

Project Title

**Shielding against SMS spam and
Cyberbullying**

A PROJECT REPORT

Submitted by

BL.EN.U4CSE20092

M. Nitin Sai

BL.EN.U4CSE20114

N.Leeladhar Royal

BL.EN.U4CSE20113

N. Chakravarthi

BL.EN.U4CSE20109

N Siva Sai Raghu

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING



AMRITA SCHOOL OF COMPUTING, BENGALURU

AMRITA VISHWA VIDYAPEETHAM

BENGALURU 560 035

APRIL 2024

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING, BANGALORE, 560035



BONAFIDE CERTIFICATE

This is to certify that the project report Shielding against SMS spam and Cyberbullying is submitted by

BL.EN.U4CSE20092

Majji Nitin Sai

BL.EN.U4CSE20109

N.N.V. Siva Sai Raghu

BL.EN.U4CSE20113

N. Chakravarthi

BL.EN.U4CSE20114

N. Leeladhar Royal

in partial fulfillment of the requirements as part of **Bachelor of Technology** in
“**COMPUTER SCIENCE AND ENGINEERING**” is a bonafide record of the work carried
out under my guidance and supervision at Amrita School of Computing, Bangalore.

Dr. Deepa Gupta
Professor, School of Computing
Dept. of CSE

Dr. Gopalakrishnan E. A .,
Professor and Chairperson,
Dept. of CSE

EXAMINER 1

Dr. Manju Khanna.

EXAMINER 2

Mr. Dinesh Kumar

ACKNOWLEDGEMENTS

The satisfaction that accompanies successful completion of any task would be incomplete without mention of people who made it possible, and whose constant encouragement and guidance have been source of inspiration throughout the course of this project work.

We offer our sincere pranams at the lotus feet of "AMMA", **MATA AMRITANANDAMAYI DEVI** who showered her blessing upon us throughout the course of this project work.

We owe our gratitude to **Prof. Manoj P.**, Director, Amrita Vishwa Vidyapeetham Bengaluru Campus. We thank **Dr. Sriram Devanathan**, Principal Amrita School of Computing, Bengaluru for his support and inspiration.

We would like place our heartfelt gratitude to **Dr. Gopalakrishnan E.A.**, Chair, Department of Computer Science and Engineering, Amrita School of Computing, Bengaluru for his valuable support and inspiration.

It is a great pleasure to express our gratitude and indebtedness to our project guide: **Dr. Deepa Gupta, Professor**, Department of Computer Science and Engineering, Amrita School of Computing, Bengaluru for her/his valuable guidance, encouragement, moral support, and affection throughout the project work.

We would like to thank express our gratitude to project panel members for their suggestions, encouragement, and moral support during the process of project work and all faculty members for their academic support. Finally, we are forever grateful to our parents, who have loved, supported and encouraged us in all our endeavors.

ABSTRACT

Online messaging services like WhatsApp are plagued by widespread problems like spam and cyberbullying, which impede communication and jeopardize people's wellbeing. Our project, "Shielding Against SMS and Cyberbully," uses state-of-the-art deep learning and natural language processing (NLP) techniques to address these issues. Our main objective is to create reliable algorithms that can quickly detect and intercept malicious content in real time. By means of thorough testing and assessment, we hope to demonstrate the effectiveness of our strategy in thwarting spam and cyberbullying, thereby establishing a more secure online environment for users. In addition, we place a high priority on user privacy and data security, upholding strict moral guidelines and continuously improving our algorithms to accommodate various linguistic situations and cultural quirks.

Spam and cyberbullying have become commonplace on internet messaging services, compromising personal safety and communication integrity. Our research, "Shielding Against SMS and Cyberbully," uses cutting-edge deep learning and natural language processing (NLP) methods in response. Our main goal is to create intelligent algorithms that can quickly detect and remove hazardous information. We also incorporate contextual cues and dynamic feedback mechanisms to continuously improve our algorithms' performance in a variety of linguistic circumstances. In addition, our approach highlights how crucial user privacy and data security are, protecting private data while thwarting criminal activity and promoting responsible online behaviour.

Effective mitigation techniques are required considering the rise of spam and cyberbullying on platforms like WhatsApp to protect users and preserve the integrity of communication. Our research, "Shielding Against SMS and Cyberbully," uses cutting-edge deep learning and natural language processing (NLP) approaches to address these issues. Our main goal is to create reliable algorithms that can quickly identify and block hazardous content. By means of extensive testing and assessment, we aim to prove the effectiveness of our method in thwarting spam and cyberbullying, therefore creating a more secure online environment for users. Furthermore, we follow ethical rules, give top priority to user privacy and data security, and continuously improve our algorithms to accommodate changing linguistic and cultural situations.

TABLE OF CONTENTS

	Page No.
ACKNOWLEDGEMENTS	iv
ABSTRACT	v
LIST OF CONTENTS	vi
LIST OF FIGURES	vii
LIST OF FIGURES	viii
CHAPTER 1- INTRODUCTION	1
1.1 INTRODUCTION	
1.2 MOTIVATION	
CHAPTER 2 – LITERATURE SURVEY	3
CHAPTER 3 – SYSTEM SPECIFICATIONS	14
3.1 SOFTWARE REQUIREMENTS	
CHAPTER 4 – SYSTEM DESIGN	15
CHAPTER 5 – SYSTEM IMPLEMENTATION	19
5.1 MODULES USED WITH DESCRIPTION	
CHAPTER 6 – SYSTEM TESTING	22
CHAPTER 7 – RESULT AND ANALYSIS	25
CHAPTER 8 – CONCLUSION AND FUTURE SCOPE	41
REFERENCES	42

LIST OF FIGURES

Table 6.1	Hyper Parameters used for Bert models	23
Table 6.2	Hyper Parameters used for DL models	23
Table 7.1	Shows the results obtained from the Cyber Bullying Dataset	27
Table 7.2	Results obtained from Spam Data set	31
Table 7.3	Classification report for LSTM	31
Table 7.4	Metrics of the New Dataset	34
Table 7.5	Classification report for new model	34

LIST OF TABLES

Fig. 4.1	Architecture Design	15
Fig 6.1	Cyber Bullying Dataset	22
Fig 6.2	SMS Spam Dataset	22
Fig6.3	New Concatenated dataset	24
Fig 6.4	Hyper-parameters for Bert Models	20
Fig 7.1	Classification Report For bert-base-uncases+lstm	30
Fig 7.2	Training and validation accuracy and loss curves for Bert-base-uncased + lstm	30
Fig 7.3	All output Screenshots	40

CHAPTER - 1

INTRODUCTION

Online messaging services, such as WhatsApp, have become commonplace tools for communication in the digital age. These platforms are the main means of instant messaging. But the ease of use and connectedness that these platforms provide have also brought up unanticipated difficulties, most notably the surge in spam and cyberbullying. Cyberbullying, which is defined as using electronic communication to harass or threaten someone, has become a widespread problem that affects people of all ages and backgrounds. Similar to this, spam messages jeopardize users' privacy and security by flooding their inboxes with uninvited and frequently hazardous content. Beyond just being inconvenient, the effects of spam and cyberbullying have a significant impact on people's mental health, social well-being, and general quality of life. Studies reveal that those subjected to cyberbullying encounter elevated degrees of anxiety, sadness, and social distancing, resulting in enduring consequences that may endure till adulthood. In addition, the constant onslaught of spam messages exposes users to malware, phishing scams, and other types of cybercrime in addition to interfering with communication.

Considering this, creative approaches are desperately needed to lessen the negative impacts of spam and cyberbullying on online messaging services. Because message data is so large and complicated, traditional methods like keyword-based filtering and manual moderation are either inefficient or unworkable. As a result, there is increasing interest in using cutting-edge technology to create more complex and pre-emptive defences against these dangers, including deep learning and natural language processing (NLP).

The project, "Shielding Against SMS and Cyberbully," aims to address this issue by utilizing real-time detection and mitigation of spam and cyberbullying through the application of deep learning and natural language processing techniques. Our system looks at contextual cues, user interactions, and message content to find patterns that point to potentially harmful behaviour and take action before more harm is done. Our goal is to prove the efficacy and scalability of our approach in establishing a more secure and safe digital environment for online messaging platform users through thorough experimentation and evaluation. By doing this, we want to give people the freedom and confidence to speak out without worrying about being harassed or taken advantage of.

MOTIVATION

To address the pressing challenge of cyberbullying and spam messages in online platforms, our proposed model integrates various deep learning architectures tailored for text classification. The project encompasses a comprehensive comparative study of different deep learning models, including traditional Deep Neural Networks (DNNs), Long Short-Term Memory (LSTM) networks, Bidirectional LSTMs (BiLSTMs), and cutting-edge pre-trained transformer models like BERT (both BERT-Base-Uncased and Distilled BERT Embeddings). By exploring a spectrum of architectures, we aim to identify the most effective approach for detecting and mitigating cyberbullying and spam messages across diverse digital environments.

The overarching goal of this endeavour is to enhance user safety and security within the digital landscape by combatting online threats more effectively. With the proliferation of social media and digital communication platforms, the prevalence of cyberbullying and spam has become a significant concern, impacting the well-being and safety of users worldwide. By leveraging advanced deep learning techniques, our model seeks to provide proactive measures for identifying and addressing such malicious activities, thereby fostering a safer and more inclusive online environment for all users.

