client sockets

`Socket (String hostName, int port)` - Creates a socket connecting the local host to the named host and port; can throw an `UnknownHostException` or an `IOException`.

`Socket (InetAddress ipAddress, int port)` - Creates a socket using a preexisting `InetAddress` object and a port; can throw an `IOException`.

A socket can be examined at any time for the address and port information associated with it, by use of the following methods:

- `InetAddress getAddress ()` - Returns the `InetAddress` associated with the `Socket` object.

- `IntgetPort ()` - Returns the remote port to which this Socket object is connected.
- `IntgetLocalPort ()` - Returns the local port to which this Socket object is connected.

Once the Socket object has been created, it can also be examined to gain access to the input and output streams associated with it. Each of these methods can throw an IO Exception if the sockets have been invalidated by a loss of connection on the Net.

- `Input Streamget Input Stream ()` - Returns the `InputStream` associated with the invoking socket.
- `Output Streamget Output Stream ()` - Returns the `OutputStream` associated with the invoking socket.

5.1.10 TCP/IP server sockets:

Java has a different socket class that must be used for creating server applications. The `ServerSocket` class is used to create servers that listen for either local or remote client programs to connect to them on published ports. `ServerSockets` are quite different from normal `Sockets`.

When the user create a `ServerSocket`, it will register itself with the system as having an interest in client connections.

- `ServerSocket(int port)` - Creates server socket on the specified port with a queue length of 50.
- `Serversocket(int port, int maxQueue)` - Creates a server socket on the specified port with a maximum queue length of `maxQueue`.

- `ServerSocket(int port, int maxQueue, InetAddress localAddress)`-Creates a server socket on the specified port with a maximum queue length of `maxQueue`. On a multihomed host, `localAddress` specifies the IP address to which this socket binds.
- `ServerSocket` has a method called `accept()` - which is a blocking call that will wait for a client to initiate communications, and then return with a normal `Socket` that is then used for communication with the client.

5.1.11 Url:

The Web is a loose collection of higher-level protocols and file formats, all unified in a web browser. One of the most important aspects of the Web is that Tim Berners-Lee devised a saleable way to locate all of the resources of the Net. The Uniform Resource Locator (URL) is used to name anything and everything reliably.

The URL provides a reasonably intelligible form to uniquely identify or address information on the Internet. URLs are ubiquitous; every browser uses them to identify information on the Web.

CHAPTER 6

PROJECT DESCRIPTION

6.1 PROBLEM DEFINITION

The problem at hand is to develop an effective and reliable cyberbullying detection system for social media platforms using Machine Learning (ML) techniques. With the rapid growth of social media usage, cyberbullying has become a pressing concern, causing emotional distress, social isolation, and even potential harm to victims. Manual monitoring of user-generated content is impractical due to the sheer volume of data. Therefore, an ML-based solution is sought to automatically identify instances of cyberbullying, allowing for timely interventions and fostering a safer online environment. The primary objective is to design a model capable of accurately classifying social media posts and comments as cyberbullying or non-cyberbullying, while addressing challenges such as data bias, false positives, false negatives, and the evolution of cyberbullying tactics and language trends. Furthermore, ensuring the system complies with privacy regulations and user acceptance is crucial for its successful deployment and positive impact on the well-being of social media users.

6.2 MODULE DESCRIPTION

6.2.1 Load data

This module is responsible for loading the labelled dataset containing social media posts or comments for training and testing the cyberbullying detection system. It reads the dataset from a file or

database, extracting the text data and corresponding labels (cyberbullying or non-cyberbullying).

6.2.2 Data pre-processing

This module is designed to pre-process the raw text data to make it suitable for feature extraction and K-SVM classification. Clean the text data by removing special characters, URLs, and other irrelevant information. Convert the text to lowercase to ensure case insensitivity. Tokenize the text into individual words or tokens. Apply stemming or lemmatization to reduce words to their root form (optional).

6.2.3 Feature selection

This module performs feature extraction from the pre-processed text data, converting it into numerical feature vectors that the K-SVM can process. Utilize techniques like bag-of-words or TF-IDF to represent the text data as numerical vectors. Create feature matrices containing the transformed data, ready for training the K-SVM model.

6.2.4 Training and testing

This module is responsible for training the K-SVM classifier on the pre-processed and feature-selected data. Split the dataset into training and testing sets. Use the training set to train the K-SVM classifier with appropriate hyper parameters and kernel settings. This module assesses the performance of the trained K-SVM classifier on unseen data. Use the testing set to evaluate the K-SVM classifier's performance in detecting cyberbullying instances. Calculate accuracy, precision, recall, F1-score, and ROC-AUC to evaluate the classifier's effectiveness.

