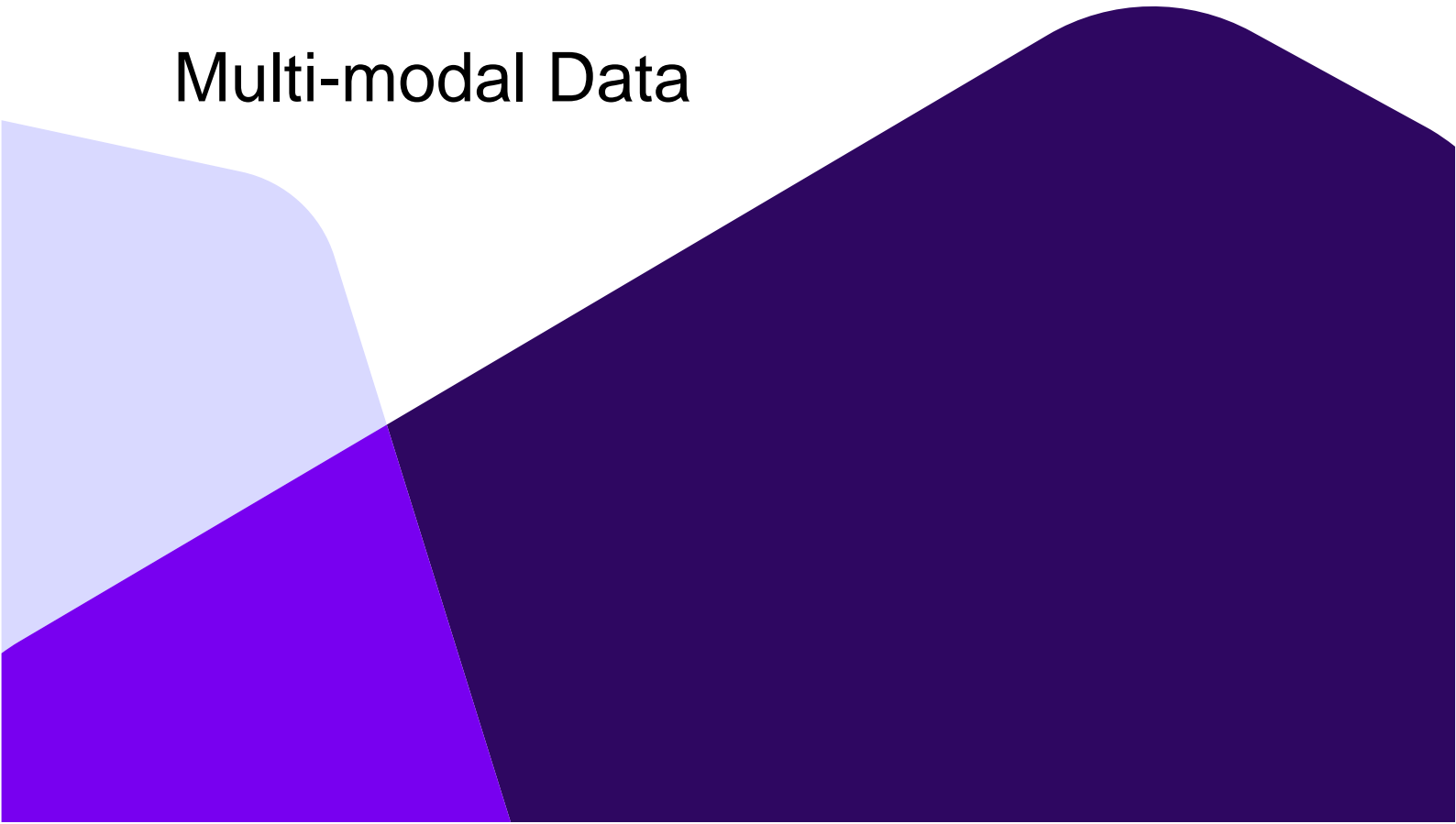


Israt Tabassum / candidate number: 8505

# A Hybrid Deep-Learning Approach for Multi-class Classification of Cyberbullying using Social Medias' Multi-modal Data



**University of South-Eastern Norway**

Faculty of Technology, Natural Sciences, and Maritime Sciences

Department of Science and Industry systems

PO Box 235

NO-3603 Kongsberg, Norway

<http://www.usn.no>

© [2024] [Israt Tabassum]

This thesis is worth 60 study points



**University of  
South-Eastern Norway**

# **A Hybrid Deep-Learning Approach for Multi-class Classification of Cyberbullying using Social Medias' Multi-modal Data**

**Master's Thesis in Computer Science**

**Israt Tabassum**

**Academic Supervisor**

Vimala Nunavath

**University of South-Eastern Norway**

Department of Science and Industry Systems

Faculty of Technology, Natural Sciences, and Maritime Sciences

Campus Kongsberg

May 2024

# Summary

Social media sites like Facebook, Instagram, Twitter, LinkedIn, and Facebook are important channels for content creation and distribution that have a big impact on business, politics, and interpersonal relationships. People like to spend their free time using social media by uploading their pictures, post, videos for sharing their daily activities with other people, and view other peoples activities. Due to their concise and captivating format, short videos have become more and more popular on these platforms recently. However, they frequently receive comments known that are mixed positive and negative and take the form of text, images, and multimodal data as memes. This makes it more difficult to recognize and deal with instances of cyberbullying. The problem like cyberbullying, a serious problem where victims of abusive online communication can experience despair, anxiety, and loneliness. Many studies have been conducted on the classification of cyberbullying. However, the majority of these studies concentrated on binary classification on multi-modal data or multi-classification on textual data. Despite significant advancements in deep learning techniques for cyberbullying classification, there was a gap in the multi-class classification of cyberbullying using multimodal data. The goal of this thesis was to close this gap by accurately classifying cyberbullying across multi-modal data types using a hybrid (RoBERTa+ViT) deep learning approach that combined models, Vision Transformer (ViT) for images and RoBERTa for text of the multi-modal data.

Two datasets were used in this thesis to classify cyberbullying: a private dataset that was collected from comments on social media videos and a public dataset that was downloaded from existing research. In this thesis, three sets of experiments were conducted for multi-class classification of cyberbullying. The first set of experiments were done on using text data by deep learning models such as LSTM, GRU, RoBERTa, BERT, DistilBERT, and Hybrid (CNN+LSTM) model for public data and RoBERTa model for private dataset, the second set of experiments were done on using image data by using deep learning models such as ResNET-50, CNN and ViT model for public dataset and ViT model for private dataset, and the last set of experiments were performed on using multi-modal data (i.e., memes) of both public and private dataset using hybrid deep learning models such as Hybrid (RoBERTa+ViT) model.

Using the public dataset, we trained nine deep learning models: ResNET-50, CNN and ViT for image data, and LSTM, GRU, RoBERTa, BERT, DistilBERT, and Hybrid (CNN+LSTM) model for textual data. The experimental results showed that the ViT model obtained an accuracy of 99.5%, F1-score of 0.995, for multi-class classification on image data. Whereas RoBERTa model performed better when compared to other models on textual data with an accuracy of 99.2% and F1-score of 0.992. With this outcome,

for private data, RoBERTa model for text data and ViT model for image data were developed. As a result, the RoBERTa model attained F1-score of 0.986 and an accuracy of 98.6%. Whereas, for image data, the ViT model achieved F1-score of 0.9319 and an accuracy of 93.20%. For multi-modal data, a hybrid model with a late fusion module (Roberta+ViT) was developed that combined RoBERTa and ViT model to classify the multi-class classification of cyberbullying and attained an accuracy of 99.24%, and 96.01% and F1-score 0.992, and 0.9599 respectively.

From the obtained results, it can be concluded that deep learning models like RoBERTa and Vision Transformer (ViT) models are very effective for classifying various forms of cyberbullying. RoBERTa works well with text, producing nearly perfect results, whereas ViT is particularly strong at handling images. Furthermore, when these models were combined into a hybrid (RoBERTa+ViT) model, they became even more effective at classifying cyberbullying in multi-modal data, such as memes.

**Keywords**

"Cyberbullying", "Multi-modal data", "Multi-Class Classification", "Deep-Learning", and "Social-media"

## **Acknowledgements**

I would like to sincerely thank my supervisor, Vimala Nunavath, for her important advice and assistance. Her guidance helped me overcome my obstacles and formed the basis of my research. Lastly, I would want to express my sincere gratitude to all of my USN instructors and faculty.

# Contents

<b>1</b>	<b>Introduction</b>	<b>17</b>
1.1	Motivation and Problem Statement . . . . .	18
1.2	Thesis Goals . . . . .	19
1.3	Research Questions . . . . .	19
1.4	Research Approach . . . . .	21
1.5	Assumptions and Limitations . . . . .	22
1.5.1	Assumptions . . . . .	23
1.5.2	Limitations . . . . .	23
1.6	Thesis Contributions . . . . .	23
1.7	Thesis Outline . . . . .	24
<b>2</b>	<b>Background</b>	<b>25</b>
2.1	Cyberbullying Identification and Classification . . . . .	25
2.1.1	Societal Effects of Cyberbullying . . . . .	25
2.1.2	Role of Technology in Social Media Monitoring . . . . .	26
2.1.3	Cyberbullying Identification and Classification by Machine Learning . . . . .	26
2.1.4	Cyberbullying Identification and Classification by Deep Learning . . . . .	27
2.2	Data Classification . . . . .	28
2.3	Deep-Learning Models for Multi-Modal Data . . . . .	28
2.3.1	The LSTM Model . . . . .	29
2.3.2	The GRU Model . . . . .	30
2.3.3	The RoBERTa Model . . . . .	30
2.3.4	The CNN Model . . . . .	32
2.3.5	The ViT Model . . . . .	32
2.3.6	The Hybrid(CNN-LSTM) Model . . . . .	33
2.3.7	The BERT Model . . . . .	34
2.3.8	The DistilBERT Model . . . . .	34
2.3.9	Hyperparameters and Hyperparameter Tuning . . . . .	35
2.3.10	Performance Metrics for Classification . . . . .	36
2.4	Software Languages and Tools for Model Deployment . . . . .	38
2.4.1	Hypertext Markup Language . . . . .	39
2.4.2	Cascading Style Sheets . . . . .	39
2.4.3	JavaScript . . . . .	39
2.4.4	Python Programming Language . . . . .	40