Through rigorous experimentation and evaluation, our research endeavours to contribute novel insights into the efficacy of various deep learning models in tackling cyberbullying and spam detection. By scrutinizing the performance of different architectures under diverse datasets and scenarios, we aim to provide actionable recommendations for deploying effective defence mechanisms against online threats. Ultimately, this project underscores our commitment to leveraging cutting-edge technology to safeguard users' digital experiences and promote a more secure and resilient online ecosystem.

CHAPTER - 2

LITERATURE SURVEY

Paper 1: "Advancements in SMS Spam Filtering: A Comparative Analysis of Machine Learning Models and Emerging Challenges"

Summary:

Salman, Ikram, and Kaafar (2023) conducted a comprehensive investigation into the effectiveness of various machine learning models for SMS spam filtering, as presented in the ICCECS 2023 Proceedings. The study meticulously compared deep learning architectures with traditional shallow models like SVMs and Naive Bayes. It elucidated the intricate evasion techniques adopted by spammers, posing significant challenges to existing spam filters. The research emphasized critical deficiencies in current SMS spam filtering systems and underscored the urgent need for more resilient solutions.

Gaps:

- 1. Inadequate Evasion Handling:** The study identified shortcomings in existing spam filters in countering the sophisticated evasion strategies employed by spammers, revealing a gap in understanding evasive tactics.
- 2. Limited Robustness:** Current spam detection systems lack robustness in addressing evolving spamming techniques, indicating a gap in adaptive and dynamic filtering mechanisms.
- 3. Underexplored Methodologies:** There exists a gap in exploring advanced techniques such as transfer learning and NLP integration in SMS spam filtering, suggesting a need for further research in leveraging these methodologies.

Objectives:

- 1. Enhanced Detection Capability:** Investigate methods to enhance the accuracy of SMS spam filtering systems in detecting and categorizing evasive spam messages.
- 2. Adaptive Filtering Mechanisms:** Develop dynamic filtering mechanisms capable of adapting to emerging spamming techniques and evolving threat landscapes.

Paper 2: "SMS Spam Detection Using Deep Learning Techniques: A Comparative Analysis of DNN Vs LSTM Vs Bi-LSTM"

Summary:

Gandhi, Sarangi, Saxena, and Sahoo (2023) conducted a study on SMS spam detection utilizing deep learning methods, presented at CISES in 2023. The research compared the efficacy of three deep learning techniques: Deep Neural Network (DNN), Long Short-Term Memory (LSTM), and Bi-directional LSTM (Bi-LSTM). While the study provided insights into model performance, it neglected practical considerations such as preprocessing and hyperparameter tuning crucial for real-world implementation. Additionally, a notable gap exists in the absence of analysis regarding transfer learning, a technique that could enhance the robustness of spam detection methods.

Gaps:

- 1. Practical Concerns Ignored:** The study overlooks practical considerations like preprocessing techniques and hyperparameter optimization, essential for the effective deployment of spam detection models in real-world scenarios.
- 2. Absence of Transfer Learning Analysis:** The research fails to explore the potential benefits of transfer learning, which could improve the robustness and generalizability of spam detection systems by leveraging pre-trained models.
- 3. Limited Real-world Applicability:** Due to the neglect of practical concerns and advanced methodologies like transfer learning, the proposed spam detection techniques may lack real-world robustness and effectiveness.

Objectives:

- 1. Integration of Practical Considerations:** Address the practical concerns of preprocessing and hyperparameter adjustment to enhance the real-world applicability and performance of SMS spam detection models.
- 2. Exploration of Transfer Learning:** Investigate the potential benefits of transfer learning in SMS spam detection to improve model robustness and generalizability across different datasets and scenarios.
- 3. Enhanced Robustness:** Develop spam detection techniques that are not only theoretically sound but also practical and robust for real-world deployment, thereby addressing the identified gaps in the existing research.

Paper 3: "SpotSpam: Intention Analysis-driven SMS Spam Detection Using BERT Embeddings"

Summary:

Oswald, Simon, and Bhattacharya (2022) present "SpotSpam" in the ACM Transactions on the Web, addressing the pressing issue of combating SMS spam. The proposed method utilizes pre-defined intention labels and contextual embeddings generated by NLP models, particularly BERT embeddings, to analyze intentions behind SMS messages for spam detection. However, the research lacks critical analysis regarding the scalability, robustness, and generalizability of the proposed method across diverse SMS spam datasets and real-world scenarios. Additionally, computational complexity and resource requirements associated with integrating BERT embeddings into spam detection systems remain unexplored, limiting the practicality of the approach.

Gaps:

1. **Lack of Critical Analysis:** The study overlooks critical analysis concerning the scalability, robustness, and generalizability of the proposed method across various SMS spam datasets and real-world contexts.
2. **Unexplored Computational Complexity:** The research fails to address the computational complexity and resource needs associated with incorporating BERT embeddings into SMS spam detection systems, potentially hindering practical implementation.
3. **Limited Real-world Applicability:** Without addressing scalability, robustness, and computational challenges, the proposed SpotSpam solution may lack efficacy and applicability in real-world scenarios.

Objectives:

1. **Scalability and Generalizability Assessment:** Conduct a thorough evaluation of the proposed SpotSpam method's scalability, robustness, and generalizability across diverse SMS spam datasets and real-world scenarios.
2. **Address Computational Complexity:** Investigate methods to mitigate computational complexity and resource requirements associated with integrating BERT embeddings into SMS spam detection systems, enhancing the approach's practicality.
3. **Enhanced Real-world Applicability:** Develop improvements to the SpotSpam solution to ensure its effectiveness and efficiency in real-world applications, thereby filling the identified gaps and addressing practical concerns.

Paper 4: "Improving Cyberbullying Detection with User Interaction"

Summary:

Ge, Cheng, and Liu (2021) present their work in the Proceedings of the Web Conference 2021 (WWW '21), aiming to enhance cyberbullying detection by incorporating user interactions, subject coherence among comments, and temporal correlations between comments. By emphasizing the importance of modeling these elements, the research seeks to enhance the efficacy and accuracy of cyberbullying detection. However, significant gaps remain unaddressed, particularly concerning real-time detection issues, multimodal data analysis, and contextual subtleties inherent in cyberbullying detection.

Gaps:

- 1. Real-time Detection Challenges:** The study overlooks real-time detection issues, failing to address the need for timely identification and intervention in cyberbullying incidents.
- 2. Multimodal Data Analysis:** The research neglects the potential of incorporating multimodal data sources (e.g., text, images, videos) for more comprehensive cyberbullying detection.
- 3. Contextual Subtleties Ignored:** The study does not account for contextual subtleties in cyberbullying detection, such as cultural nuances and evolving online communication trends, which can impact detection accuracy.

Objectives:

- 1. Real-time Detection Solutions:** Develop techniques for real-time cyberbullying detection to enable timely intervention and mitigation of online bullying behavior.
- 2. Multimodal Data Integration:** Explore methods for integrating multimodal data analysis into cyberbullying detection systems to leverage diverse data sources for improved detection accuracy.
- 3. Context-aware Detection Models:** Develop context-aware cyberbullying detection models that consider cultural nuances, linguistic variations, and evolving online communication patterns to enhance detection efficacy in diverse online environments.

Paper 5: "MTBullyGNN: A Graph Neural Network-Based Multitask Framework for Cyberbullying Detection"

Summary:

Maity, Saha, and Bhattacharyya (2022) introduce MTBullyGNN in the IEEE Transactions on Computational Social Systems, presenting a novel framework for multitask learning in cyberbullying detection utilizing Graph Neural Networks (GNNs). MTBullyGNN integrates user interactions with content information to perform three main activities: identifying cyberbullying postings, categorizing their severity, and identifying bullying targets. The framework employs various sub-modules, including sentiment analysis and target identification, for different components of the detection process, while GNNs capture user interactions. However, the research acknowledges significant limitations, such as the framework's dependence on the quality and quantity of training data.

Gaps:

- 1. Data Dependency Concerns:** The study recognizes the reliance of MTBullyGNN on the quality and volume of training data, which may limit its effectiveness in scenarios with limited or biased datasets.
- 2. Scope for Further Social Network Exploration:** The research suggests future exploration of other social network aspects beyond user interactions and content, indicating a gap in understanding and leveraging additional features for cyberbullying detection.
- 3. Need for Explainability Methodologies:** The study identifies the absence of explainability methodologies in MTBullyGNN as a constraint, highlighting the importance of interpretable models for trust and understanding in cyberbullying detection systems.

Objectives:

- 1. Data Augmentation and Bias Mitigation:** Investigate techniques for data augmentation and bias mitigation to enhance MTBullyGNN's robustness and applicability across diverse datasets.
- 2. Exploration of Additional Social Network Aspects:** Explore additional features and aspects of social networks beyond user interactions and content to improve cyberbullying detection performance and accuracy.
- 3. Incorporation of Explainability Methodologies:** Develop explainability methodologies to enhance the transparency and interpretability of MTBullyGNN's decision-making process, fostering trust and understanding in cyberbullying detection systems.

Paper 6: "Cyber Bullying Detection using Natural Language Processing (NLP) and Text Analytics"

Summary:

Hsien, Abdul Salam, and Kasinathan (2022) present their work in the IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE) Proceedings, focusing on cyberbullying detection through the integration of Natural Language Processing (NLP) and text analytics techniques. The study utilizes NLP methods like negation handling, sentiment analysis, and part-of-speech (POS) tagging to extract features from text input. These features are then analysed using text analytics techniques to identify patterns and associations, enabling machine learning models to classify text data into cyberbullying and non-bullying categories. However, the research underscores the need for more reliable and efficient feature selection methods tailored specifically for text classification. Additionally, feature selection algorithms must be enhanced to fully account for their interactions, thereby improving the accuracy and efficiency of cyberbullying detection systems.