6.2.5 Evaluation and performance

This module analyses the results obtained from the testing module to evaluate the cyberbullying detection system's overall performance. Display the evaluation metrics and performance measures to provide insights into the classifier's accuracy and robustness. Identify potential areas for improvement or fine-tuning of the system.

6.3 SYSTEM FLOW DIAGRAM

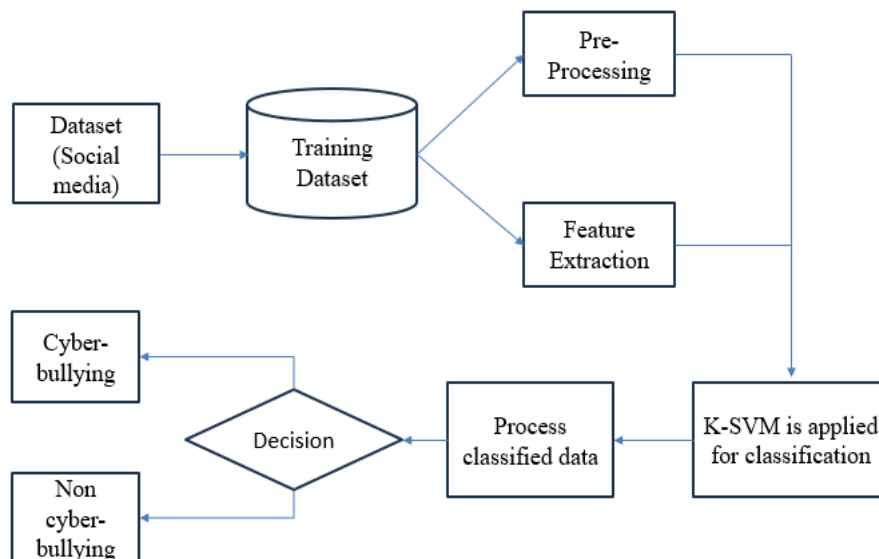


Figure 6.1 System flow diagram

6.4 INPUT DESIGN

Input design for the proposed cyberbullying detection system involves specifying how users interact with the system to provide input data for analysis. In this case, the primary input would be social media posts or comments that need to be assessed for potential cyberbullying. Here's a basic outline of the input design:

- Users will input text data in the form of social media posts, comments, or messages into the cyberbullying detection system. This can be achieved through a user interface (UI) where users can type or paste the text they want to analyse.
- Alternatively, the system can integrate with social media platforms (e.g., Twitter, Facebook) to access and analyse users' posts and comments directly from their profiles. This integration requires users to grant appropriate permissions.
- For continuous monitoring, the system can allow users to enable analysis of their social media activity. This feature can be optional, and users can toggle it on or off.
- In addition to analysis, the system can provide an option for batch processing. Users can upload a file containing multiple social media posts or comments to be analysed collectively.
- To ensure privacy and security, the system may require users to authenticate themselves before accessing their social media data for analysis.

6.5 OUTPUT DESIGN

Output design for the proposed cyberbullying detection system involves presenting the results and feedback to users in a clear and understandable manner. Here's an outline of the output design:

- The main output of the system will be the cyberbullying detection results for each input text. It will categorize the input as either "Cyberbullying" or "Non-Cyberbullying," based on the analysis performed by the K-SVM algorithm.

- Optionally, the system can provide a confidence level or severity score along with the detection results. This score indicates the system's level of certainty about its classification, allowing users to gauge the reliability of the result.
- If the system is integrated with social media platforms and set for, users may receive immediate notifications if cyberbullying is detected in their recent posts or comments.
- For batch processing, the system can present a summary of the cyberbullying detection results for all the input texts collectively. This summary might include the total number of cyberbullying instances and non-cyberbullying instances.
- The output design should employ a user-friendly interface with clear and concise labels, making it easy for users to understand the results without confusion.
- Depending on the complexity of the data and the user's preferences, the system can use visualizations, such as charts or graphs, to represent the results more intuitively.
- If any errors occur during the analysis, the system should provide informative error messages and guidance on how to resolve the issues.

CHAPTER 7

SYSTEM TESTING AND IMPLEMENTATION

7.1 SYSTEM TESTING

System testing for the proposed cyberbullying detection system is a crucial step to ensure its functionality, accuracy, and performance before deployment. It involves evaluating the system's behaviour under various scenarios and verifying that it meets the desired requirements. Here's an outline of the system testing process:

1. Unit Testing:

- Test individual components of the system in isolation to ensure they function correctly.
- Verify the pre-processing, feature extraction, and K-SVM classifier modules independently.
- Use mock data and edge cases to test different functionalities.