2.4.5	Google Colaboratory . . . . .	40
2.4.6	Visual Studio Code . . . . .	40
<b>3</b>	<b>Literature Review</b>	<b>42</b>
3.1	Existing Research on Applying DL for Cyberbullying Binary-class Classification . . . . .	42
3.2	Existing Research on Applying DL for Multi-Classification . . . . .	49
3.2.1	Identifying Gap of The Research on Cyberbullying of Social Media . . . . .	52
<b>4</b>	<b>Research Methodology</b>	<b>54</b>
4.1	Datasets . . . . .	54
4.1.1	Public Dataset Collection . . . . .	54
4.1.2	Private Dataset Collection . . . . .	55
4.2	Data pre-processing . . . . .	56
4.2.1	Data Preprocessing for Textual Data . . . . .	57
4.2.2	Data preprocessing for Images . . . . .	59
4.2.3	Multi-Modal Data Preprocessing . . . . .	62
4.2.4	Feature Extraction . . . . .	64
4.2.5	Split Data into Training and Validation Datasets . . . . .	66
4.3	Proposed Solution . . . . .	66
4.4	Network Architectures for Text Data Classification . . . . .	68
4.4.1	Hybrid(CNN+LSTM) Model . . . . .	68
4.4.2	LSTM Model . . . . .	69
4.4.3	GRU Model . . . . .	70
4.4.4	BERT Model . . . . .	71
4.4.5	DistilBERT Model . . . . .	72
4.4.6	RoBERTa Model . . . . .	73
4.5	Network Architectures for Image Data Classification . . . . .	74
4.5.1	ResNet-50 Model . . . . .	74
4.5.2	CNN Model . . . . .	75
4.5.3	ViT Model . . . . .	76
4.6	Network Architecture for Multi-Modal Data Classification . . . . .	77
4.6.1	Late Fusion Module . . . . .	79
4.7	Models Deployment and Evaluation . . . . .	81
4.7.1	Models Testing . . . . .	81
4.7.2	Models Deployment . . . . .	81
4.8	Computing Resources . . . . .	84



4.8.1	Hardware Configuration . . . . .	84
4.8.2	Software and Libraries . . . . .	84
<b>5</b>	<b>Experiments and Results</b>	<b>85</b>
5.1	Experimental Setup for Multi-class Classification . . . . .	85
5.1.1	For text data classification . . . . .	85
5.1.2	Hyperparameter tuning for text data . . . . .	85
5.1.3	For image data classification . . . . .	85
5.1.4	Hyperparameter tuning for image data . . . . .	86
5.1.5	For multi-modal data classification . . . . .	86
5.1.6	Hyperparameter tuning for multi-modal data . . . . .	87
5.2	Results . . . . .	87
5.2.1	Experimental Results on The Public Dataset . . . . .	87
5.2.2	Experimental Results on Private Dataset . . . . .	102
5.3	Model Deployment . . . . .	107
<b>6</b>	<b>Discussion</b>	<b>113</b>
6.1	Discussing Results for each Research Question . . . . .	113
6.2	Verifying our Results with the Private Dataset . . . . .	115
6.3	Comparison with Existing Literature . . . . .	116
<b>7</b>	<b>Conclusion and Future Work</b>	<b>118</b>
7.1	Conclusion . . . . .	118
7.2	Future Work . . . . .	119
	<b>References</b>	<b>120</b>
	<b>Appendices</b>	<b>130</b>
<b>A</b>	<b>Appendix</b>	<b>131</b>
A.1	Performance Evaluation of Public Data for Each Classes . . . . .	131
A.1.1	Performance Evaluation of Textual Data . . . . .	131
A.1.1.1	Experiment with hybrid (CNN+LSTM) model for each classes . . . . .	131
A.1.1.2	Experiment with LSTM model for each classes . . . . .	132
A.1.1.3	Experiment with GRU model for each classes . . . . .	133
A.1.1.4	Experiment with BERT model for each classes . . . . .	134
A.1.1.5	Experiment with DistilBERT model for each classes . . . . .	135
A.1.1.6	Experiment with RoBERTa model for each classes . . . . .	136

A.1.2	Performance Evaluation of Image Data . . . . .	137
A.1.2.1	Experiment with ResNet model for each classes . . . . .	137
A.1.2.2	Experiment with CNN model for each classes . . . . .	138
A.1.2.3	Experiment with ViT model for each classes . . . . .	139
A.2	Performance Evaluation of Private Data for Each Classes . . . . .	140
A.2.1	Performance evaluation of text data for each classes using RoBERTa model .	140
A.2.2	Performance evaluation of text data for each classes using RoBERTa model .	141
A.3	Source Code to Replicate The Experiment . . . . .	142

## List of Figures

1	Applied Research Approach . . . . .	21
2	The Architecture of LSTM model [1]. . . . .	29
3	The Architecture of GRU model [2]. . . . .	30
4	The RoBERTa Model Architecture [3]. . . . .	31
5	CNN Architecture [4]. . . . .	32
6	ViT Architecture [5]. . . . .	33
7	Hybrid(CNN-LSTM) Model Working Process [6]. . . . .	33
8	Bert Model Architecture ( collected from [7]) . . . . .	34
9	DistilBert Model Architecture ( collected from [8]) . . . . .	34
10	Confusion Matrix in Heat Map . . . . .	37
11	An example of meme (multi-modal data) from Public Dataset . . . . .	55
12	Private Data Collection Process . . . . .	56
13	An example meme data from collected private dataset . . . . .	56
14	The Overview of the pre-processing pipeline for the both public and private datasets' text data . . . . .	57
15	Text Data before Pre-Processing . . . . .	58
16	Text Data after Pre-Processing . . . . .	59
17	The overview of the pre-processing pipeline for both public and private datasets' images	60
18	An example of Class-0 image . . . . .	61
19	An example of Class-1 image . . . . .	61
20	An example of Class-2 image . . . . .	61
21	An example of Class-3 image . . . . .	61
22	Greyscale image . . . . .	62
23	Process of Extracting Text from Images . . . . .	63
24	The of Multi-modal Data Preprocessing . . . . .	63
25	Complete Pipeline for Data Pre-processing. . . . .	64
26	Pipeline for Feature Extraction. . . . .	65
27	The proposed solution for public dataset . . . . .	67
28	The proposed solution for private dataset . . . . .	67
29	The Used CNN+LSTM Architecture for Text Data Classification . . . . .	69
30	The used LSTM Architecture for Text Data Classification . . . . .	70
31	The Used GRU Architecture for Text Data Classification . . . . .	71
32	The Used BERT Architecture for Text Data Classification . . . . .	71
33	The Used DistilBERT Architecture for Text Data Classification . . . . .	72

34	The Used RoBERTa Architecture for Text Data Classification . . . . .	74
35	The Used ResNet-50 Architecture for Image Data Classification . . . . .	75
36	The Used CNN Architecture for Image Data Classification . . . . .	76
37	The Used ViT Architecture for Image Data Classification . . . . .	77
38	Architecture of Multi-Modal Data with Hybrid Model on Late Fusion Module . . . . .	78
39	Process of Late Fusion Module . . . . .	79
40	Use Case Diagram for Model Deployment Process . . . . .	82
41	Flow Chart Diagram for Model Deployment Process . . . . .	83
42	ROC-AUC of Hybrid model for Textual Data . . . . .	88
43	Confusion Matrix of Hybrid(CNN+LSTM) model on Public Data . . . . .	89
44	ROC-AUC of LSTM model for Textual Data . . . . .	90
45	Confusion Matrix of LSTM model for Textual Data . . . . .	90
46	ROC-AUC of GRU model for Textual Data . . . . .	91
47	Confusion Matrix of GRU model for Textual Data . . . . .	92
48	ROC-AUC for BERT model . . . . .	92
49	Confusion matrix for BERT model . . . . .	93
50	ROC-AUC for DistilBERT model . . . . .	94
51	Confusion matrix for DistilBERT model . . . . .	94
52	ROC-AUC for RoBERTa model . . . . .	95
53	Confusion matrix for RoBERTa model . . . . .	96
54	ROC-AUC of ResNet model on Public Data . . . . .	97
55	Confusion Matrix of ResNet model on Public Data . . . . .	97
56	ROC-AUC of CNN model on Public Data . . . . .	98
57	Confusion Matrix of CNN model on Public Data . . . . .	99
58	ROC-AUC for ViT model on Public Data . . . . .	100
59	Confusion Matrix of ViT model on Public Data . . . . .	100
60	Result of Multi-modal Data for Public Dataset . . . . .	102
61	Confusion Matrix of Hybrid(RoBERTa+ViT) model . . . . .	102
62	ROC-AUC of RoBERTa model on Private Data . . . . .	103
63	Confusion Matrix of RoBERTa model on Private Data . . . . .	104
64	ROC-AUC for ViT model on Private Data . . . . .	105
65	Confusion Matrix of ViT model for Private Data . . . . .	105
66	Confusion Matrix of Hybrid(RoBERTa+ViT) Model for Private Data . . . . .	106
67	Result of Multi-modal Data for Private Dataset . . . . .	107
68	Cyberbullying Classification Web-Page . . . . .	107

69	Cyberbullying Classification Using The Model of Private and Public Data . . . . .	108
70	Showing The Result Description for text, image, and multi-modal data in the GUI . .	108
71	Testing Text For Class label: 1 . . . . .	109
72	Showing The Result after Uploading Input as Text of Figure:71 . . . . .	109
73	Testing Text For Class label: 3 . . . . .	109
74	Showing The Result after uploading input Image of figure:73 . . . . .	109
75	Test Image For Private Data . . . . .	110
76	Showing The Result after uploading input Image of figure:75 . . . . .	110
77	Testing The Multi-Modal Data For The Model . . . . .	110
78	Showing The Result after uploading input Image of figure:77 . . . . .	110
79	Test Image For The Model . . . . .	111
80	Showing The Result after uploading input Image of figure:79 . . . . .	111
81	Testing the Multi-modal data For The Model . . . . .	111
82	Showing The Result after uploading input multi-modal data of figure:81 . . . . .	111
83	Test Meme For The Model . . . . .	112
84	Showing The Result after uploading input Image of figure:83 . . . . .	112
85	Confusion Matrix for Class 0 for Hybrid Model . . . . .	132
86	confusion matrix of class 1 for Hybrid model . . . . .	132
87	confusion matrix of class 2 for Hybrid model . . . . .	132
88	confusion matrix of class 3 for Hybrid model . . . . .	132
89	Confusion Matrix for Class 0 for LSTM Model . . . . .	133
90	confusion matrix of class 1 for LSTM model . . . . .	133
91	confusion matrix of class 2 for LSTM model . . . . .	133
92	confusion matrix of class 3 for LSTM model . . . . .	133
93	Confusion Matrix for Class 0 for GRU Model . . . . .	134
94	confusion matrix of class 1 for GRU model . . . . .	134
95	confusion matrix of class 2 for GRU model . . . . .	134
96	confusion matrix of class 3 for GRU model . . . . .	134
97	Confusion Matrix for Class 0 for BERT Model . . . . .	135
98	confusion matrix of class 1 for BERT model . . . . .	135
99	confusion matrix of class 2 for BERT model . . . . .	135
100	confusion matrix of class 3 for BERT model . . . . .	135
101	Confusion Matrix for Class 0 for DistilBERT Model . . . . .	136
102	confusion matrix of class 1 for DistilBERT model . . . . .	136
103	confusion matrix of class 2 for DistilBERT model . . . . .	136