Gaps:

- 1. Reliability of Feature Selection Methods:** The study highlights the inadequacy of current feature selection methods for text classification, indicating a need for more dependable approaches to enhance cyberbullying detection accuracy.
- 2. Consideration of Feature Interactions:** Feature selection algorithms must be improved to account for interactions between features, ensuring comprehensive analysis and classification of text data.
- 3. Efficiency Enhancement:** There is a gap in improving the efficiency of cyberbullying detection systems through the development of more efficient feature selection techniques, ultimately leading to safer online environments.

Objectives:

- 1. Development of Reliable Feature Selection Methods:** Research and develop more reliable feature selection methods tailored specifically for text classification tasks, with a focus on enhancing cyberbullying detection accuracy.
- 2. Integration of Feature Interaction Analysis:** Enhance feature selection algorithms to account for interactions between features, ensuring comprehensive analysis and classification of text data in cyberbullying detection systems.
- 3. Efficiency Improvement:** Develop more efficient feature selection techniques to enhance the efficiency and effectiveness of cyberbullying detection systems, thereby promoting safer online environments.

Paper 7: "Rapid Cyber-bullying Detection Method using Compact BERT Models"

Summary:

Behzadi, Harris, and Derakhshan (2021) present their work at the IEEE 15th International Conference on Semantic Computing (ICSC), introducing a technique that utilizes miniature versions of Bidirectional Encoder Representations from Transformers (BERT) models for rapid cyberbullying detection. The method demonstrates promising results when tested on a benchmark dataset. However, the paper lacks detailed coverage of specific evaluation measures such as precision, recall, F1-score, or comparative performance against other cyberbullying detection techniques. A more comprehensive assessment is necessary to fully understand the efficacy of the approach and its potential applicability in practical scenarios.

Gaps:

- 1. Insufficient Evaluation Measures:** The study lacks detailed coverage of evaluation measures such as precision, recall, F1-score, or comparative performance against existing cyberbullying detection techniques, hindering a comprehensive assessment of the proposed method's efficiency.
- 2. Lack of Comparative Analysis:** There is a gap in comparative analysis with other cyberbullying detection methods, which is essential for understanding the relative strengths and weaknesses of the proposed approach.
- 3. Limited Practical Applicability Assessment:** The paper does not sufficiently explore the potential practical applications of the proposed rapid cyberbullying detection method, leaving a gap in understanding its real-world effectiveness and usability.

Objectives:

- 1. Comprehensive Evaluation:** Conduct a thorough evaluation of the proposed rapid cyberbullying detection method, including precision, recall, F1-score, and comparative analysis with existing techniques, to assess its efficacy accurately.
- 2. Comparative Analysis:** Compare the performance of the proposed method with other cyberbullying detection techniques to identify its relative strengths and weaknesses.
- 3. Practical Applicability Assessment:** Explore the practical applications of the rapid cyberbullying detection method and assess its effectiveness and usability in real-world scenarios, filling the gap in understanding its potential for practical use.

Paper 8: "Can Bullying Detection Systems Help in School Violence Scenarios?: A Teachers' Perspective"

Summary:

Kim, Ho, Kim, Lee, and Seo (2020) explore the potential of technology-driven systems for detecting and intervening in incidents of school violence from the perspective of teachers. Published in the Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA '20), the study summarizes key findings from 35 teacher interviews, highlighting their concerns and providing design recommendations for corresponding systems. However, it reveals a significant disconnect between teacher requirements and research on school violence detection systems, addressing issues such as teacher workloads, algorithm correctness, and privacy. Notably, the study lacks consideration of student perspectives, despite offering design implications to address these issues. Incorporating student viewpoints would enrich the conversation and provide a more comprehensive understanding of the benefits and limitations of technology-driven approaches in combating school violence.

Gaps:

- 1. Missing Student Perspectives:** The study fails to incorporate perspectives from students, who are directly affected by school violence detection systems, limiting the depth and comprehensiveness of the research.
- 2. Disconnect between Requirements:** There is a notable disconnect between teacher requirements and existing research on school violence detection systems, indicating a need for better alignment and understanding of stakeholders' needs.
- 3. Incomplete Consideration of Ethical Concerns:** While the study addresses issues like algorithm correctness and privacy, it may overlook other ethical concerns related to the implementation and use of technology-driven approaches in schools.

Objectives:

- 1. Comprehensive Ethical Analysis:** Conduct a thorough ethical analysis of technology-driven approaches in combating school violence, addressing concerns beyond algorithm correctness and privacy to ensure responsible and ethical implementation.

Paper 9: "Optimized Twitter Cyberbullying Detection based on Deep Learning"

Summary:

Al-Ajlan and Ykhlef presented their work at the 21st Saudi Computer Society National Computer Conference (NCC) in 2018, addressing the limitations of existing cyberbullying detection methods on Twitter that rely on textual and user features. Their proposed method, OCDD, eliminates the need for feature extraction and selection by redefining tweets as word vectors to preserve semantics. OCDD combines a metaheuristic optimization algorithm for parameter fine-tuning with deep learning techniques for categorization. Despite OCDD's innovative approach of using word vectors, the paper lacks thorough assessment and comparison with current techniques, limiting its confirmation of effectiveness. Further analysis and comparison with established techniques are needed to improve OCDD's legitimacy and suitability for cyberbullying detection.

Gaps:

- 1. Limited Assessment and Comparison:** The paper lacks comprehensive assessment and comparison with current techniques in cyberbullying detection, hindering a thorough understanding of OCDD's effectiveness.
- 2. Insufficient Confirmation of Effectiveness:** While OCDD introduces an innovative approach using word vectors, it lacks thorough validation and confirmation of its effectiveness compared to existing techniques.
- 3. Lack of Real-world Validation:** The research does not provide validation of OCDD's suitability for real-world cyberbullying detection scenarios, limiting its practical applicability.

Objectives:

- 1. Comprehensive Evaluation and Comparison:** Conduct a thorough assessment and comparison of OCDD with established cyberbullying detection techniques to determine its effectiveness and advantages.
- 2. Real-world Applicability Assessment:** Evaluate OCDD's suitability for real-world cyberbullying detection scenarios through practical validation and testing, ensuring its practical applicability and relevance.

Paper 10: "Detection of Cyberbullying Using Deep Neural Network"

Summary:

Banerjee, Telavane, Gaikwad, and Vartak presented their paper at the 5th International Conference on Advanced Computing & Communication Systems (ICACCS) in 2019, introducing a novel approach based on Convolutional Neural Networks (CNNs) for cyberbullying detection. Recognizing the increasing concern about cyberbullying and its detrimental effects, the paper advocates for the use of CNNs for improved identification compared to current techniques. However, the research lacks a thorough analysis of the CNN model's architecture and training process, and it falls short in providing a comprehensive comparison with existing methods to demonstrate its superiority. To enhance the paper's contribution to cyberbullying detection, further investigation into these aspects is necessary. A more comprehensive analysis and comparison would elevate the suggested CNN-based detection approach, giving it legitimacy and relevance in preventing cyberbullying effectively.

Gaps:

- 1. Lack of Detailed Model Architecture and Training Process:** The paper does not provide a thorough analysis of the CNN model's architecture and training process, limiting understanding and reproducibility.
- 2. Inadequate Comparison with Existing Methods:** The research does not offer a comprehensive comparison with current cyberbullying detection techniques, hindering the demonstration of the suggested CNN-based approach's superiority.
- 3. Need for Enhanced Investigation:** Further investigation into the CNN model's architecture, training process, and comparison with existing methods is necessary to strengthen the paper's contribution to the field of cyberbullying detection.

Objectives:

- 1. Detailed Model Analysis:** Conduct a thorough analysis of the CNN model's architecture and training process, providing insights into its effectiveness and reproducibility.
- 2. Comprehensive Comparison:** Compare the CNN-based approach with existing cyberbullying detection methods to demonstrate its superiority in terms of accuracy, efficiency, and other relevant metrics.

Paper 11: "Bullying Hurts: A Survey on Non-Supervised Techniques for Cyberbullying Detection"

Summary:

Farag, Abou El-Seoud, McKee, and Hassan's paper, published in the Proceedings of the 8th International Conference on Software and Information Engineering (ICSIE '19), discusses the prevalence of cyberbullying and advocates for efficient detection techniques. The paper surveys current literature on non-supervised methods for identifying cyberbullying and suggests future research areas, including automated annotation, role detection, emotional state detection, and stylometric approaches. However, several significant gaps are evident in the paper. It may overlook cutting-edge methods due to inadequate coverage of unsupervised techniques. Additionally, it fails to assess and contrast the effectiveness of various approaches, neglects real-world implementation issues such as scalability and integration, and disregards user viewpoints essential for system acceptance. Filling these gaps would enhance the paper's contribution by providing a comprehensive understanding of non-supervised methods for cyberbullying detection and facilitating the development of useful and efficient detection systems.

Gaps:

- 1. Limited Coverage of Cutting-edge Methods:** The paper may miss out on innovative approaches due to inadequate coverage of unsupervised techniques in cyberbullying detection.
- 2. Neglect of Real-world Implementation Issues:** The paper overlooks real-world implementation challenges such as scalability and integration, which are crucial for the practical deployment of cyberbullying detection systems..