2. Integration Testing:

- Test the integration of different modules to ensure seamless communication and data flow between them.
- Validate that data is passed correctly from Pre-processing to feature extraction and then to the K-SVM classifier.

3. Functional Testing:

- Verify that the system correctly identifies cyberbullying content and classifies it accurately as "Cyberbullying" or "Non-Cyberbullying."

- Test the detection functionality, if applicable, to ensure timely notifications are generated for new social media posts or comments.

4. Performance Testing:

- Evaluate the system's speed and responsiveness under different workloads.
- Measure the time taken for cyberbullying analysis on various input sizes.
- Assess resource utilization and ensure the system can handle concurrent requests.

5. Accuracy and Validation:

- Use a representative and diverse dataset with known ground truth to validate the system's accuracy.
- Compare the system's detected cyberbullying instances against the ground truth to calculate metrics such as precision, recall, F1-score, and ROC-AUC.

7.2 SYSTEM IMPLEMENTATION

System implementation for the proposed cyberbullying detection system involves the actual development and deployment of the software. Here's a general outline of the implementation process:

1. Software Development:

- Set up the development environment with the required programming languages, libraries, and frameworks, such as Python, scikit-learn, and NLTK (Natural Language Toolkit).
- Develop the pre-processing module to clean, tokenize, and pre-process the input text data.

- Implement feature extraction techniques, such as bag-of-words or TF-IDF, to convert the pre-processed text data into numerical feature vectors.
- Develop the K-SVM classifier module and train it using the labeled dataset of cyberbullying and non-cyberbullying examples.

2. User Interface (UI) Design:

- Design a user-friendly and intuitive interface for users to input text or upload batch data for analysis.
- Ensure that the UI is responsive and accessible across different devices and browsers.

3. Testing:

- Conduct thorough testing of each component and module to verify their correctness and functionality.
- Perform unit testing, integration testing, functional testing, and other types of testing as described in the "System Testing" section.

CHAPTER 8

SYSTEM MAINTENANCE

The objectives of this maintenance work are to make sure that the system gets into work all time without any bug. Provision must be for environmental changes which may affect the computer or software system. This is called the maintenance of the system. Nowadays there is the rapid change in the software world. Due to this rapid change, the system should be capable of adapting these changes. In this project the process can be added without affecting other parts of the system. Maintenance plays a vital role. The system is liable to accept any modification after its implementation. This system has been designed to favor all new changes. Doing this will not affect the system's performance or its accuracy. Maintenance is necessary to eliminate errors in the system during its working life and to tune the system to any variations in its working environment. It has been seen that there are always some errors found in the system that must be noted and corrected. It also means the review of the system from time to time.

The review of the system is done for:

- Knowing the full capabilities of the system.
- Knowing the required changes or the additional requirements.
- Studying the performance.

8.1 TYPES OF MAINTENANCE

- Corrective maintenance
- Adaptive maintenance
- Perfective maintenance
- Preventive maintenance

8.1.1 Corrective maintenance

Changes made to a system to repair flaws in its design coding or implementation. The design of the software will be changed. The corrective maintenance is applied to correct the errors that occur during that operation time. The user may enter invalid file type while submitting the information in the particular field, then the corrective maintenance will display the error message to the user in order to rectify the error.

Maintenance is a major income source. Nevertheless, even today many organizations assign maintenance to unsupervised beginners, and less competent programmers.

The user's problems are often caused by the individuals who developed the product, not the maintainer. The code itself may be badly written maintenance is despised by many software developers unless good maintenance service is provided, the client will take future development business elsewhere. Maintenance is the most important phase of software production, the most difficult and most thankless

8.1.2 Adaptive maintenance:

It means changes made to system to evolve its functionalities to change business needs or technologies. If any modification in the

modules the software will adopt those modifications. If the user changes the server then the project will adapt those changes. The modification server work as the existing is performed.

8.1.3 Perfective maintenance:

Perfective maintenance means made to a system to add new features or improve performance. The perfective maintenance is done to take some perfect measures to maintain the special features. It means enhancing the performance or modifying the programs to respond to the users needs or changing needs. This proposed system could be added with additional functionalities easily. In this project, if the user wants to improve the performance further then this software can be easily upgraded.

8.1.4 Preventive maintenance:

Preventive maintenance involves changes made to a system to reduce the changes of features system failure. The possible occurrence of error that might occur are forecasted and prevented with suitable preventive problems. If the user wants to improve the performance of any process then the new features can be added to the system for this project.