104	confusion matrix of class 3 for DistilBERT model . . . . .	136
105	Confusion Matrix for Class 0 for RoBERTa Model . . . . .	136
106	confusion matrix of class 1 for RoBERTa model . . . . .	136
107	confusion matrix of class 2 for RoBERTa model . . . . .	136
108	confusion matrix of class 3 for RoBERTa model . . . . .	136
109	Confusion Matrix for Class 0 for ResNet Model . . . . .	138
110	confusion matrix of class 1 for ResNet model . . . . .	138
111	confusion matrix of class 2 for ResNet model . . . . .	138
112	confusion matrix of class 3 for ResNet model . . . . .	138
113	Confusion Matrix for Class 0 for CNN Model . . . . .	139
114	confusion matrix of class 1 for CNN model . . . . .	139
115	confusion matrix of class 2 for CNN model . . . . .	139
116	confusion matrix of class 3 for CNN model . . . . .	139
117	Confusion Matrix for Class 0 for ViT Model . . . . .	139
118	confusion matrix of class 1 for ViT model . . . . .	139
119	confusion matrix of class 2 for ViT model . . . . .	139
120	confusion matrix of class 3 for ViT model . . . . .	139
121	Confusion Matrix for Class 0 for RoBERTa Model of Private Data . . . . .	141
122	confusion matrix of class 1 for RoBERTa model of Private Data . . . . .	141
123	confusion matrix of class 2 for RoBERTa model of Private Data . . . . .	141
124	confusion matrix of class 3 for RoBERTa model of Private Data . . . . .	141
125	Confusion Matrix for Class 0 for ViT Model of Private Data . . . . .	141
126	confusion matrix of class 1 for ViT model of Private Data . . . . .	141
127	confusion matrix of class 2 for ViT model of Private Data . . . . .	141
128	confusion matrix of class 3 for ViT model of Private Data . . . . .	141

## List of Tables

1	Confusion matrix for binary-class classification . . . . .	37
2	Confusion matrix for multi-class classification . . . . .	37
3	Summary of existing literature on social media cyberbullying classification . . . . .	47
4	Summary of existing literature on social media cyberbullying classification . . . . .	51
5	Summary of Publicly Collected Dataset . . . . .	55
6	Total data distribution of private dataset . . . . .	56
7	Total data distribution of dataset for each classes . . . . .	64
8	Performance of various Deep-Learning Models for Textual Data of Public dataset . .	87
9	Performance Evaluation of Image Dataset . . . . .	96
10	Performace Evaluation for Multi-modal data's Model for Public Data . . . . .	101
11	Performace Evaluation for Private Data . . . . .	103
12	Comparison with Applied Literature's Dataset for Public Dataset . . . . .	115
13	Comparison with Existing Literature of Multi-class Classification of Cyberbullying . .	117
14	Performance of Each Classes by Hybrid(CNN+LSTM) Model . . . . .	131
15	Performance of Each Classes by LSTM Model . . . . .	132
16	Performance of Each Classes by GRU Model . . . . .	133
17	Performance of Each Classes by BERT Model . . . . .	134
18	Performance of Each Classes by DistilBERT Model . . . . .	135
19	Performance of Each Classes by RoBERTa Model . . . . .	136
20	Performance of Each Classes by ResNet Model . . . . .	137
21	Performance of Each Classes by CNN Model . . . . .	138
22	Performance of Each Classes by ViT Model . . . . .	139
23	Performance of Each Classes by RoBERTa Model . . . . .	140
24	Performance of Each Classes by ViT Model . . . . .	141

# Glossary

- **Multi-modal data:** Multi-modal data combines various types of information, such as text, images, video, and audio.
- **Cyberbullying:** Cyberbullying is an act of harassing, threatening, or embarrassing someone by using digital tools like websites, social media, and messaging services. This type of cyberbullying can have a serious negative impact on victims' mental health and general wellbeing.
- **Hybrid Model:** In deep learning, a hybrid model is a combination of different modeling methods.
  - **Hybrid(CNN+LSTM) model:** The hybrid(CNN+LSTM) model combines two deep learning architectures: CNN and LSTM.
  - **Hybrid(RoBERTa+ViT) model:** The hybrid (RoBERTa+ViT) model combines two deep learning architectures: ViT (Vision Transformer) for handling visual data and RoBERTa (a robustly optimized BERT architecture) for text processing. By combining the best features of both models, this hybrid approach makes it possible to classify text and images at the same time.
- **Research methodology:** Research methodology is a systematic, theoretical examination of the methods used in a field of study. It entails conducting a theoretical analysis of the body of methods and principles associated with a field of study to ensure that the research is sound and the findings are reliable.
- **Deep Learning:** Artificial intelligence that uses layered neural networks to look at different kinds of data is called a deep learning model. These models usually don't have task-specific rules programmed into them; instead, they learn how to do things by looking at examples. This is because deep learning is very good at finding patterns and making predictions.
- **OCR:** Optical Character Recognition, or OCR, works at the forms of letters and numbers in images and turns them into text that can be edited, saved.
- **Graphical User Interface (GUI):** A GUI, short for Graphical User Interface, is a user interface that enables users to interact with electronic devices through graphical icons and visual indicators, rather than relying on text-based interfaces, typed command labels, or text navigation. Graphical User Interfaces (GUIs) enhance the user experience by presenting a visual representation that imitates real-world actions, such as pressing buttons or opening files. This visual



context simplifies the management of software and devices, making them more user-friendly and accessible.

# Acronyms

**AI** Artificial Intelligence

**ML** Machine Learning

**DL** Deep Learning

**CNN** Convolutional Neural Network

**LSTM** Long Short-Time Memory

**GRU** Gated Recurrent Unit

**RoBERTa** Robustly Optimized BERT Pre-training Approach

**ViT** Vision Transformer

**ResNet** Residual Network

**BERT** Bidirectional Encoder Representations from Transformers

**DistilBERT** Distilled BERT

**MCC** Multi-Class Classification

**TP** True Positive

**TN** True Negative

**FP** False Positive

**FN** False Negative

**OCR** Optical Character Recognition

**GUI** Graphical User Interface

# 1 Introduction

Social media platforms, which include a variety of websites and apps such as Facebook <sup>i</sup>, Twitter <sup>ii</sup>, Instagram <sup>iii</sup>, and many more, have completely changed how individuals create, share, and interact with one another in online communities [9]. Short videos have become increasingly popular among the different kinds of content, which are only a few seconds long and show funny and interesting things [10]. All categories of people show their acting, singing, dancing, and other skills in the short video [11]. There are frequently a ton of comments posted in social platform under the comment section [12]. Comments are posted in several modalities including text, images, audio, and video. These different modalities data is known as a *multi-modal* [13] data. The comments posted using multi-modal data on social media include both positive and negative comments. Receiving negative comments all the time has the potential to cause severe psychological consequences, such as depression or suicide, and to have a substantial negative impact on an individual's physical and mental health by eroding their self-confidence [14].

According to the 2014 EU-Kids Online Report [15], 20% of kids between the ages of 11 and 16 have experienced cyberbullying. According to the quantitative research of [16], youths experience cyber-victimization at a rate of 20% to 40%. These all highlight how critical it is to identify a strong and all-encompassing solution to this pervasive issue. The issue needs more progress to find a concrete solution, and it is crucial to keep social media platforms secure and free from negative interactions as short videos continue to draw millions of viewers globally [17]. Automated cyberbullying detection and prevention can effectively address this issue. There are some approaches available to identify bullying incidents [18] and way to support victims [19]. Teenagers often use online platforms with safety centers, such as YouTube's Safety Centre <sup>iv</sup> and Twitter's Safety and Security <sup>v</sup>. In addition, early classification of cyberbullying can greatly reduce the problems of cyberbullying. With the continuous evolution of technology and extensive research conducted within the field of artificial intelligence (AI), there exists the potential to classify cyberbullying.

Deep learning (DL) is a very advanced computational approach at present. This has had a significant impact on various industries by allowing machines to interpret complex data with remarkable efficiency and accuracy. DL has transformed and made significant contribution in various domains such

---

<sup>i</sup><https://www.facebook.com/>

<sup>ii</sup><https://twitter.com/>

<sup>iii</sup><https://www.instagram.com/>

<sup>iv</sup><https://www.youtube.com/howyoutubeworks/policies/community-guidelines/#staying-safe>

<sup>v</sup><https://help.twitter.com/en/safety-and-security>

as healthcare [20], automotive industry [21], in retail [20, 21]. It also widely used for detecting and classifying the cyberbullying from social media. So far, many research has been done of cyberbullying using deep-learning models either using text or images [22–29]. Even with significant advancements, there are still obstacles and challenges exist for classification of cyberbullying using multi-modal data. More advanced techniques such as natural language processing capabilities are clearly needed, as current models frequently fail to capture details of language, such as different forms of bullying [25]. Through the use of a multi-class classification system, social media platforms are able to effectively address particular types of cyberbullying with more subtle and targeted interventions [30]. Although many work has been done on binary classification on multi-modal data, and multi-classification for textual data of cyberbullying, no work has been done to classify the types of cyberbullying into multi-class classification for multimodal data so far.

Therefore, the objective of this thesis is to utilize several deep learning models such as Hybrid (CNN-LSTM), LSTM, GRU, RoBERTa, BERT, DistilBERT, ResNET-50, CNN, ViT, and Hybrid (RoBERTa+ViT) model to multi-classify the cyberbullying using multi-modal data that are posted in several social media platforms.