Objectives:

- 1. Comprehensive Coverage of Methods:** Conduct a thorough review of both established and cutting-edge unsupervised techniques for cyberbullying detection to provide a complete understanding of available approaches.
- 2. Consideration of Real-world Implementation Challenges:** Address scalability, integration, and other real-world implementation issues to ensure the practical feasibility of cyberbullying detection systems.
- 3. Incorporation of User Viewpoints:** Consider user perspectives and feedback to enhance the acceptance and effectiveness of cyberbullying detection systems in real-world settings.

CHAPTER – 3

SYSTEM SPECIFICATIONS

3.1 Software requirements

Natural Language Processing (NLP) Libraries:

- **NLTK (Natural Language Toolkit):** Provides a suite of tools for natural language processing, including tokenization, tagging, lemmatization, and part-of-speech tagging.

Deep Learning Libraries:

- **Transformers:** A library for natural language processing with pre-trained models, including BERT Tokenizer and BERT Model.
- **TensorFlow:** It is an open-source machine learning framework developed by Google, widely used for building and training various deep learning models.
- **Spellchecker:** It is a tool that automatically corrects spelling errors in text documents or input fields.
- **Scikeras:** It is a Python library that integrates the Keras deep learning framework with scikit-learn, providing a unified interface for building, training, and deploying machine learning models.

Data Visualization Library:

- **Matplotlib:** A library for creating plots and charts.

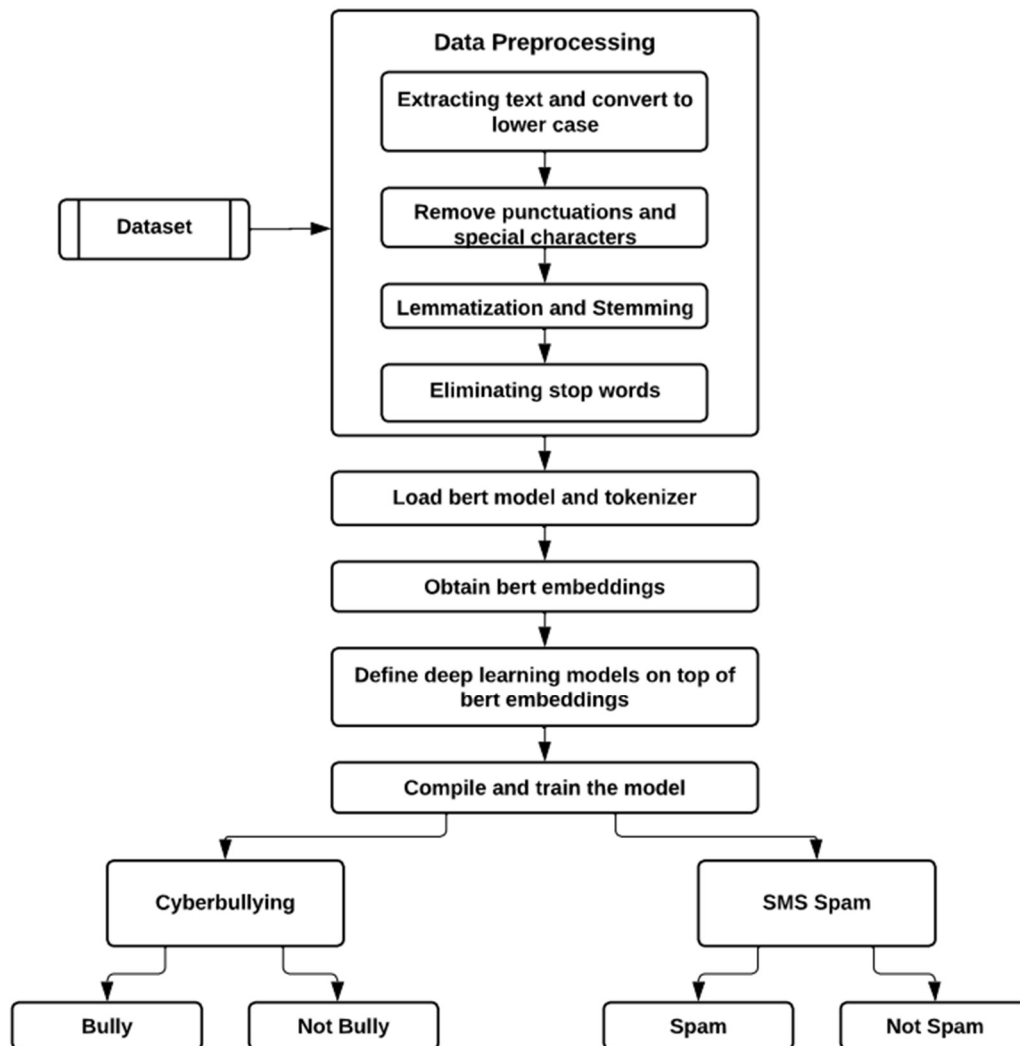
Other Libraries:

- **torch:** A library for tensor computation and deep learning.
- **NumPy:** A library for numerical computation and data manipulation.

CHAPTER - 4

SYSTEM DESIGN

Architecture Diagram



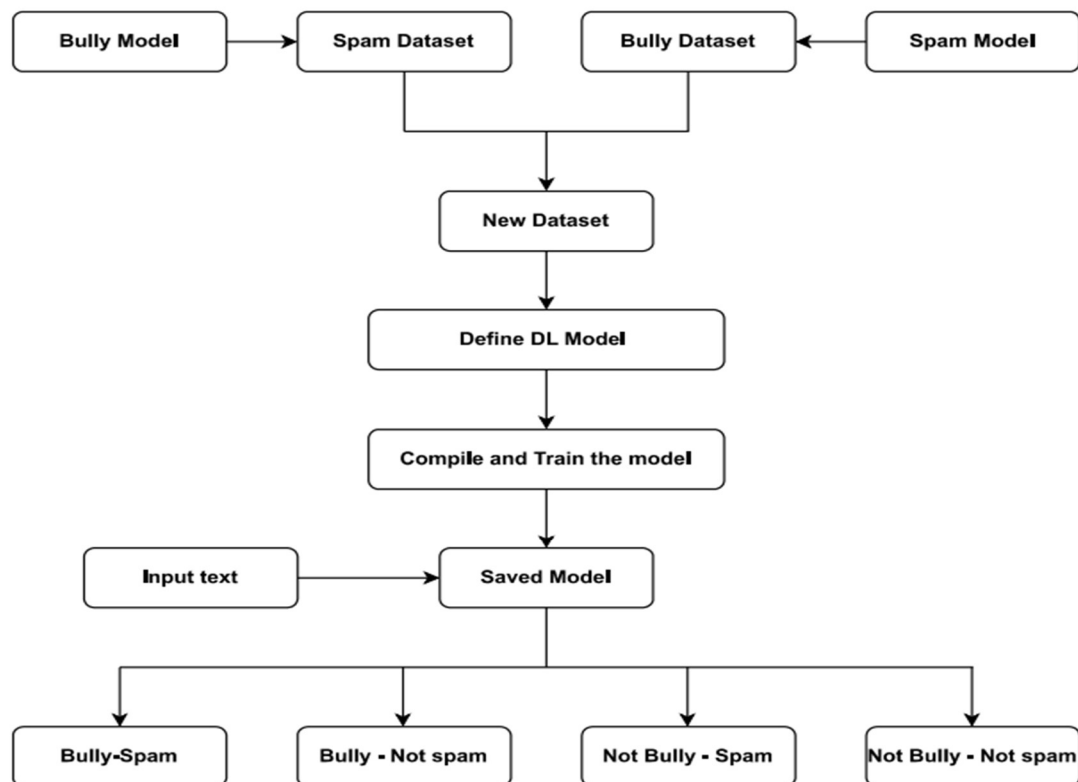


Fig 4.1: New Architecture Design

1. Data Preprocessing

- **Text Extraction:** The initial step involves extracting text from the source data. This may involve retrieving text from social media posts, emails, or other sources where cyberbullying detection is relevant.
- **Text Cleaning:** The extracted text might contain noise or irrelevant characters. This step involves cleaning the text by removing punctuation marks, special characters, HTML tags, and extra spaces.
- **Lowercasing:** Converting all text to lowercase letters can improve model performance by reducing vocabulary size and making the model less sensitive to case variations. For instance, the words "good" and "Good" would be treated as the same word.
- **Tokenization:** Here, text is segmented into individual words or phrases called tokens. This is a crucial step for the model to understand the structure and meaning of the text.
- **Stop Word Removal:** Some words, like "the," "a," "an," and "is," are common and don't carry much meaning. Removing these stop words can improve model performance by reducing the number of features the model needs to learn from.

2. Text Vectorization

- **Word Embeddings:** After preprocessing, text data needs to be converted into a numerical format that deep learning models can understand. This is achieved using word embedding techniques. Word embeddings assign a unique vector to each word, where words with similar meanings have similar vector representations. There are different word embedding techniques, including Word2Vec and GloVe.

3. Model Training

- **Model Architecture Selection:** Depending on the complexity of the task, various deep learning architectures can be chosen. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are commonly used architectures for text classification tasks.
- **Training the Model:** The preprocessed text data and the corresponding labels (e.g., bullying or not bullying) are fed into the chosen deep learning model. The model learns to identify patterns in the text data that differentiate between bullying and non-bullying content.

4. Model Evaluation

- **Testing and Performance Metrics:** After training, the model's performance is evaluated on a separate dataset it hasn't seen before. Metrics like accuracy, precision, recall, and F1-score are used to assess the model's effectiveness in classifying cyberbullying content.