CHAPTER 9

CONCLUSION AND FUTURE WORK

In conclusion, the proposed cyberbullying detection system, utilizing the (K-SVM) algorithm, offers a robust and efficient solution to address the growing concern of cyberbullying on social media platforms. By leveraging machine learning techniques, the system can automatically identify instances of cyberbullying in social media content. The implementation of the system involves several essential modules, including data loading, data pre-processing, feature selection, K-SVM training, testing, evaluation, and performance analysis. The system's advantages lie in its ability to deliver high accuracy in distinguishing between cyberbullying and non-cyberbullying content, enabling users to take prompt action to create a safer online environment. Future work for the cyberbullying detection system could focus on enhancing its performance and effectiveness by exploring more advanced machine learning techniques. One promising avenue is the integration of deep learning models, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), which can capture complex patterns and semantic relationships in text data. Additionally, incorporating sentiment analysis and contextual information could improve the system's ability to understand the intent behind social media content. Another crucial aspect for future work is increasing the system's adaptability to diverse languages and cultural nuances, enabling it to effectively detect cyberbullying across different regions and communities.

APPENDICES

APPENDIX 1

SOURCE CODE

FirstPhase.form:

```
package riskassessment;

import java.io.File;

import java.util.ArrayList;

import org.jsoup.Jsoup;

import org.jsoup.nodes.Document;

import org.jsoup.nodes.Element;

import org.jsoup.select.Elements;

public class FirstPhase extends javax.swing.JFrame {

    public static ArrayList cluster1=new ArrayList();

    public static ArrayList cluster2=new ArrayList();

    public static ArrayList cluster3=new ArrayList();

    public static ArrayList cluster4=new ArrayList();

    public FirstPhase() {

        initComponents();

    }

    private void initComponents() {
```

```

jPanel1 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

jTextArea1 = new javax.swing.JTextArea();

jButton2 = new javax.swing.JButton();

ArrayList FileNames=MainFrame.FileNames;

for(int i=0;i<FileNames.size();i++)

{

    String fn=FileNames.get(i).toString().trim();

    System.out.println(fn.trim());

    try

    {

        File input = new File(fn.trim());

        Document doc = Jsoup.parse(input, "UTF-8");

        Elements links = doc.select("post");

        for (Element link : links)

        {

            String postid=link.attr("id");

            System.out.println("postid: "+postid.trim());

```

```

doc = Jsoup.parse(link.toString())

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTextArea jTextArea1;

}

```

SecondPhase.form:

```

package riskassessment;

import java.io.File;

import java.io.FileInputStream;

import java.util.ArrayList;

public class SecondPhase extends javax.swing.JFrame {

    public static ArrayList Content=new ArrayList();

    public static ArrayList Result=new ArrayList();

    public static ArrayList cyberbullyingWords=new ArrayList();

    public SecondPhase() {

        jTextArea1.setText("");

        ArrayList cluster1=FirstPhase.cluster1;

```

```

        ArrayList cluster2=FirstPhase.cluster2;

        ArrayList cluster3=FirstPhase.cluster3;

        ArrayList cluster4=FirstPhase.cluster4;

        try {

            for (javax.swing.UIManager.LookAndFeelInfo info :
                javax.swing.UIManager.getInstalledLookAndFeels()) {

                if ("Nimbus".equals(info.getName())) {

                    javax.swing.UIManager.setLookAndFeel(info.getClassName());

                    break;

                }

            }

        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(SecondPhase.class.getName()).l
            og(java.util.logging.Level.SEVERE, null, ex);

        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(SecondPhase.class.getName()).l
            og(java.util.logging.Level.SEVERE, null, ex);

        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(SecondPhase.class.getName()).l
            og(java.util.logging.Level.SEVERE, null, ex);

        } catch (javax.swing.UnsupportedLookAndFeelException ex)
        {
            java.util.logging.Logger.getLogger(SecondPhase.class.getName()).