## 1.1 Motivation and Problem Statement

With the rise of social media, cyberbullying has emerged as a major social issue, with short video reels generating extensive engagement through comments such as text, images, and memes. Negative comments can encourage harmful behaviors that have an impact on people's mental health, potentially leading to serious consequences such as depression and suicide. Deep learning technologies have produced promising results in classifying cyberbullying by processing and learning from large, complex datasets. Current cyberbullying detection and classification methods, which are primarily focused on binary-based multi-modal data [31], [32], [33], [34] or multi-class and multi-label textual data [35], [36], [37]. However, there is no research has thoroughly investigated multi-class classification of cyberbullying for multi-modal data.

This thesis aims to bridge the gap by employing advanced deep-learning models to classify cyberbullying in a multi-modal context. The study will use both public and uniquely collected private datasets from social media platforms, and with a focus on comments associated with short videos. This study aims to improve the accuracy of cyberbullying multi-class classification and contribute to safer online environments by creating and testing multiple deep learning models. Hence, in this thesis, deep learning models especially transformer architectures will be used to classify multi-class classification

of cyberbullying based on multi-modal data on social media.

## 1.2 Thesis Goals

This section outlines the sub-goals of this thesis to achieve the main objective of the thesis.

- Conduct a thorough literature review to review and analyze the current methods and techniques for multi-class classification of cyber-bullying using multi-modal data, especially in the context of social media's post and in the context of social-media short video's comments for public and private dataset respectively. This entails executing a comprehensive literature review in order to comprehend the current approaches, obstacles, and gaps in the field.
- Collect a high-quality dataset of multi-modal data from literature review and short videos comments on social media. This dataset is unique in its composition, consisting of both textual, images and multi-modal comments from short video reel's. The aim is to ensure that the dataset not only represents a wide range of perspectives but is also pertinent to the task of multi-class classification of cyber-bullying incidents.
- Develop a sophisticated multi-modal deep learning pipeline. This pipeline is designed to simultaneously process and analyze the textual, image and multi-modal data, collected from the existing literature and comments section of short video's for the public and private dataset respectively. The ultimate goal here is to accurately identify and classify instances of cyberbullying.
- Evaluate each model's performance using a variety of metrics, including accuracy, F1-score, recall, and precision. Comparing and verifying results of public datasets with the private dataset after getting the result.
- Design and implement a user-friendly graphical user interface (GUI) for the cyberbullying classification system, to represent the classification for both public and private datasets.

## 1.3 Research Questions

To achieve the main objective of this thesis, the following research questions have been outlined.

1. How to collect, label and pre-process a multi-modal cyberbullying dataset from various social media platforms?

To answer this research questions, a public dataset used in existing research works [38], and [37] will be downloaded and utilized. In addition, a private dataset will be collected via APIFY<sup>vi</sup>, and the labeling of the text and image dataset will be done in accordance with the related research [39] and using ChatGPT<sup>vii</sup>. These two datasets will be pre-processed by removing duplicates, filling the null and missing values, resizing images, and balancing the label of each classes.

2. Which deep learning models are best suitable for multi-class classification of cyberbullying using text data?

To answer this research question, six deep-learning models such as a hybrid (CNN+LSTM) model, LSTM model, GRU model, BERT model, DistilBERT model, and RoBERTa model will be developed, and then their performance will be evaluated and compared.

3. Which deep learning models are best suitable for multi-class classification of cyberbullying using image data?

To answer this research question, three deep-learning models such as ResNet-50, CNN, and ViT models will be developed using image data, and then their performance will be evaluated and compared.

4. Which deep learning models are best suitable for multi-class classification of cyberbullying using multi-modal data?

To address this research question, a hybrid fusion model will be developed to perform multi-class classification using multi-modal data. In the fusion module, we will build a deep learning hybrid model i.e., (RoBERTa+ViT) model to classify multi-modal data into multiple classes using late fusion module.

5. How should the results from the built deep learning models be presented on the developed GUI?

To answer this question, a Graphical User Interface (GUI) will be developed and deployed to present the results of the built deep learning models.

6. Does deep-learning perform better than the state-of-the-art algorithms?

To answer this research question, the results of developed deep learning models will be compared with the state-of-the-art work.

---

<sup>vi</sup>[www.apify.com](http://www.apify.com)

<sup>vii</sup><https://chat.openai.com/>

## 1.4 Research Approach

In this thesis, we adopted an “applied research” methodology as shown in Figure:1, to deliver practical solutions addressing cyberbullying. Applied research is a type of research in science that focuses on resolving practical issues and enhancing real-life circumstances [40], [41], and [42]. Various steps involve in the applied research methodology are described as follows.

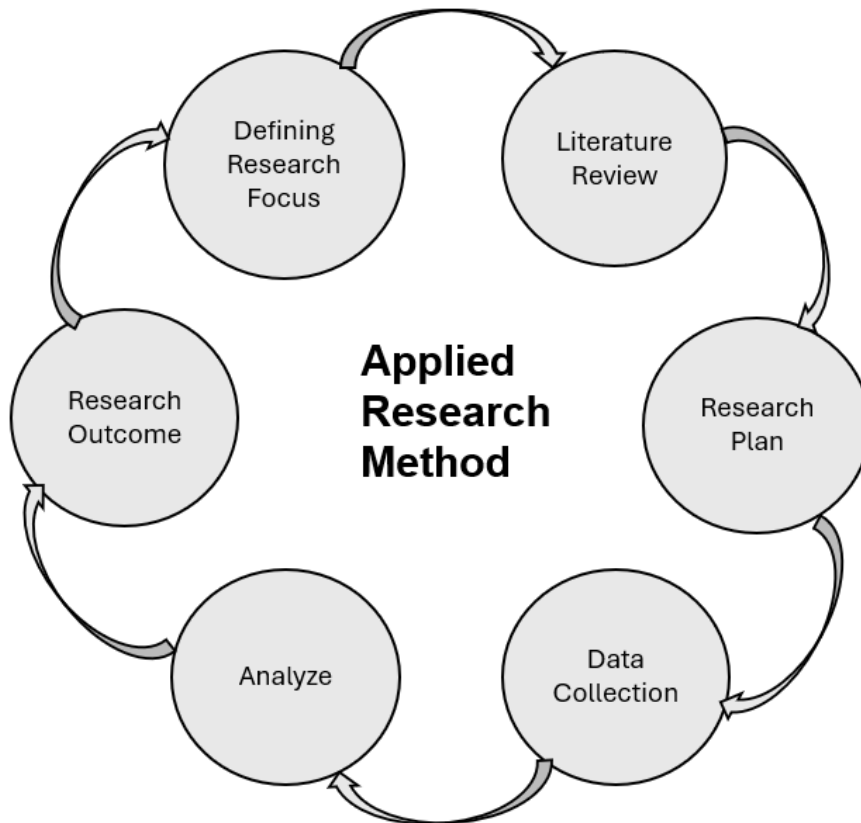


Figure 1: Applied Research Approach

viii

- *Defining Research Focus*: This is the first step in the applied research methodology. This step involves identifying and specifying the primary goal or problem that the study intends to address. It is important because it determines the direction of all future research activities [42]. In this thesis, we begin by defining our research focus, which is the multi-class classification of cyberbullying, identifying by specific area of interest.
- *Literature Review*: A literature review is a systematic collection, analysis, and synthesis of published information on a specific topic. This procedure assists researchers in determining the current state of knowledge, including existing gaps and advancements [42]. So, we conduct a

thorough literature review to determine how current methodologies and deep learning models have been utilized and contributed. It has been a critical step to become acquainted with cutting-edge methodologies and identify gaps in the scholar's previous research.

- *Research Plan:* This is the Third step in the applied research methodology. This includes outlining the approach, resources, timelines, and procedures for conducting the research. A well-structured research plan keeps the project on track and addresses all necessary aspects in a systematic manner [42]. Hence, we create a planned strategy for the research's execution, complete with timetables and procedures.
- *Data Collection:* The fourth step in the applied research methodology is data collection. Data collection is the process of gathering information from multiple sources in order to answer a research question. This may include experimental data, survey results, or data from existing databases [42]. Therefore, we collect the data needed for the study using a variety of techniques, including surveys, experiments, and observations from social media's platform and research review.
- *Analyze:* This is the fifth step in the applied research methodology. Analysis entails processing and examining collected data to reach conclusions or extract insights. This step is critical in converting raw data into useful information [42]. So, we process and analyzed the collected data to extract meaningful insights and patterns. We have applied Hybrid(CNN+LSTM) model, long-short term memory (LSTM) model, GRU model, BERT model, DistilBERT model, and RoBERTa model performs when processing textual data, and it is enhanced when processing visual data with ResNET-50 model, ViT model and convolutional neural networks (CNNs) model. Hybrid model (RoBERTa+ViT) performs when processing the multi-modal data.
- *Research Outcome:* The final step in the applied research methodology is research outcome. The final step is to evaluate and report the research results based on predefined metrics. This step aids in understanding the effectiveness and relevance of the research findings [42]. A range of criteria, including ROC-AUC, F1-score, Recall, Accuracy, and Precision, will be used to examine various deep learning models, guaranteeing a comprehensive evaluation of their efficacy in precisely classification cases of cyberbullying.

## 1.5 Assumptions and Limitations

This section outlines the assumptions we made and the limitations that we encountered in the process of completing this thesis.



### 1.5.1 Assumptions

1. The data collected from various social media platforms, which includes both text, image and memes comments, are typical of online interactions. This includes the assumption that these samples accurately reflect cyberbullying behaviors.
2. It can be assumed that the deep learning models used in the study (Hybrid(CNN+LSTM), LSTM, GRU, BERT, DistilBert, RoBERTa, ResNet-50, CNN, ViT, Hybrid(RoBERTa+ViT)) generalize well to new data beyond the training datasets. These datasets provide reliable performance metrics that accurately reflect the effectiveness of the models.
3. The research assumes that GUI will extract the text from the memes data correctly and will show the classification result for multi-modal data correctly.

### 1.5.2 Limitations

1. The most difficult challenge we faced during the experiments was the limited computational resources. Deep learning models require a significant amount of computational power. As a result, we were unable to perform more complex hyperparameter tuning for textual datasets.
2. Since private dataset has been collected from short-video's comments, there are many noise data. Removing all noises may remove useful information or unusual data points that improve classification performance.
3. Each class may contains several types of data, which can lead to a data being classified into multiple classes. As a result, the data may end up in a different class during each execution phase. Each class contains texts that fall into several different categories, making training and assessing them difficult.