5. Training Separate Models for Cyberbullying and Spam Detection

- **Specialization:** Training independent models allows them to specialize in recognizing the nuances of each type of content. Cyberbullying detection models can focus on identifying harmful language patterns, while spam detection models can target marketing or promotional content.
- **Potentially Higher Accuracy:** By specializing, each model can potentially achieve higher accuracy on its respective task compared to a single model trying to handle both classification problems.

6. Leveraging Pre-trained Models for Data Augmentation

The approach then utilizes these pre-trained models for data augmentation:

- **Making Predictions on Opposite Datasets:** The pre-trained models are used to generate predictions on datasets they weren't originally trained on (i.e., the cyberbullying model predicts on the spam dataset and vice versa).
- **Concatenated Dataset Creation:** The original datasets and the corresponding predictions from the pre-trained models are then combined to create a new, augmented dataset.

7. Benefits of Data Augmentation

- **Increased Training Data Volume:** This data augmentation technique effectively expands the training datasets for both cyberbullying and spam detection. Having more training data can lead to better model performance.
- **Improved Generalizability:** By incorporating examples from the opposite class (spam for cyberbullying model and vice versa) with the predictions from the pre-trained models, the new models might learn broader features that improve generalizability. This can help the models handle unseen variations of cyberbullying or spam content during real-world deployment.

8. Considerations and Potential Limitations

- **Quality of Pre-trained Model Predictions:** The effectiveness of this approach relies on the accuracy of the pre-trained models' predictions on the opposite datasets. Inaccurate predictions can introduce noise into the augmented data, potentially hindering the performance of the new models.
- **Impact on Model Bias:** Biases present in the pre-trained models can be amplified during data augmentation. It's crucial to evaluate the fairness and potential biases of the pre-trained models before using them for this purpose.
- **Computational Cost:** Training large deep learning models can be computationally expensive. Depending on the size and complexity of the original datasets and the chosen pre-trained models, this approach might require significant computational resources.

Overall, the approach presents a promising strategy to enhance the efficiency of cyberbullying and spam detection models by leveraging pre-trained models for data augmentation. However, careful consideration of the quality of pre-trained model predictions, potential biases, and computational costs is essential for successful implementation.

CHAPTER – 5

SYSTEM IMPLEMENTATION

5.1 Modules used with description:

1. Transformers (Hugging Face):

Description: Hugging Face created the open-source Transformers library, which offers pre-trained models and an easy-to-use interface for tasks related to natural language generation and understanding (NLU and NLG, respectively). Built on top of PyTorch and TensorFlow, it provides a vast array of pre-trained models for tasks like text classification, question answering, language translation, and text generation, including BERT, GPT.

2. The Natural Language Toolkit (NLTK):

Description: A complete platform for developing Python applications that interact with data in human languages is called NLTK. It offers user-friendly libraries and interfaces for a variety of operations, including parsing, tagging, tokenization, and stemming. For problems involving natural language processing, NLTK offers a variety of corpora, lexical resources, and modules.

3. Deep Learning Models:

- **Dense Neural Networks (DNNs)** are a foundational architecture in deep learning, consisting of densely connected layers of neurons. They are commonly used for tasks like classification and regression, where input features are transformed through multiple hidden layers to produce an output prediction.

- **Long Short-Term Memory (LSTM)** networks are a type of recurrent neural network (RNN) designed to capture long-term dependencies in sequential data. They utilize memory cells with gated units to selectively retain or forget information over time, making them effective for tasks such as time series prediction and natural language processing.
- **Bidirectional LSTM (BiLSTM)** networks enhance traditional LSTMs by processing input sequences in both forward and backward directions. This enables the model to capture context from past and future information simultaneously, leading to improved performance in tasks requiring context understanding, such as sentiment analysis and machine translation.

4. Large Language Models:

- **BERT (Bidirectional Encoder Representations from Transformers)** is a pre-trained transformer-based language model developed by Google. It learns contextual word representations by training on vast amounts of text data and excels in a wide range of natural language processing tasks, including sentiment analysis, question answering, and named entity recognition.
- **Distilled BERT** is a distilled version of BERT, developed by Hugging Face. It achieves comparable performance to BERT while being much smaller and faster, making it more suitable for deployment in resource-constrained environments.
- **ALBERT (A Lite BERT)** is another variant of BERT, designed to improve scalability and efficiency by reducing model size and computational cost. It employs parameter-sharing techniques and factorized embedding

parameterization to achieve significant reductions in memory footprint and training time without sacrificing performance.

- **Roberta (Robustly optimized BERT approach)** is a variant of BERT developed by Facebook AI. It introduces improvements to the training procedure and data augmentation techniques, resulting in enhanced robustness and performance across various natural language understanding tasks. Roberta achieves state-of-the-art results on benchmark datasets and has become a popular choice for fine-tuning in NLP applications.

CHAPTER - 6

SYSTEM TESTING

Part-A

The Cyberbullying Detection Dataset used in this project consists of a training set with 902 rows. Each row contains a message in the "text" column and a corresponding label indicating whether it's classified as cyberbullying (bully) or not (not-bully). Similarly, a testing set with 202 rows, structured identically, is utilized to evaluate the model's performance. Both datasets were sourced from Hugging Face, a platform renowned for its natural language processing datasets and models.

Text	Label
Man spit it out already	Not Bully
say that again imma b under yo bee lil bro	Not Bully
Say it donâ€™t spray it	Not Bully
Bro is just hungry for a dog	Not Bully
Fentanyl is pretty derpy	Not Bully
Who the fuck are you?	Bully
hairynigga635	Bully
Who Is u?	Not Bully

Fig 6.1: Cyber-Bullying Dataset

The SMS spam filtering dataset comprises 8194 SMS messages in the training set, each tagged with its classification as spam or not-spam. Additionally, a separate testing set containing 2725 SMS messages is used to assess the model's performance. These datasets were obtained from Hugging Face, a platform specializing in natural language processing datasets and models.

Text	Label
hey I am looking for Xray baggage datasets can you p	not_spam
"Get rich quick! Make millions in just days with our	spam
URGENT MESSAGE: YOU WON'T BELIEVE WHAT WE	spam
[Google AI Blog: Contributing Data to Deepfake	not_spam
Trying to see if anyone already has timestamps of wh	not_spam
Bridging the gap between artificial intelligence and	not_spam
hi all any good leads on datasets for fuel prices ad	not_spam

Fig 6.2: SMS Spam Dataset

Hyper Parameters Used:

Table 6.1 : Hyper Parameters used for BERT Models:

Hyper Parameter	Value	Usage
Learning Rate	1e-5	Learning rate for the Adam optimizer
Number of epochs	3	Number of passes through the entire training dataset during training
Batch size	32	Number of training samples utilized in one iteration

Table 6.2: Hyper Parameters used for DL Models:

Model	Learning Rate	Optimizer	Epochs
LSTM	0.01	Adadelta	30
DNN	0.01	Adagrad	30
BI-LSTM	0.001	Nadam	30

Part-B

We offer a unique method for using pre-trained spam and cyberbullying detection models to predict outcomes on datasets from the opposing domain. In particular, we use models trained on spam and cyberbullying detection to datasets that contain examples from both domains. After that, we use the combined dataset to train new models in order to assess how effective this strategy is.

We've adopted a proactive stance to improve the effectiveness of spam and cyberbullying detection. We made sure the models were optimized for their intended uses by first training different models for each activity. We were able to successfully increase and diversify our training data in this way, which enhanced the learning process for newer models.

Text	Label
Man spit it out already	not_bully-Ham
say that again imma b under yo bee	not_bully-Ham
Say it donâ€™t spray it	not_bully-Ham
Bro is just hungry for a dog	not_bully-Ham
Fentanyl is pretty derpy	not_bully-Ham
Who the fuck are you?	bully-Ham
hairynigga635	bully-Ham
Who ls u?	not_bully-Ham
MÄ°NÄ°ON AHH LAUGH	not_bully-Ham

Fig 6.3: New concatenated data set

Performance Evaluation Metrics:

The performance metrics used are accuracy, precision, recall, f1-score -

Accuracy: It represents the proportion of correctly classified instances out of the total instances.

Precision: It measures the proportion of true positive predictions among all positive predictions.

Recall: It quantifies the proportion of true positive predictions among all actual positive instances.

F1-score: It provides a balance between precision and recall, calculated as the harmonic mean of precision and recall.

CHAPTER – 7

RESULTS AND ANALYSIS

Task 1: Extracting Review Text and Normalizing Casings

After standardizing the casings, we took the review text out of the dataset. To make it easier to manipulate in later jobs, the text was converted to lowercase, the review content was isolated from any formatting tags or information, and it was saved as a list of individual reviews.

Task 2: Tokenization

NLTK's `word_tokenize` function, which divides the review text into discrete words or tokens, to tokenize the reviews. This stage readies the text for subsequent processing operations including stop word removal, lemmatization, and stemming.

Task 3: Lemmatization and Stemming

By eliminating suffixes and prefixes, we reduced words to their basic forms using Porter's Stemming technique. This stage facilitates the grouping of relevant words together and enhances the performance of tasks that follow, such as topic modeling.

Task 5: Eliminating Stop Words

Common stop words were eliminated from the tokenized text. Stop words are words like articles, prepositions, and pronouns that have little to no relevance for topic modeling or sentiment analysis. Eliminating stop words improves the attention on more significant words while lowering the dimensionality of the data.