```



```

log(java.util.logging.Level.SEVERE, null, ex);

    }

    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {

            new SecondPhase().setVisible(true);

        }

    });

}

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

}

```

TrainingTesting.form:

```

private void initComponents() {

    jPanel1 = new javax.swing.JPanel();

    jLabel1 = new javax.swing.JLabel();

    jButton1 = new javax.swing.JButton();

    jScrollPane1 = new javax.swing.JScrollPane();

    jTextArea1 = new javax.swing.JTextArea();

    jScrollPane2 = new javax.swing.JScrollPane();

    jTextArea2 = new javax.swing.JTextArea();

```

```

jPanel1.setBackground(new java.awt.Color(102, 0, 102));

for(int i=0;i<((allNormalData.size()*80)/100);i++)

{

    String data=allNormalData.get(i).toString().trim();

    String result=allNormalResults.get(i).toString().trim();

    jTextArea1.append(data.trim()+" --> "+result.trim()+"\n");

    allTrainingData.add(data.trim());

    allTrainingResults.add(result.trim());

}

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JTextArea jTextArea1;

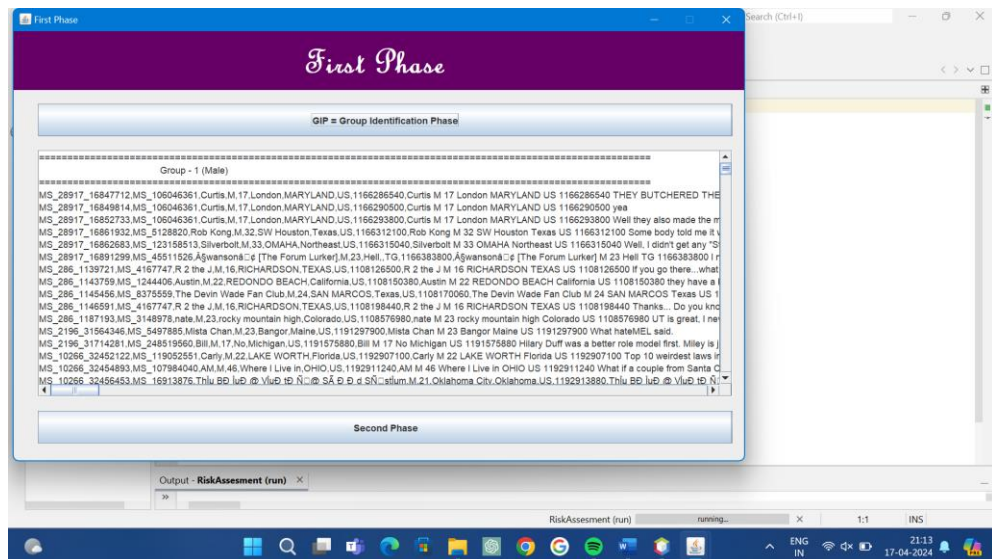
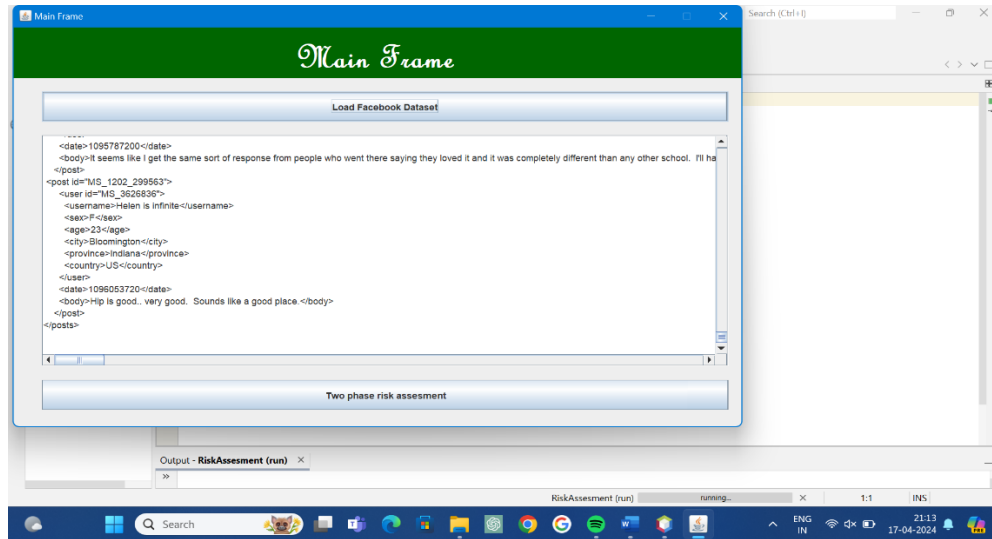
private javax.swing.JTextArea jTextArea2;

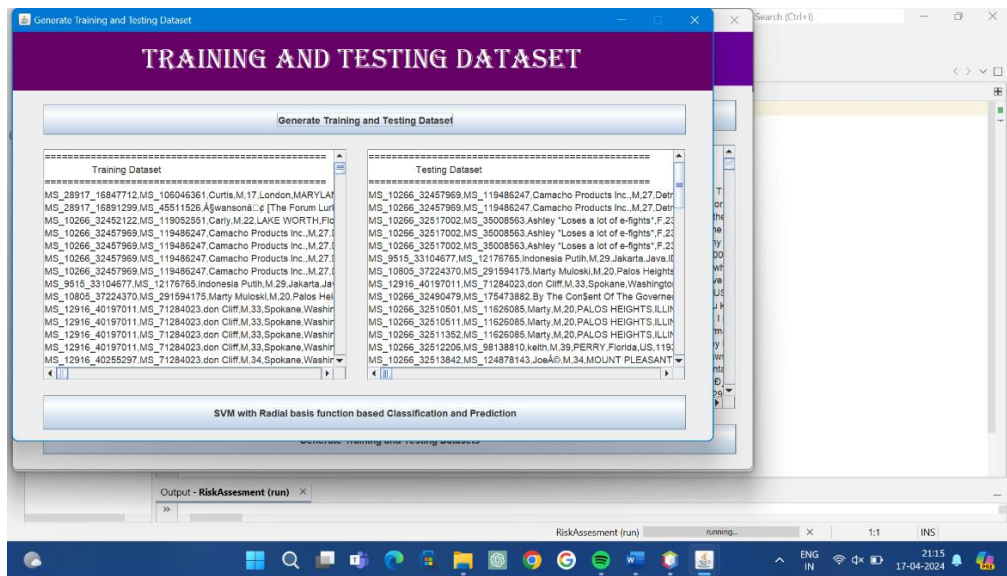
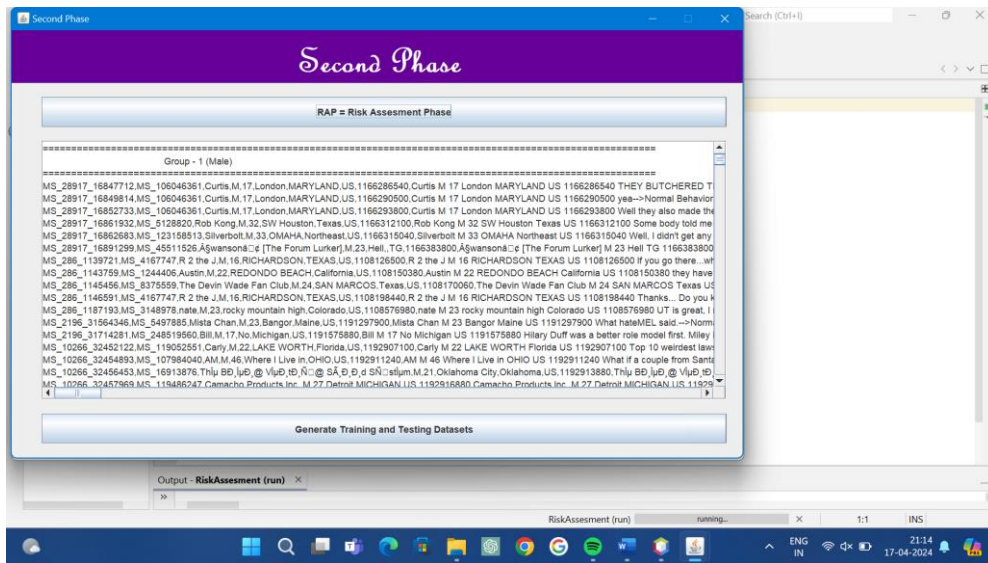
}

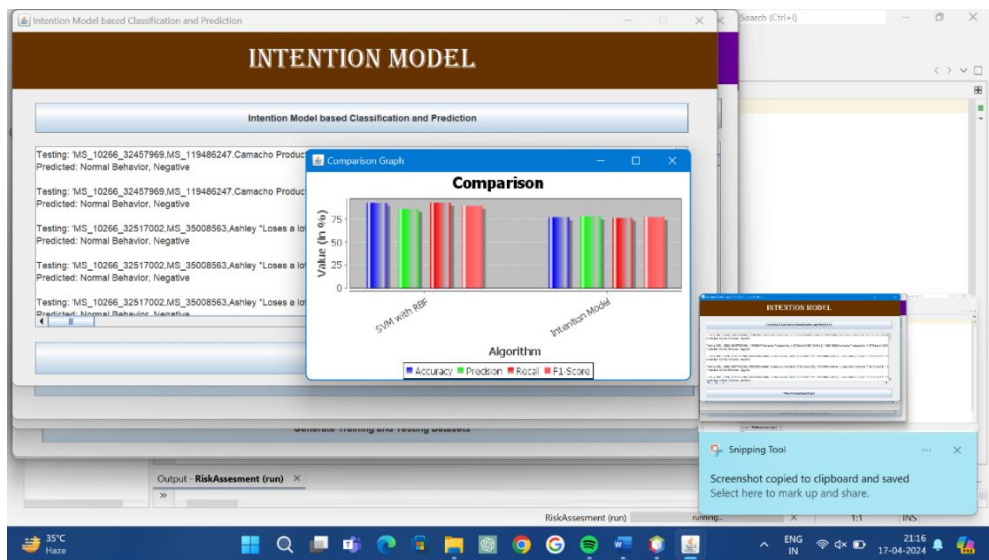
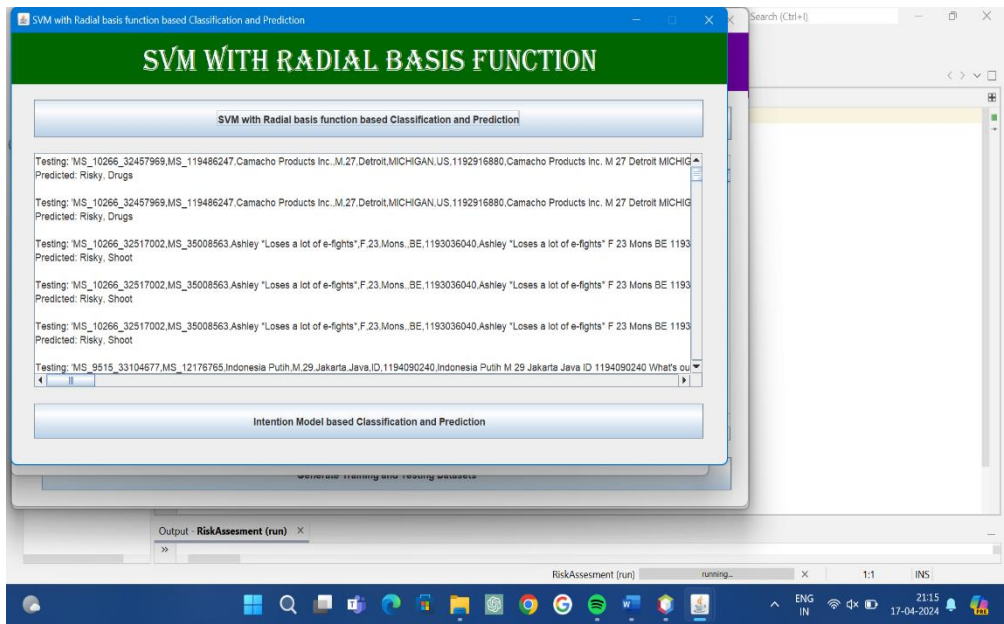
```