## 1.6 Thesis Contributions

The main contributions of this thesis are:

- This study contributes by collecting multi-modal dataset from existing research known as public dataset and creating a new dataset of text, image and memes comments from social media videos comments known as private dataset. This two datasets are unique in its composition and was created specifically for training and testing multi-modal cyberbullying classification models.

- Developing various deep learning models for multi-class classification of cyberbullying on both public and private datasets for text, image data.
- Development of different deep-learning models for multi-class classification of cyberbullying using multi-modal data.
- Creating a Graphical User Interface (GUI) to present the results of deep learning models for cyberbullying classification.
- Comparing the efficiency of our deep learning models' result with the state-of-the-art result.

## 1.7 Thesis Outline

The rest of the thesis is organized as follows:

**Chapter 2:** This chapter provides background information for understanding theories, technologies, and domains used in the thesis.

**Chapter 3:** This chapter provides a comprehensive review of the current state-of-the-art of previous studies on applying deep-learning for multi-classification of cyberbullying.

**Chapter 4:** This chapter explains data collection and pre-processing process, network architecture, training, and models implementing process.

**Chapter 5:** In this chapter, we provide our obtained experimental results from applying the methods described in Section 4.

**Chapter 6:** In this chapter, we discuss the results obtained in chapter 5, reflect on the research conducted by discussing each research question, and compare the obtained results with the state-of-the-art studies.

**Chapter 7:** This chapter concludes the thesis by summarizing the main achieved results, and outlines the potential future research improvements to reach desired outcomes.

## 2 Background

This section presents the background theory of the problem domain and Deep Learning model's theory and algorithms that have been utilized to answer the research questions of this thesis which are outlined in sub-section: 1.3.

### 2.1 Cyberbullying Identification and Classification

As social media platforms change and grow, the way people talk to each other online on these sites gets more complicated. There are some risks and problems that come with using social media *i.e.* privacy concern, cyberbullying, mental health issue, social media addiction, isolating from family and friends, scams, hacking [43]. One notable concern is cyberbullying from all of them. *Cyberbullying* is defined as the intentional use of internet communication to harass, threaten, bullying, or defame others in order to hurt people. There are some people, who write negative comments on social media's comment section as well as upload aggressive post on social site for defaming another people. As a result, victims of cyberbullying may suffer from anxiety, depression, social isolation, and may even consider or engage in self-harm. So, it has become a major issue at present [44]. It can affect people of all ages and circumstances, but certain demographic groups are more susceptible to this online harassment.

#### 2.1.1 Societal Effects of Cyberbullying

According to APJII (Association of Indonesian Internet Providers) research conducted in 2019, 49% of the 5900 respondents were cyberbullying victims [45]. According to research, the following categories are among the most common targets of cyberbullying:

- *Adolescents and Teens*: Due to extensive use of digital technologies and social media platforms, adolescents and teenagers are especially vulnerable to cyberbullying [46].
- *LGBTQI*: Members of the LGBTQI (*lesbian, gay, bisexual, transgender, queer, intersex*) community are frequently subjected to online harassment based on their sexual orientation or gender identity [47].
- *Minorities and Marginalized Groups*: Cyberbullying can target individuals from minority racial, ethnic, or religious backgrounds based on their identity. Many people are bullied in online because of their skin color [48].
- *Persons with Disabilities*: Individuals with disabilities may be victims of cyberbullying that targets their physical or cognitive conditions [49].

- *Woman*: Cyberbullying based on gender, including harassment and threats, is a concerning issue that affects women and girls [50].

## **2.1.2 Role of Technology in Social Media Monitoring**

At present, AI has been widely applied successfully in various domains, *i.e. education, agriculture, transportation, healthcare, customer service, e-commerce, finance and many more* [51]. Commonly, artificial intelligence refers to systems that execute actions in the physical or digital realm by perceiving their environment, processing, and interpreting vast amounts of information and data. AI systems are capable of adapting their behavior by analyzing how the environment and their conclusions are impacted by their previous actions [52]. Cyberbullying problem can be nicely handled by using machine learning and deep learning method which are both forms of artificial intelligence (AI).

## **2.1.3 Cyberbullying Identification and Classification by Machine Learning**

Humans have used a wide variety of instruments from the beginning of time to complete different activities more quickly and easily. Different machines have been invented as a result of human innovation. These devices made life easier for humans by allowing them to fulfill a variety of demands, such as computing, industry, and travel. And the first one is machine learning. Arthur Samuel defines machine learning as the branch of study that enables computers to learn without the need for explicit programming. A well-known program that played checkers was created by Arthur Samuel. Machine learning, or ML, is the process of teaching machines how to process data more effectively. The need for machine learning is growing due to the number of datasets that are available. Machine learning is used by many sectors to retrieve pertinent data. Learning from the data is the aim of machine learning. [53].

Machine learning-based cyberbullying monitoring and identification leverages sophisticated algorithms to address the increasing incidence of cyberbullying. This method entails compiling a variety of datasets covering different digital communication channels, such as text messages, comments, emails, and social media exchanges. The meticulous selection of characteristics from this data, including sentiment analysis, linguistic patterns, and contextual data, is essential to its success. These characteristics are fed into machine learning algorithms, which are trained to distinguish between instances of cyberbullying and legitimate communication. In this process, methods like logistic regression, support vector machines, and deep learning architectures like convolutional and recurrent neural networks are frequently used.

After being trained, these models are put through a thorough review process to make sure they can correctly identify instances of cyberbullying. Performance is measured using metrics like recall, precision, and F1-score; cross-validation methods provide validation on several datasets. The models are deployed and integrated into numerous online platforms and communication channels after they have been validated in order to track user interactions in real-time. Cyberbullying incidences can be quickly reduced by triggering automatic alerts or interventions when the model identifies potentially harmful activity.

A vast amount of research has been done on cyberbullying using machine learning model [54]. Arif and Mohammad [55], presented systematic review for cyberbullying identification and classification using machine learning model.

#### **2.1.4 Cyberbullying Identification and Classification by Deep Learning**

Deep learning is a relatively recent discipline within the machine learning field. Artificial neural networks refer to deep learning algorithms inspired by brain structure and function. Furthermore, deep learning algorithms are trained to extract and comprehend meaningful representations from the data itself rather than simply following traditional programmed instructions. Meaningful representation is obtained by combining simple yet non-linear modules, each of which transforms a representation at one level (beginning with raw input data) into a representation at a higher level. Thus, deep learning algorithms have been demonstrated to be effective in classifying all types of data. These algorithms are categorized into three types: learning that is supervised, semi-supervised, or unsupervised. Furthermore, Deep Learning necessitates vast amounts of data and expensive computing hardware, such as a powerful graphics processing unit (GPU) [34].

Deep learning architectures, which are well-suited for cyberbullying monitoring, identification, and classification tasks. The first step in the procedure is gathering a variety of datasets with instances of friendly contacts and cyberbullying. Following preprocessing, these datasets are put into deep learning models, which automatically extract pertinent textual properties. These models learn hierarchical representations of the data through successive layers of neurons, which allows them to identify contextual cues and subtle nuances that are indicative of cyberbullying behavior. Iterative optimization procedures are used during the training of deep learning models, with the goal of maximizing classification accuracy and minimizing prediction errors. Although this training stage frequently calls for substantial computational resources and a vast quantity of annotated data, the results can be quite accurate cyberbullying identification systems. Standard measures like accuracy, recall, and F1-score are used to assess the performance of deep learning models once they have been trained. Cross-

validation is one of the validation strategies that guarantees the models' capacity to generalize across different datasets. Deep learning models are deployed and integrated into communication channels and internet platforms for real-time monitoring after they have been validated. When the models identify potentially harmful activity, automated notifications or actions can be set off, allowing for prompt intervention and event reduction pertaining to cyberbullying.

Deep learning's ability to analyze large amounts of data and identify patterns in it makes it perfect for creating complex algorithms that can accurately identify and classify cyberbullying. So, as a result, a huge number of research has been done for cyberbullying using deep learning model. Chapter:3, has been discussed into more details and clearly about the existing work for cyberbullying using deep learning model.

## 2.2 Data Classification

Classification is the process of estimating the mapping function that connects an input sample to a target class or label [56]. Single-label and multi-label classifications are two categories into which the classification techniques can be divided based on the label association to the input samples [57].

- **Single-label Classification:** The single-label classification problem is divided into two categories: binary and multi-class classification [58].
  - *Binary classification:* This classification involves categorizing input data samples into two categories. Binary classification is the fundamental requirement for any classification technique [57], [58].
  - *Multi-class classification:* This classification occurs when input samples match one or more target labels [57], [58], [27], [59].
- **Multi-label Classification:** Multi-label classification assigns a set of target labels to each input sample, unlike single-label classification. The number of target labels for each input varies dynamically. This complicates the implementation of multi-label classifiers [60].

## 2.3 Deep-Learning Models for Multi-Modal Data

In this section, we have described all the deep-learning models basic overview that we used in our experiment.

Deep-Learning model focuses on making and using artificial neural networks to solve the complex problem. It is based on how the human brain is built and how it works. Computational models consisting of several processing layers can acquire representations of data with various levels of abstraction through deep learning. The state-of-the-art has been significantly enhanced by these techniques in numerous fields, including drug discovery and genomics, speech recognition, visual object recognition, and object identification in each layer based on the representation in the preceding layer, deep learning uncovers complex structure inside massive data sets. Advances in the processing of pictures, video, speech, and audio have been made possible by deep convolutional nets, while recurrent nets have shed light on sequential data, including text and speech [21]. Some deep-learning models has been discussed in the following subsections:

### 2.3.1 The LSTM Model

LSTM stands for Long Short-Term Memory. is a type of Recurrent Neural Network (RNN) architecture designed to overcome the vanishing gradient problem and better capture long-term dependencies in sequential data. Neurons in an RNN are connected to one another by directed cycles. Because the RNN model processes a sequence of words or inputs using internal memory, it processes the data in a sequential fashion. Each element's output depends on all of the inputs from earlier nodes and remembers information, RNNs execute the same task for every element. Because of the special construction of an LSTM, which consists of input, output, and forget gates among other components, the network may retain information for a long time. By regulating the information flow, these gates enable the network to keep or delete data according to its applicability. Time series prediction, speech recognition, language modeling, and other sequential data challenges are common applications for LSTMs. Relevant articles with names that emphasize sequence modeling or long-term dependency management frequently examine LSTM applications or enhancements [6], [1]. Figure:2 shows the architecture of LSTM model from the study of Van *et al.* [1].