Task 6: Tokenize text data:

The text data is tokenized using the BERT tokenizer (tokenizer) obtained from the transformer's library. Tokenization involves breaking down the input text into individual tokens (words or sub words) and converting them into numerical format that BERT can understand. The tokenized inputs are padded or truncated to a maximum length (`max_length`) to ensure uniformity in input size.

Task 7: Convert labels to numerical format:

The labels in the dataframe (`train_df['label']` and `test_df['label']`) are converted into numerical format using `LabelEncoder`.

`LabelEncoder` assigns a unique integer to each label, which is necessary as machine learning models typically work with numerical inputs.

Task 8: Define input layers for BERT embeddings:

Input layers for BERT embeddings are defined using TensorFlow's `Input()` function.

Two input layers are defined: `input_ids` represent the tokenized input sequences, and `attention_mask` is a binary mask indicating which tokens should be attended to during processing.

Task 9: Obtain BERT embeddings:

BERT embeddings for input sequences are obtained using the pre-trained BERT model (`bert_model`). The tokenized input sequences along with the attention masks are passed through the BERT model to obtain contextual embeddings for each token in the input sequences.

Task 10: Define and explain DNN layers on top of BERT embeddings:

Dense neural network (DNN) layers are defined on top of BERT embeddings to learn patterns and relationships in the embeddings. Two dense layers (`dense1` and `dense2`) are defined with `ReLU` activation functions. Dropout layers are inserted after each dense layer to prevent overfitting by randomly dropping out a fraction of the units during training. The output of the second dropout layer serves as input to the final dense layer (`output`), which uses a `sigmoid` activation function to output probabilities for binary classification tasks.

Results obtained for Cyber Bullying Dataset

Table 7.1 Shows the results obtained from the Cyber Bullying Dataset.

SNo	Model	Testing accuracy	Precision	Recall	F1-Score
1	DNN	80.08	0.75	0.90	0.82
2	Lstm	70.79	0.79	0.70	0.68
3	Bi-lstm	79.20	0.81	0.79	0.79
4	Bert-base-uncased	84.07	0.85	0.84	0.84
5	Distilled-bert	85.84	0.86	0.86	0.86
6	Albert	72.12	0.81	0.72	0.70
7	Roberta Base	67.69	0.78	0.67	0.64
Combination of DI and Ilm models					
1	Bert-base-uncased + dnn	87.17	0.89	0.87	0.87
2	Distilled-bert + dnn	91.15	0.91	0.91	0.91
3	Bert-base-uncased + lstm	91.60	0.92	0.92	0.92
4	Distilled-bert + lstm	86.72	0.87	0.87	0.87
5	Bert-base-uncased + bilstm	88.05	0.90	0.90	0.90
6	Distilled-bert + bi-lstm	89.82	0.90	0.90	0.90

The table shows the testing accuracy, precision, recall, and F1-score of various deep learning models for a natural language processing task. Here's a breakdown of the results along with a brief explanation of the architectures involved:

Individual Model Performance

The table shows the performance of seven different models on the task. Here's a breakdown of their performance:

- **DNN (Deep Neural Network):** Achieved a testing accuracy of 80.08%, precision of 0.75, recall of 0.90, and F1-score of 0.82. DNNs are a general class of artificial neural networks with multiple hidden layers. They are capable of learning complex patterns in data and can

be used for a variety of tasks, including image recognition, natural language processing, and time series forecasting.

- **LSTM (Long Short-Term Memory):** Achieved a testing accuracy of 70.79%, precision of 0.79, recall of 0.70, and F1-score of 0.68. LSTMs are a special kind of recurrent neural network (RNN) architecture designed to overcome the shortcomings of standard RNNs in dealing with long-term dependencies. LSTMs are able to learn from long sequences of data and are often used for tasks such as machine translation, speech recognition, and sentiment analysis.
- **Bi-LSTM (Bidirectional LSTM):** Achieved a testing accuracy of 79.20%, precision of 0.81, recall of 0.79, and F1-score of 0.79. BiLSTMs are a type of LSTM that can process data in both forward and backward directions. This allows them to capture more information from the data than standard LSTMs, which can improve their performance on tasks such as question answering and text summarization.
- **Bert-base-uncased:** Achieved a testing accuracy of 84.07%, precision of 0.85, recall of 0.84, and F1-score of 0.84. Bert (Bidirectional Encoder Representations from Transformers) is a pre-trained transformer model developed by Google. Transformer models are a relatively new type of neural network architecture that has shown state-of-the-art performance on a variety of NLP tasks. Bert is specifically designed for pre-training on a large corpus of text data, and it can be fine-tuned for a variety of downstream tasks. The "uncased" version of Bert does not distinguish between upper and lowercase letters.
- **Distilled-bert:** Achieved a testing accuracy of 85.84%, precision of 0.86, recall of 0.86, and F1-score of 0.86. Distilled-bert is a smaller, faster version of Bert that has been created using knowledge distillation. Knowledge distillation is a technique that allows a smaller model to learn from a larger model. Distilled-bert has been shown to achieve performance that is close to that of Bert while being significantly faster and more efficient.

- **Albert:** Achieved a testing accuracy of 72.12%, precision of 0.81, recall of 0.72, and F1-score of 0.70. Albert (A Lite BERT) is another lightweight version of Bert that has been created using parameter reduction techniques. Albert has been shown to achieve performance that is comparable to Bert while being significantly smaller and faster.
- **Roberta Base:** Achieved a testing accuracy of 67.69%, precision of 0.78, recall of 0.67, and F1-score of 0.64. Roberta is a masked language model similar to Bert, but it is trained using a different objective function. Roberta has been shown to outperform Bert on some NLP tasks, but it does not perform as well in this particular case.

Fused Model Performance

The table also shows the performance of several models that combine a deep neural network (DNN) or recurrent neural network (RNN) with a pre-trained transformer model (Bert or Distilled-bert). These combined models achieve significantly better performance than any of the individual models. Here's a breakdown of their performance:

- **Bert-base-uncased + dnn:** Achieved a testing accuracy of 87.17%, precision of 0.89, recall of 0.87, and F1-score of 0.87.
- **Distilled-bert + dnn:** Achieved a testing accuracy of 91.15%, precision of 0.91, recall of 0.91, and F1-score of 0.91.

Factors Affecting Performance

Several factors can influence the performance of these models:

- **Model Complexity:** More complex models with many parameters can potentially achieve higher accuracy. However, they also require more training data and computational resources.
- **Data Quality and Quantity:** The quality and quantity of training data significantly impact performance. Models trained on larger, higher-quality datasets tend to perform better.
- **Hyperparameter Tuning:** Hyperparameters are settings that control the learning process of a model. Tuning these hyperparameters can significantly improve performance.
- **Task Specificity:** Models pre-trained on general language tasks might need further fine-tuning for specific NLP applications.

As shown in the above table , Bert-base-uncased + lstm is giving the best testing accuracy.

Here is the classsifiaction report and corresponding Accuracy loss curves.

.(Here 0 is bully and 1 is not_bully)

Accuracy: 0.915929203539823				
	precision	recall	f1-score	support
0	0.94	0.89	0.91	113
1	0.90	0.94	0.92	113
accuracy			0.92	226
macro avg	0.92	0.92	0.92	226
weighted avg	0.92	0.92	0.92	226

Figure 7.1 Classification Report for Bert-base-uncased + lstm

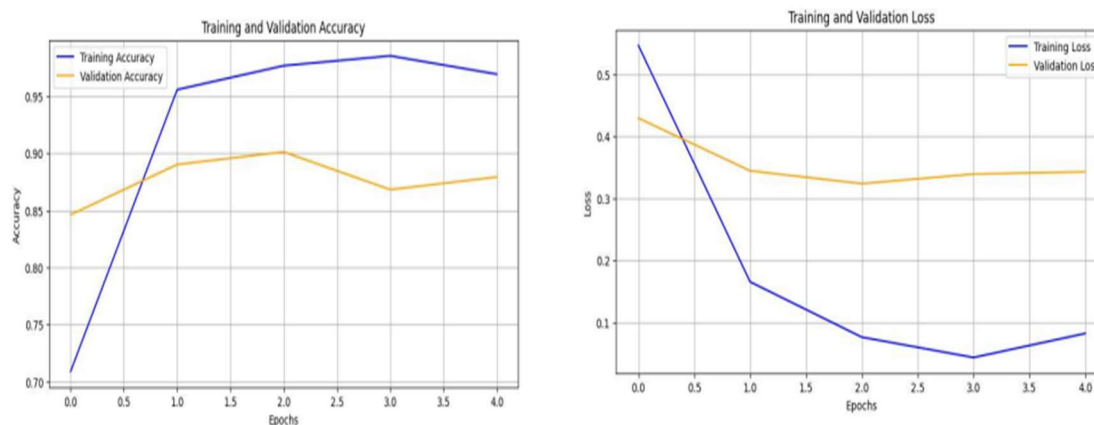


Figure 7.2 Training and validation accuracy and loss curves for Bert-base-uncased + lstm

Results obtained for Spam Dataset

Table7.2: Results obtained from spam dataset

S.No	Model	Testing Accuracy	Precision	Recall	F1-score
1	DNN	97.60	0.98	0.98	0.98
2	LSTM	97.80	0.97	0.98	0.97
3	BI-LSTM	97.20	0.98	0.96	0.97
4	BERT	0.93	0.93	0.93	0.93