APPENDIX 2

SCREEN SHOTS







REFERENCES

1. Belal Abdullah Hezam Murshed, Jemal Abawajy Suresha Mallappa, Mufeed Ahmed Naji Saif, and Hasib Daowd Esmail al-Ariki (2022) ‘DEA- RNN: A Hybrid Deep Learning Approach for Cyberbullying Detection in Twitter Social Media Platform’, Volume:10, DOI:10.1109/ACCESS.2022.3153675.
2. Fatma Elsafoury, Stamos Katsigiannis, Zeeshan Pervez, Naeem Ramzan (2021) ‘When the Timeline Meets the Pipeline: A Survey on Automated Cyberbullying Detection’, Volume:9, DOI:10.1109/ACCESS.2021.3098979.
3. Jose Maria Molero, Jorge Perez Martin, Alvaro Rodrigo, Anselmo Penas (2023) ‘Offensive Language Detection in Spanish Social Media: Testing From Bag-of-Words to Transformers Models’, Volume: 11, DOI: 10.1109/ACCESS.2023.3310244.
4. Krishanu Maity, Raghav Jain, Prince Jha, Sriparna Saha (2023) ‘Explainable Cyberbullying Detection in Hinglish: A Generative Approach’, Early Access , DOI: 10.1109/TCSS.2023.3333675.
5. Liba Kestsbain, Biju Issac, Xiaomin Chen, Seibu Mary Jacob (2023) ‘A Multi-Stage Machine Learning and Fuzzy Approach to CyberHateDetection’, Volume:11, DOI:10.1109/ACCESS.2023.3282834.
6. Mohammed Hussein Obain, Shaw kat Kamal Guirguis, Sal eh Mesbah Elkaffas (2023) ‘Cyberbullying Detection and Severity DeterminationModel’, Volume:11, DOI:10.1109/ACCESS.2023.33131.
7. Murtaza, Henry Friday Nweke, Ihsan Ali, Ghulam Mujtaba, Haruna Chiroma, Hasan Ali Khattak, Abdullah Gani (2019)