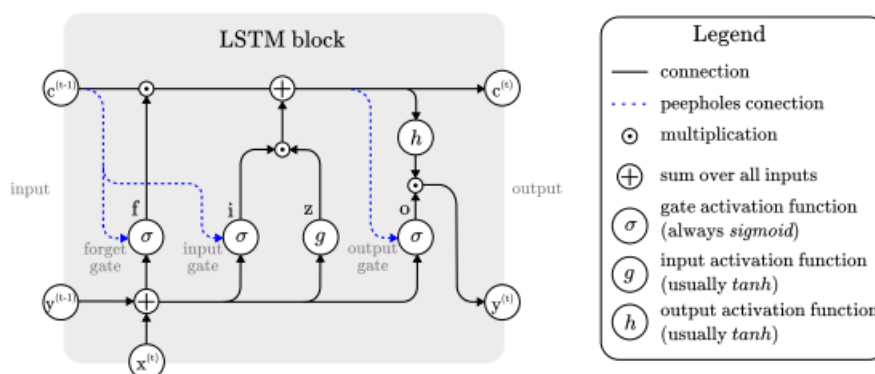


Figure 2: The Architecture of LSTM model [1].

### 2.3.2 The GRU Model

An LSTM variant that is a slightly simpler is the Gated Recurrent Unit. It has an extra "reset gate" and merges the input and forget gates into a single "update gate." The final model is getting more and more traction and is less complicated than typical LSTM models [61]. To be more specific, the update gate is created when GRUs join the input and forget gates of the LSTM. In addition, GRUs have an additional gate known as the reset gate. These gates control and safeguard GRUs. The reset gate functions in the same way as the update gate, controlling the amount of new data that is added to the current unit. It assists the network in determining which state variables need to be stored in memory or ignored. In addition, the update gate and reset gate seek to identify both short- and long-term dependencies. When backpropagation occurs during training, the weights of the pertinent gates are likewise adjusted [62]. Figure:3 shows the architecture of GRU model from the study of Fang *et al.* [2].

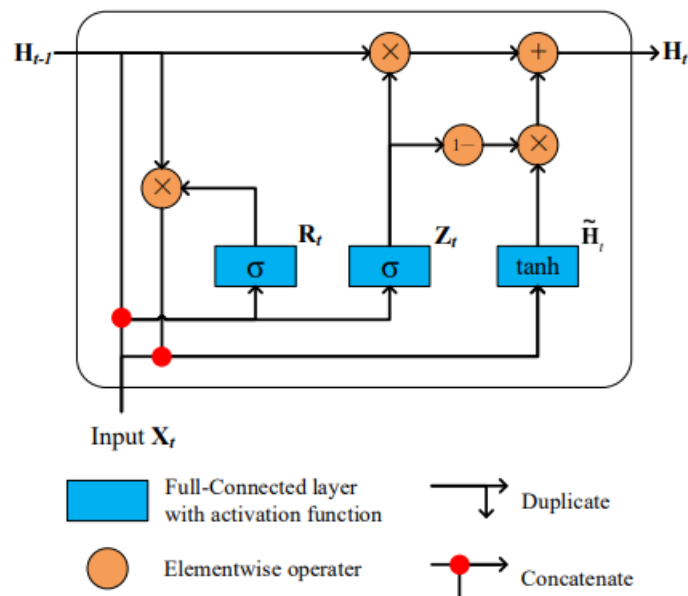


Figure 3: The Architecture of GRU model [2].

### 2.3.3 The RoBERTa Model

Bidirectional Encoder Representation from Transformers (BERT) is extended by the RoBERTa model. The Transformers [63] family of models, which was created for sequence-to-sequence modeling to solve the long-range dependencies issue, includes the BERT and RoBERTa. Transformer models consist of three parts: heads, transformers, and tokenizers. The sparse index encodings are created from the raw text by the tokenizer. The sparse content is then transformed by the transformers into contextual embedding for more in-depth training. In order to exploit the contextual embedding for the downstream activities, the heads are implemented to cover the transformers model. Compared to



other language models, BERT has the ability to acquire contextual representation from both ends of sentences, which sets it apart from the others. BERT employed 30K vocabulary at the character level using byte-pair encoding for the tokenization process. As opposed to this, RoBERTa employed byte-level Byte-Pair Encoding and a bigger vocabulary collection with 50K subword units. Aside from that, by training on more data, longer sequences, and longer times, the RoBERTa model improves the BERT model.

The text in the RoBERTa model is divided into subwords using the byte-level Byte-Pair Encoding tokenizer. The frequently used terms won't be divided by this tokenizer. Nevertheless, uncommon words will be divided into subwords. The word "Transformers," for example, will be divided into the words "Transform" and "ers." The text must be converted into a meaningful numerical representation for the model to comprehend it. The raw text is encoded with input ids and an attention mask by the RoBERTa tokenizer. The input ids stand for the token's numerical representation and indexes. However, to group the sequence together, the attention mask is provided as an optional input. The attention mask shows which tokens need to be paid attention to and which should not.

The RoBERTa base model receives the input ids and attention mask. The RoBERTa base model consists of 12 RoBERTa foundation layers, 768 hidden state vectors, and 125 million parameters. In order to make it easier for the subsequent layers to extract the relevant information from the word embedding, the RoBERTa base layers seek to produce a meaningful word embedding as the feature representation [64]. Figure:4 shows the architecture of RoBERTa model from the study of Huang et al. [3].

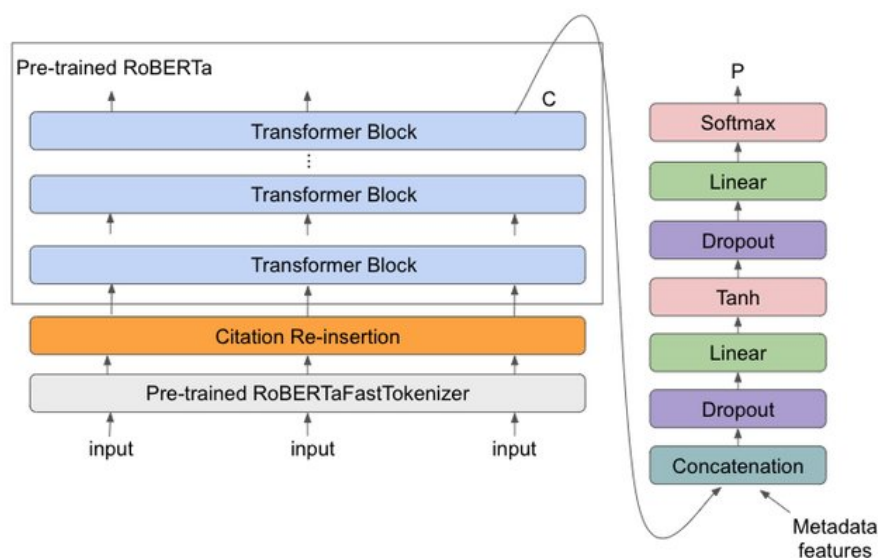


Figure 4: The RoBERTa Model Architecture [3].

### 2.3.4 The CNN Model

A unique kind of neural network used in image processing is the convolutional neural network, or CNN. Nonetheless, the CNN approach has proven useful for classifying texts. CNN layers are referred to as feature maps since a convolutional layer in the CNN model connects a subset of the input to its earlier layers. Pooling layers are used by the CNN model to lower computing complexity. CNN's pooling procedures conserve crucial information by reducing the output size of one stack layer to the next. While there are other pooling methods available, max-pooling—in which the pooling window has a max value element—is the most frequently employed. The output of the pooling layer is fed into and mapped to the subsequent layers by the flattening layer. In CNN, the last layer is usually fully connected [6]. Figure:5 shows the architecture of CNN model from the study of Phung *et al.* [4].

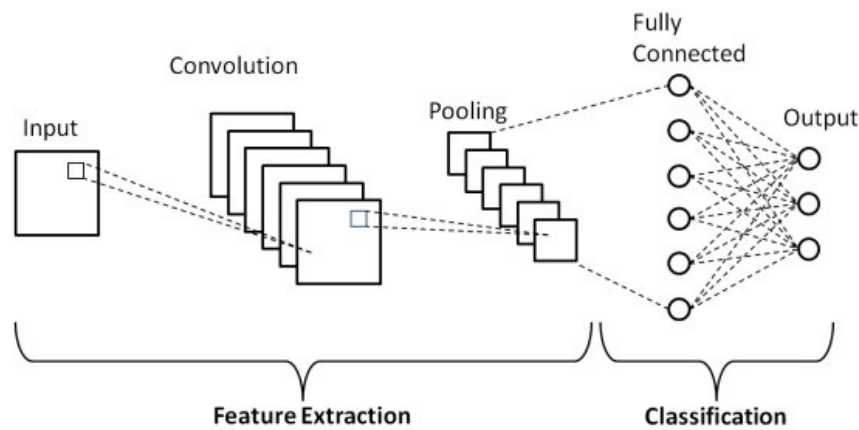


Figure 5: CNN Architecture [4].

### 2.3.5 The ViT Model

The Vision Transformer (ViT) model represents an innovative strategy for image classification that capitalizes on the transformer architecture, which was initially developed for applications in natural language processing. The ViT algorithm commences by partitioning the input image into segments of consistent size that do not overlap. By linearly embedding each patch into a planar vector, an order of image tokens is produced. In conjunction with a positional embedding that can be learned, these characters function as the input for the transformer model. The transformer processes these tokens using feedforward networks and multiple layers of self-attention mechanisms. In contrast to CNN's pixel-array architecture, ViT employs a sequence of visual identifiers. Figure:6 shows the architecture of ViT model that has taken from the study of Dosovitskiy *et al.* [5].