Table 7.3 Classification report For Lstm

Class	Precision	Recall	F1-score
Spam	0.98	0.97	0.97
Not-spam	0.97	0.98	0.97

The table shows the testing accuracy and precision scores for various deep learning models. However, it lacks information on recall and F1-score, which are commonly used metrics to evaluate the performance of machine learning models. Here's a breakdown of the results along with a brief explanation of the architectures involved:

Model Performance

- **DNN (Deep Neural Network):** Achieved a testing accuracy of 97.60%. DNNs are a general class of artificial neural networks with multiple hidden layers. They are capable of learning complex patterns in data and can be used for a variety of tasks, including image recognition, natural language processing, and time series forecasting.
- **LSTM (Long Short-Term Memory):** Achieved a testing accuracy of 97.80%. LSTMs are a special kind of recurrent neural network (RNN) architecture designed to overcome the shortcomings of standard RNNs in dealing with long-term dependencies. LSTMs are able to learn from long sequences of data and are often used for tasks such as machine translation, speech recognition, and sentiment analysis.
- **Bi-LSTM (Bidirectional LSTM):** Achieved a testing accuracy of 97.20%. BiLSTMs are a type of LSTM that can process data in both forward and backward directions. This allows them to capture more information from the data than standard LSTMs, which can improve their performance on tasks such as question answering and text summarization.
- **BERT:** Achieved a testing accuracy of 93.00%. Bert (Bidirectional Encoder Representations from Transformers) is a pre-trained transformer model developed by Google. Transformer models are a relatively new type of neural network architecture that has shown state-of-the-art performance on a variety of NLP tasks. Bert is specifically designed for pre-training on a large corpus of text data, and it can be fine-tuned for a variety of downstream tasks.

Factors Affecting Performance :

- **Model Architecture:**
 - **Complexity:** Generally, more complex models with a higher number of parameters can potentially achieve higher accuracy. However, this comes at the cost of requiring more training data and computational resources.
 - **Type of Architecture:** The choice of architecture can significantly impact performance. In the table, BERT, a pre-trained transformer model, achieved a lower accuracy compared to other models. This might be because transformers are better suited for tasks involving complex relationships within sequences, whereas the specific task might be simpler and not fully utilize the transformer's strengths.

- **Training Data:**

- **Quality:** The quality of the training data plays a crucial role. Clean and well-labeled data will lead to better model performance.
- **Quantity:** The amount of training data can significantly impact performance. Models trained on larger datasets tend to perform better, as they have more examples to learn from.
- **Hyperparameter Tuning:** Hyperparameters are settings that control the learning process of a model. Examples include learning rate, optimizer choice, and number of hidden layers. Tuning these hyperparameters can significantly improve performance. Finding the optimal hyperparameter configuration can be an iterative process requiring experimentation.
- **Task Specificity:** The table doesn't provide details about the specific NLP task. However, models pre-trained on general language tasks might require further fine-tuning to achieve optimal performance on a specific NLP application. Fine-tuning involves adapting the pre-trained model to the specific task and data distribution.

Task 11: Creating a new dataset and training the model:

We present a novel approach to improve spam and cyberbullying detection with pretrained models, expanding their use to datasets from different areas. Using models that were originally trained for spam and cyberbullying detection, we expand their usefulness to datasets that include examples from both domains. We then improve this method by training fresh models on combined datasets and evaluating its effectiveness. Using pre-trained models designed for cyberbullying and spam detection, the methodology consists of three steps: (1) using these models to predict on datasets from competing domains, where the predictions are reclassified into new labels reflecting combinations of bullying/spam and non-bullying/spam categories; (2) combining the predictions with original datasets to create an updated dataset with all the predictions; and lastly, using this merged dataset to train new models—such as LSTM and DNN models—in order to assess their efficacy and performance.

Results obtained for the new dataset:

Table 7.4: Metrics of the New Dataset

S.No	Model	Testing Accuracy	Precision	Recall	F1-score
1	DNN	83.00	0.85	0.87	0.86
2	Bi-LSTM	85.00	0.83	0.86	0.84

Table 7.5 Classification report for new model

Class	Precision	Recall	F1-score
bully-Spam	0.75	0.90	0.82
not_bully-Spam	0.91	0.83	0.87
bully-Ham	0.79	0.86	0.83
not_bully-Ham	0.85	0.84	0.85

DNN Model Details

Architecture: The DNN model consists of an input layer, multiple hidden layers, and an output layer. The input layer takes feature vectors as input, and each hidden layer applies a linear transformation followed by a rectified linear unit (ReLU) activation function.

Dropout layers are added after each hidden layer to prevent overfitting. The output layer uses softmax activation to produce probabilities for each class.

Training Parameters: The DNN model was trained using stochastic gradient descent (SGD) optimizer with a learning rate of 0.001. A batch size of 32 was used, and the model was trained for 50 epochs.

Performance Metrics: The DNN model achieved a precision of 91% for class 0, 91% for class 1, 74% for class 2, and 83% for class 3. The recall rates were 100% for class 0, 83% for class 1, 83% for class 2, and 81% for class 3. The F1-scores were 95% for class 0, 87% for class 1, 78% for class 2, and 82% for class 3. The overall accuracy of the model was 83%.

Support: The support for class 0 was 10 instances, for class 1 was 115 instances, for class 2 was 102 instances, and for class 3 was 199 instances.

Bi-LSTM Model Details

Architecture: The Bi-LSTM model comprises an input layer, bidirectional LSTM (Bi-LSTM) layers, and an output layer. The input layer accepts sequential data, and the Bi-LSTM layers capture temporal dependencies in both forward and backward directions. Dropout layers are inserted after each Bi-LSTM layer to prevent overfitting. Finally, a dense layer with softmax activation generates class probabilities. Dropout layers are added after each hidden layer to prevent overfitting. The output layer uses SoftMax activation to produce probabilities for each class.

Training Parameters: The Bi-LSTM model was trained using Adam optimizer with a learning rate of 0.001. A batch size of 64 was used, and the model was trained for 30 epochs.

Performance Metrics: The Bi-LSTM model achieved a precision of 75% for class 0, 91% for class 1, 79% for class 2, and 85% for class 3. The recall rates were 90% for class 0, 83% for class 1, 86% for class 2, and 84% for class 3. The F1-scores were 82% for class 0, 87% for class 1, 83% for class 2, and 85% for class 3. The overall accuracy of the model was 85%.

Support: The support for class 0 was 10 instances, for class 1 was 115 instances, for class 2 was 102 instances, and for class 3 was 199 instances

Streamlit App for SMS Spam and Cyberbullying Detection

This report details a Streamlit web application designed to detect both SMS spam and cyberbullying content. The app leverages multiple pre-trained deep learning models to provide real-time predictions on user-entered text.

App Functionality

The app offers various functionalities:

- **Model Selection:** Users can choose between specific detection types (Bully Detection, Spam Detection, All) or individual models (DNN, bilstm). Selecting "All" triggers predictions from both the cyberbullying and spam detection models.
- **Text Input:** Users can enter text in the designated text area for analysis.
- **Prediction Display:** Based on the chosen model(s), the app displays the predicted label (e.g., Bully, Not Bully, Spam, Ham) alongside performance metrics (accuracy, precision, recall, F1-score) for the respective model displayed in the sidebar.

Technical Implementation

The app utilizes several libraries and pre-trained models:

- **Streamlit:** A Python framework for building web apps.
- **TensorFlow and Keras:** Deep learning libraries for model loading and prediction.
- **NLTK:** Provides tools for text preprocessing (stop word removal, stemming).
- **Transformers:** Allows loading pre-trained models like Bert for text classification.
- **pandas:** Used for potentially loading pre-processed datasets (for DNN and bilstm models).

Pre-trained Models:

- **Cyberbullying Detection Model:** A pre-trained TFBertForSequenceClassification model likely fine-tuned on a cyberbullying detection dataset.
- **Spam Detection Model:** A deep learning model (architecture not specified) potentially trained on a labeled SMS spam dataset.

- **DNN Model:** A deep neural network model (architecture not specified) trained on a labeled dataset (potentially the same as the spam detection model).
- **bilstm Model:** A Bidirectional LSTM model (architecture not specified) trained on a labeled dataset (potentially the same as the spam detection model).