- ‘Predicting Cyberbullying on Social Media in the Big Data Era Using Machine Learning Algorithms: Review of Literature and Open Challenges’, Volume:7, DOI:10.1109/ACCESS.2019.28354.
8. Ravuri Daniel, T. Satyanarayana Murthy, Ch. D. V. P. Kumari, E. Laxmi Lydia, Mohamad Khairi Ishak, Myriam Hadjouni, Samih M. Mostafa (2023) ‘Ensemble Learning With Tournament Selected Glowworm Swarm Optimization Algorithm for Cyberbullying Detection on Social Media’, VOLUME:11, DOI:10.1109/ACCESS.2023.33269.
9. Semiu Salawu, Yulan He, and Joanna Lumsden (2020) ‘Approaches to Automated Detection of Cyberbullying: A Survey’ Volume: 11, DOI: 10.1109/TAFRC.2017.2761757.
10. Teoh Hwai Teng and Kasturi Dewi Varathan (2023) ‘Cyberbullying Detection in Social Networks: A Comparison Between Machine Learning and Transfer Learning Approaches’, Volume: 11, DOI: 10.1109/ACCESS.2023.3275130.



Nandha

College of Technology

Affiliated to Anna University, Chennai, Approved by AICTE, New Delhi.
Erode - 638 052



10th NATIONAL CONFERENCE ON

EMERGING TRENDS IN ENGINEERING AND TECHNOLOGY

Certificate of Participation

ISBN Number: 978-81-967307-5-8

This is to certify that Dr./Prof./Mr./Ms JENIDA P has participated and

CYBERBULLYING DETECTION ON SOCIAL

presented a paper entitled MEDIA TEXTS USING K-SVM ALGORITHM in the

National Conference on "Emerging Trends in Engineering and Technology" held on 06th April 2024.

ORGANISING SECRETARY

HEAD OF THE DEPARTMENT

CONVENER

PRINCIPAL



Nandha

College of Technology

Affiliated to Anna University, Chennai, Approved by AICTE, New Delhi.
Erode - 638 052



10th NATIONAL CONFERENCE ON

EMERGING TRENDS IN ENGINEERING AND TECHNOLOGY

Certificate of Participation

ISBN Number: 978-81-967307-5-8

This is to certify that Dr./Prof./Mr./Ms KAVISHNI .S has participated and

presented a paper entitled MEDIA TEXTS USING K-SUM ALGORITHM in the

National Conference on "Emerging Trends in Engineering and Technology" held on 06th April 2024.


ORGANISING SECRETARY


HEAD OF THE DEPARTMENT


CONVENOR
PRINCIPAL



Nandha College of Technology

Affiliated to Anna University, Chennai, Approved by AICTE, New Delhi.
Erode - 638 052



10th NATIONAL CONFERENCE ON EMERGING TRENDS IN ENGINEERING AND TECHNOLOGY

Certificate of Participation

ISBN Number: 978-81-967307-5-8

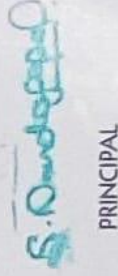
This is to certify that Dr./Prof./Mr./Ms Sowmya . G has participated and
presented a paper entitled MEDIA TEXTS USING K-SVM ALGORITHM in the
CYBERBULLING DETECTION ON SOCIAL

National Conference on "Emerging Trends in Engineering and Technology" held on 06th April 2024.


ORGANISING SECRETARY


HEAD OF THE DEPARTMENT


CONVENER


PRINCIPAL

Fwd: Acceptance of your Manuscript ID: IJICTE.436780 - IJICTE



karthikeyan c 2 days ago

to me ▾



----- Forwarded message -----

From: **International Journal of Information and Communication Technology**

<journaleditor@igi-global.com>

Date: Mon, 1 Apr, 2024, 08:49

Subject: Acceptance of your Manuscript ID: IJICTE.436780 - IJICTE

To: <ckarthikeyan.mecse@ksriet.ac.in>

Dear C.Karthikeyan.

We are pleased to inform you that we have received the article entitled "MACHINE LEARNING BASED CYBERBULLYING DETECTION ON SOCIAL MEDIA TEXTS" on 15 February 2024. Our reviewer have been accepted for publication in the International Journal of Information and Communication Technology Education(IJICTE) ISSN: 1550-1876|EISSN: 1550-1337.

Thank you for choosing to publish in our Journal.

Co-Editor

International Journal of Information and Communication Technology Education (IJICTE).