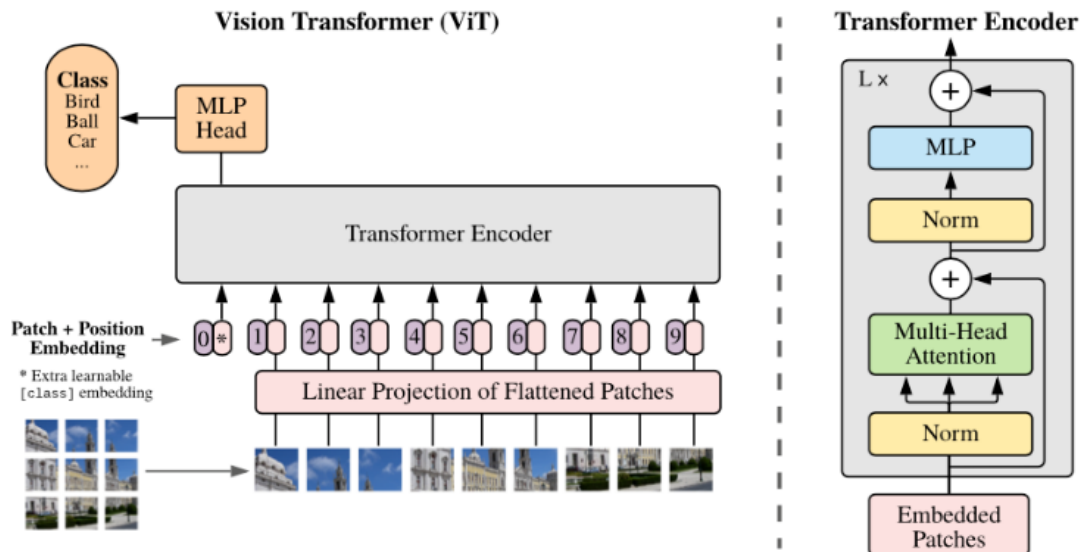


Figure 6: ViT Architecture [5].

### 2.3.6 The Hybrid(CNN-LSTM) Model

The convolutional neural network (CNN), which is constrained by the size of the local window and local textual features can be extracted. CNN is unable to determine the long-term dependency of lengthy texts, such as news articles. Text's long-term reliance can be learned using another deep learning recurrent neural network model that is based on long short-term memory (LSTM). Thus, an CNN-LSTM Hybrid model is can be build for text classification tasks, *such as* [65], [66], and [67]. Figure:7 shows the process of working CNN-LSTM model together as hybrid model has taken from the study of Tasdelen *et al.* [6] .

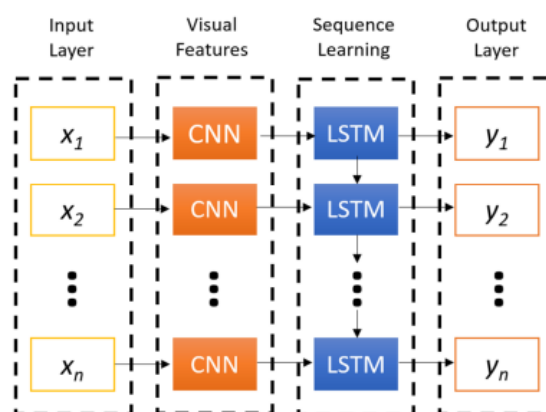


Figure 2. The basic architecture of the CNN-LSTM network.

Figure 7: Hybrid(CNN-LSTM) Model Working Process [6].

### 2.3.7 The BERT Model

BERT (Bidirectional Encoder Representations from Transformers) is a groundbreaking model in natural language processing (NLP). This model, developed by Google researchers and described in [68]. BERT's core technique is to train a language model bidirectionally, which is a significant departure from previous models that typically processed text in a single direction (left-to-right or right-to-left). This bidirectionality enables the model to understand a word's context based on all of its surroundings (both from the left and right), rather than just one side (see figure:8) [7].

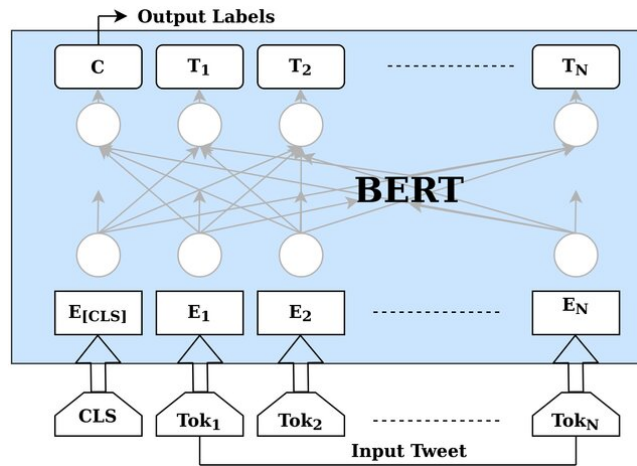


Figure 8: Bert Model Architecture ( collected from [7])

### 2.3.8 The DistilBERT Model

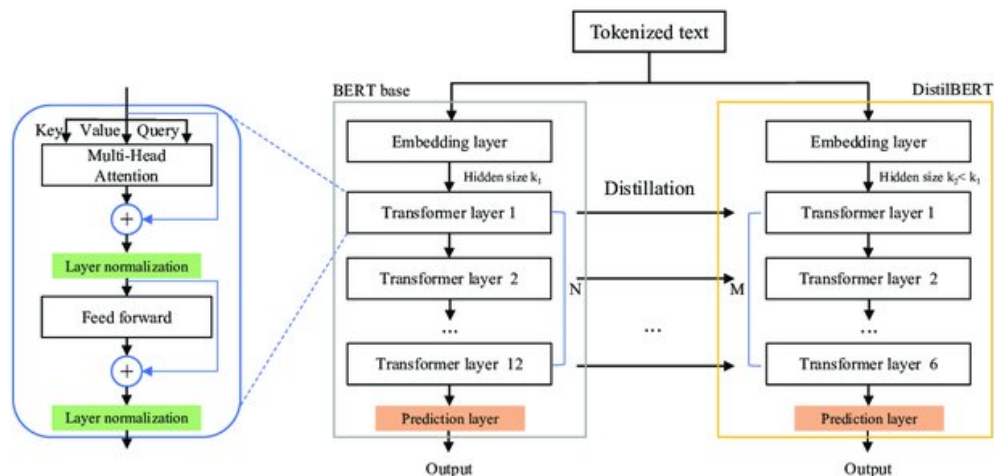


Figure 9: DistilBert Model Architecture ( collected from [8])

DistilBERT is a streamlined version of the well-known BERT model (Bidirectional Encoder Representations from Transformers), designed for increased efficiency and speed. Hugging Face created this

model to address the resource-intensive nature of BERT, reducing its size by roughly 40% while retaining approximately 97% of its performance capabilities. The smaller size not only improves computational efficiency, but it also makes it suitable for environments with limited resources, such as mobile devices or applications that require quick response times. This model is particularly adaptable, finding applications in a variety of NLP tasks such as text classification, question answering, and language inference [69]. Figure:9 shows the architecture of DistilBERT model, which is collected from the study of Adel *et al.* [8]

### 2.3.9 Hyperparameters and Hyperparameter Tuning

A key idea in machine learning and deep learning, hyperparameters and hyperparameter tuning are important in deciding how well a model performs.

#### Hyperparameters

The parameters that specify a model's structure or configuration are known as hyperparameters, and they are not discovered by training from the data. They are predetermined and don't change during the training session. The number of hidden layers in a neural network, the number of trees in a random forest, the regularization parameter in regression models, and the learning rate in gradient descent are a few examples of hyperparameters. To achieve the best possible model performance, it is imperative to select the right hyperparameters [70].

#### Hyperparameter Tuning

The process of determining the ideal set of hyperparameters for a particular model and dataset is referred to as hyperparameter tuning, hyperparameter optimization, or model selection. Given that hyperparameter tuning has a substantial effect on a model's performance, it is an essential phase in the machine learning and deep-learning process. Through a methodical approach to finding the ideal set of hyperparameters, we can enhance the model's capacity for generalization and attain superior outcomes with previously unexplored data. Usually, this procedure entails utilizing different search methods, including grid search, random search, Bayesian optimization, or evolutionary algorithms, to comb through a predetermined range of hyperparameters [70].

- Grid Search: Grid search involves specifying a preset set of hyperparameters and training and evaluating the model for every possible combination of hyperparameters in the grid. The ideal collection of hyperparameters is determined by combining the values that produce the greatest results on a validation set [70].

- **Random Search:** A preset distribution of hyperparameters is randomly sampled, and the model's performance is assessed for each sampled set of hyperparameters using random search. When the search space is big, this method performs better than grid search [71].
- **Bayesian Optimization:** Using probabilistic models, Bayesian optimization is an iterative model-based optimization method that approximates the objective function. In order to quickly explore the hyperparameter space and identify the ideal set of hyperparameters, it modifies the search based on prior evaluations [70].
- **Evolutionary Algorithms:** Genetic algorithms and other evolutionary algorithms mimic the process of natural selection to gradually evolve a population of potential solutions over several generations. A collection of hyperparameters is represented by each candidate solution, and the most suitable individuals are chosen to procreate and create the next generation. Until a halting requirement is satisfied, this iterative procedure keeps going [72], [70].

### 2.3.10 Performance Metrics for Classification

We have to assess the model's performance after it has been built and the data has been trained. To evaluate the model for this, we must employ performance measures. To do this, abide by these guidelines:

- **Confusion Matrix:** Confusion matrix provides an insightful and thorough display of classifier performance. It is a magnifier that gives us a better understanding of the classifier's internal workings rather than merely another method of calculating Precision, Recall, or any other assessment metric. Confusion matrix analysis may also shed light on the relationships between various data objects and features as well as the underlying structure of the data. Confusion matrices are widely utilized in many different fields, including computer vision [73], natural language processing [74], acoustics [75], and many more. They have long been a part of the evaluation of scientific theories and engineering applications. In its most basic form, a confusion matrix represents the percentages of four possible classification outcomes: True Positive (TP), False Positive (FP), True Negative (TN), and False negative (FN). It displays the performance of a binary classifier in a table with two rows and two columns [76], [77], and [78]. This idea can be easily applied to the presentation of findings from the Multi-class classification model [79], in which each object in the data set can only ever belong to one of several unique classes at any one moment. [80]. Table:1 shows the confusion matrix for binary classification, and table:2 shows the confusion matrix for multi-class classification.