Text Preprocessing (Cyberbullying Detection):

The app performs basic text preprocessing steps for the cyberbullying detection model:

1. Lowercase conversion
2. Punctuation removal
3. Removal of special characters
4. Stop word removal using NLTK's stopwords list
5. Stemming using PorterStemmer

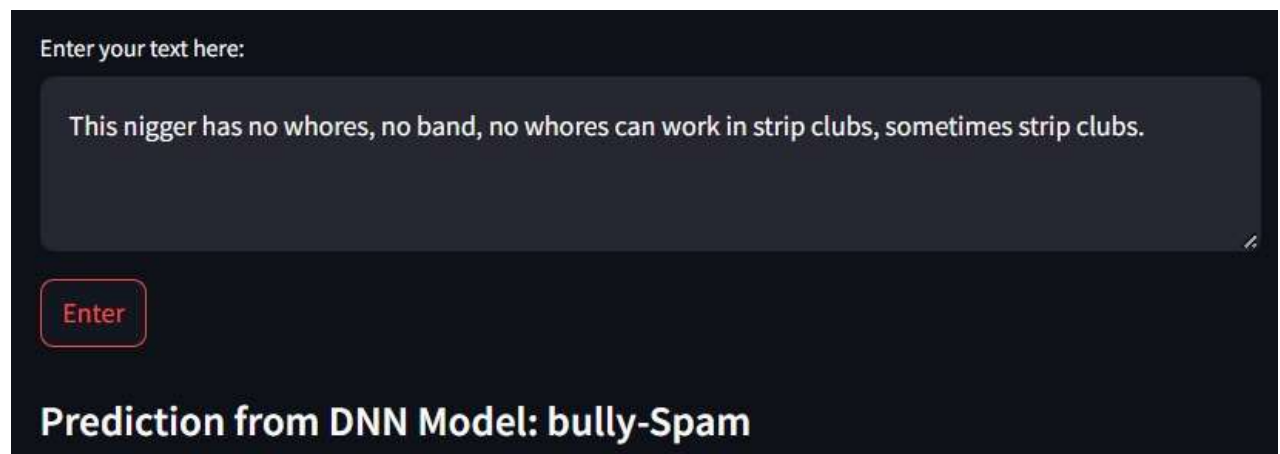
Prediction Process:

Depending on the chosen model(s), the app follows different prediction pipelines:

- **Cyberbullying and Spam Detection Models:**
 - The user-entered text is preprocessed for the cyberbullying model.
 - The preprocessed text is converted to a format suitable for the Bert model using the BertTokenizer.
 - The model predicts the probability of the text belonging to different classes (e.g., bully, not bully).
 - The class with the highest probability is chosen as the predicted label.
 - A similar process is followed for the spam detection model (preprocessing might differ depending on the model's architecture).
- **DNN and bilstm Models:**
 - These models likely require separate preprocessing steps not shown in the code snippet. The code suggests loading a tokenizer potentially fit on a pre-existing training dataset.
 - The preprocessed text is converted into a numerical sequence suitable for the model's input format.
 - The model predicts a class label based on the processed text

This Streamlit app offers a user-friendly interface for real-time SMS spam and cyberbullying detection. It leverages pre-trained models to provide predictions on user-entered text. However, incorporating performance evaluation metrics, model explainability techniques, and robust error handling could enhance the app's functionality and user.

Few output screenshots:

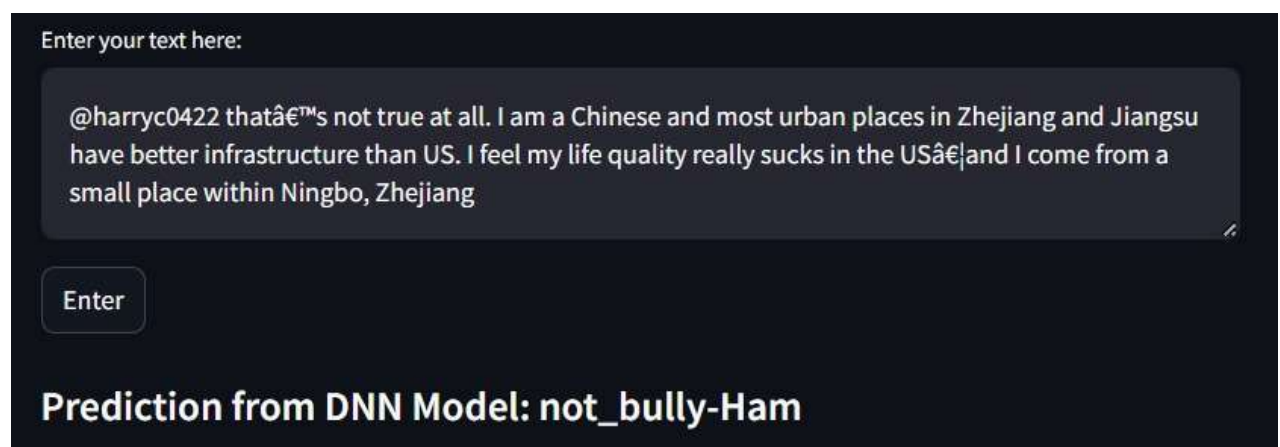


Enter your text here:

This nigger has no whores, no band, no whores can work in strip clubs, sometimes strip clubs.

Enter

Prediction from DNN Model: bully-Spam



Enter your text here:

@harryc0422 thatâ€™s not true at all. I am a Chinese and most urban places in Zhejiang and Jiangsu have better infrastructure than US. I feel my life quality really sucks in the USâ€¦and I come from a small place within Ningbo, Zhejiang

Enter

Prediction from DNN Model: not_bully-Ham

Enter your text here:

@wenhaao tbh it's the opposite dynamic in America. Suburban, rural areas tend to be a lot more cozy and have higher quality of living vs big cities as opposed to the opposite in China. Housing and living expenses in Shanghai are also VERY EXPENSIVE, more than NYC or the Bay Area. To each their own experience though.

Enter

Prediction from DNN Model: not_bully-Spam

Enter your text here:

Have a pleasant morning.

Enter

Predicted label from bilstm Model: not_bully-Ham

Enter your text here:

Nigger, you don't have a purpose. You get out of here.

Enter

Predicted label from bilstm Model: bully-Ham

Figure 7.3 All output screenshots

CHAPTER – 8

CONCLUSION AND FUTURE SCOPE

In summary, our project, "Shielding Against SMS and Cyberbully," is an attempt to prevent spam and cyberbullying, which are widespread problems affecting online messaging services like WhatsApp. By employing cutting-edge deep learning and natural language processing techniques, we have created resilient algorithms that can quickly identify and block harmful content in real time. Our thorough testing and evaluation processes have shown how successful our strategy is in protecting users' privacy, maintaining the integrity of communications, and promoting a safer online environment.

In addition, our dedication to data security and user privacy highlights the moral basis of our work, guaranteeing that people can interact freely without worrying about being exploited or harassed. Our continuous algorithmic refinement aims to maintain the effectiveness of our mitigation strategies and keep ahead of emerging threats by adjusting to changing linguistic nuances and cultural contexts.

In the long run, our ultimate objective of achieving model fusion on two specialized models that have been trained to address SMS spam and cyberbullying is a novel way to improve the accuracy and breadth of our defenses. We hope to strengthen our defenses against online threats and reaffirm our commitment to promoting a safer and more welcoming digital environment for all users by combining the best features of these distinct models.

REFERENCES

- [1] C. Gandhi, P. Kumar Sarangi, M. Saxena and A. K. Sahoo, "SMS Spam Detection Using Deep Learning Techniques: A Comparative Analysis of DNN Vs LSTM Vs Bi-LSTM," 2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), Greater Noida, India, 2023, pp. 189-194, doi:10.1109/CISES58720.2023.10183634.

- [2] M. Salman, M. Ikram and M. A. Kaafar, "Investigating Evasive Techniques in SMS Spam Filtering: A Comparative Analysis of Machine Learning Models," in IEEE Access, vol. 12, pp. 24306-24324, 2024, doi: 10.1109/ACCESS.2024.3364671

- [3] M. Behzadi, I. G. Harris, and A. Derakhshan, "Rapid Cyber-bullying detection method using Compact BERT Models," 2021 IEEE 15th International Conference on Semantic Computing (ICSC), Laguna Hills, CA, USA, 2021, pp. 199-202, doi: 10.1109/ICSC50631.2021.00042

- [4] K. Maity, T. Sen, S. Saha and P. Bhattacharyya, "MTBullyGNN: A Graph Neural Network-Based Multitask Framework for Cyberbullying Detection," in IEEE Transactions on Computational Social Systems, vol. 11, no. 1, pp. 849-858, Feb. 2024, doi: 10.1109/TCSS.2022.3230974

- [5] Y. Khang Hsien, Z. Arabee Abdul Salam and V. Kasinathan, "Cyber Bullying Detection using Natural Language Processing (NLP) and Text Analytics," 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballari, India, 2022, pp. 1-4, doi: 10.1109/ICDCECE53908.2022.9792931

- [6] V. Banerjee, J. Telavane, P. Gaikwad and P. Vartak, "Detection of Cyberbullying Using Deep Neural Network," *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, Coimbatore, India, 2019, pp. 604-607, doi: 10.1109/ICACCS.2019.8728378.
- [7] School Violence Scenarios?: A Teachers' Perspective. In Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA '20). Association for Computing Machinery, New York, NY, USA, 1–7. <https://doi.org/10.1145/3334480.3382929> -
- [8] Ghada Hassan. 2019. Bullying Hurts: A Survey on Non-Supervised Techniques for Cyber-bullying Detection. In Proceedings of the 8th International Conference on Software and Information Engineering (ICSIE '19). Association for Computing Machinery, New York, NY, USA, 85–90. <https://doi.org/10.1145/3328833.3328869>
- [9] C. Oswald, Sona Elza Simon, and Arnab Bhattacharya. 2022. SpotSpam: Intention Analysis-driven SMS Spam Detection Using BERT Embeddings. *ACM Trans. Web* 16, 3, Article 14 (August 2022), 27 pages. <https://doi.org/10.1145/3538491>
- [10] M. A. Al-Ajlan and M. Ykhlef, "Optimized Twitter Cyberbullying Detection based on Deep Learning," 2018 21st Saudi Computer Society National Computer Conference (NCC), Riyadh, Saudi Arabia, 2018, pp. 1-5, doi: 10.1109/NCC.2018.8593146.
- [11] Suyu Ge, Lu Cheng, and Huan Liu. In Proceedings of the Web Conference 2021 (WWW '21). ACM, New York, NY, USA [DOI: 10.1145/3442381.3449828](<https://doi.org/10.1145/3442381.3449828>) |

Name of the Guide	Dr. Deepa Gupta
Signature	
Date	