		Predicted Class	
		Class 1	Class 2
True Class	Class 1	TP(1)	FN(2)
	Class 2	FP(1)	TP(2)

Table 1: Confusion matrix for binary-class classification

		Predicted Class		
		Class 1	Class 2	Class 3
True Class	Class 1	TP(1)	FN(2)	FN(3)
	Class 2	FP(1)	TP(2)	FN(3)
	Class 3	FP(1)	FP(2)	TP(3)

Table 2: Confusion matrix for multi-class classification

However, Confusion matrix also can be represented through heat map. *Heat maps* use color shades to depict two-dimensional numerical tables [81]. The most popular method for representing a confusion matrix is to preserve its form while creating a heat map out of the values of each field [82]. Figure:10 shows a heap map for confusion matrix.

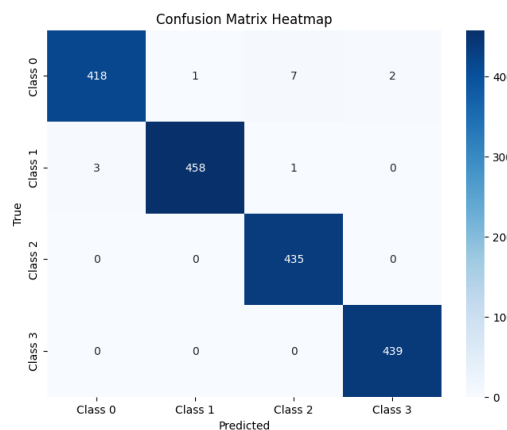


Figure 10: Confusion Matrix in Heat Map

- **F1-score:** The F1-Score combines precision and recall into a singular value. It is particularly useful when working with unbalanced datasets in which one class predominates over the other [80]. Equation1 shows the formula to calculate the f1-score.

$$F1\text{-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (1)$$

- **Recall:** Recall is a way to measure how well a model can find all important instances in a dataset. It is the number of cases of real bullying that were correctly found out of the total number of real bullying cases. The metric of recall assesses the model's capacity to accurately detect and classify all instances of cyberbullying that truly exist. [83], [80]. Equation2 shows the formula

to calculate the recall.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (2)$$

- *Accuracy*: Accuracy measures the overall correctness of prediction made by a model [80], [83]. It is the proportion of cyberbullying and non-cyberbullying instances correctly classified out of the total number of instances. Equation3 shows the formula to calculate the accuracy.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Number of Instances}} \quad (3)$$

- *Precision*: Precision measures how well cyberbullying cases can be picked out of the expected positive cases. It is the amount of correct predictions compared to the total number of correct predictions. It assesses the accuracy of a model's positive predictions [80]. Equation4 shows the formula to calculate the precision.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (4)$$

- *ROC-AUC*: A popular statistic for assessing the effectiveness of binary classification models is the Receiver Operating Characteristic - Area Under the Curve (ROC-AUC). It assesses a model's capacity to discriminate between positive and negative classes throughout the whole range of thresholds. The genuine positive rate (sensitivity) vs the false positive rate (1 - specificity) for various threshold values is shown graphically by the ROC curve. Higher numbers denote greater performance. The AUC condenses the ROC curve into a single scalar value, ranging from 0 to 1 [84] [70].
- *early stopping*: During training, validation can be used to identify the onset of overfitting. Early stopping, or stopping training before convergence, can then be implemented to prevent overfitting. However, the precise criterion for validation-based early stopping is typically determined on the fly or training is halted interactively. This trick explains how to choose a stopping criterion in a methodical way. Depending on the situation, this trick can be used to improve generalization or expedite learning processes [85].

## 2.4 Software Languages and Tools for Model Deployment

In this section, we will discuss about the tools we utilized in this part for model deployment.



### **2.4.1 Hypertext Markup Language**

The common markup language used to create online pages and web apps is called HTML (Hypertext Markup Language). It gives web content structure by defining elements like headings, paragraphs, images, links, and forms with different tags and attributes. Tim Berners-Lee, the man behind the World Wide online, first presented HTML, and it has grown to be an essential tool for creating online pages and apps from the beginning. Web pages are structured using HTML, which enables content to be shown and organized consistently across various browsers and devices. When it comes to model deployment, web apps that communicate with machine learning models are usually created using HTML to establish their user interface (UI). This entails creating input forms, showing model predictions, and giving people access to the outcomes [86], [87].

### **2.4.2 Cascading Style Sheets**

A style sheet language called CSS (Cascading Style Sheets) is used to specify how an HTML document is presented. By defining how HTML elements should be shown on the computer, in print, or in other media types, it makes it possible to separate content from presentation. In order to improve the visual presentation of web pages. It gives users more control over stylistic elements such as layout, color scheme, and font choice. By styling HTML elements, CSS improves the visual appeal and usability of web pages. It enables designers to specify a website's style and feel, guaranteeing uniformity and adaptability to various screen sizes. CSS is frequently used in model deployment to alter the look of online applications. This includes adjusting fonts, colors, margins, padding, and responsive design for varying screen sizes [87].

### **2.4.3 JavaScript**

Programming languages like JavaScript are frequently used to produce interactive web effects for browsers. In addition to providing for dynamic content changes, event management, form validation, and much more, it allows the manipulation of HTML and CSS. Brendan Eich developed JavaScript and it was first created as a web browser client-side scripting language. Web sites can become more dynamic and engaging by adding behavior and interactivity through the use of JavaScript. It enables programmers to manage user input, design adaptable user interfaces, and carry out event-driven operations. JavaScript is frequently used in model deployment to improve the user experience of web apps that communicate with machine learning models. Developing functionalities like client-side validation, interactive visualizations, and real-time updates may fall under this category [88], [87].

#### **2.4.4 Python Programming Language**

Python is a popular programming language with many uses, including web development. It is quite adaptable. Python is frequently used for backend development, data processing, and integration with machine learning models in the context of model deployment. Flask and Django are popular frameworks for Python web application development. Guido van Rossum designed Python, which was originally made available in 1991. Since then, Python has grown to be one of the most widely used programming languages globally, praised for its ease of use, readability, and large standard library. Python's versatility, user-friendliness, and extensive library and framework ecosystem make it a popular choice for model deployment. It enables developers to manage data processing chores, quickly construct and launch web apps, and seamlessly incorporate machine learning models into live systems. Python is frequently used in model deployment for backend development, where it manages operations including serving model predictions, connecting with databases, and processing HTTP requests. Building web applications is made easier by frameworks such as Flask and Django, which offer tools and protocols for handling routing, request management, and response creation [89].

#### **2.4.5 Google Colaboratory**

Google Colaboratory is a free cloud-based tool that lets users build and run Python code in an online environment. It offers a Jupyter notebook interface via which users may see outputs, write and execute code cells, and see data visualization. Google introduced Google Colab as a component of its GCP (Google Cloud Platform) offerings. By offering a free platform with access to GPU and TPU resources for code execution, it seeks to increase the accessibility of machine learning research and education. Google Colab is frequently used for many different things, such as machine learning, data analysis, and teaching. It provides free access to computer resources like GPUs and TPUs, which are necessary for effectively training machine learning models. It also offers smooth notebook sharing and storing integration with Google Drive. There is no setup necessary for users to access Google Colab through a web browser. They can install and use third-party libraries, write and run Python code, create new notebooks or upload ones that already exist, and work in real-time collaboration with others [90], [91].

#### **2.4.6 Visual Studio Code**

Microsoft created Visual Studio Code, sometimes shortened to VS Code, which is a free and open-source code editor. For authoring, debugging, and deploying code on a variety of platforms and programming languages, it offers a lightweight yet robust environment. Because of its cross-platform

compatibility, performance, and flexibility, Visual Studio Code has grown in favor among developers since its initial release by Microsoft. It is compatible with Windows, macOS, and Linux and was developed with web technologies like Electron. Developers utilize VS Code for a variety of tasks, such as data science, web development, and software development. In order to adapt the editor to various workflows, it provides capabilities like syntax highlighting, code completion, debugging, version control integration, and a vast marketplace of extensions. To write and manage code projects, developers can utilize Visual Studio Code, which they can download and install on their local computers. It offers built-in terminal access, Git integration, support for multiple development tools and frameworks, and compatibility with a wide range of programming languages [92].

## 3 Literature Review

There is significant research going on in the areas of developing deep learning (DL) algorithms for cyberbullying classification. In this chapter, we present the existing literature on applying deep learning for the binary class and multi-classification of cyberbullying using multi-modal data, and then outline the limitation of existing works.

### 3.1 Existing Research on Applying DL for Cyberbullying Binary-class Classification

Many research studies have been done using deep-learning models to perform binary class classification. In this section, we present some of the research on binary-class classification to understand used deep-learning models on cyberbullying, collection of data, information, and ideas about models and fusion module for this thesis.

Chandrasekaran *et al* [31] introduced a novel model called FSSDL-CBDC (*Feature Subset Selection with Deep Learning - Cyber Bullying Detection and Categorization*) for cyberbullying identification and classification. The authors utilized deep learning models for the feature subset selection in the context of detecting and classifying cyberbullying on social networks using benchmark dataset. In order to identify and classify cyberbullying (CB) occurrences within social networks, researchers have integrated the salp swarm algorithm (SSA) with a deep belief network (DBN), resulting in the development of the SSA-DBN model. The utilization of the salp swarm algorithm (SSA) in conjunction with the deep belief network (DBN), referred to as the SSA-DBN model, has been employed for the purpose of detecting and categorizing cyberbullying (CB) instances within social networks. In order to enhance the identification capabilities of their proposed FSSDL-CBDC technique, the researchers conducted a series of simulations on a benchmark dataset to provide a thorough evaluation, and ended up with 99.983% accuracy. However, unsupervised feature selection (FS) for outlier detection (OD) in streaming data (SD) for fields like intrusion detection and network security, where large amounts of high-dimensional data that need to be analyzed in near real time are becoming more of a problem.

N. K. Singh *et al* [33] worked on cyberbullying identification on social media using deep learning techniques. The authors used a dataset contains 48,000 tweets from Twitter that included messages associated with demographic characteristics such as age, religion, gender, and ethnicity. Traditional machine learning algorithms such as Naive Bayes, Logical Regression, and Support Vector Machine (SVM) were utilized alongside ensemble machine learning models such as Random Forest and XG-

Boost. In addition, the authors incorporated models such as Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) into their methodology. Among all the models, GRU outperformed with a F-1 score of 0.92, which is notable.

Alotaibi *et al.* [34] developed a multi-channel deep learning framework for cyberbullying identification and classification on social media. The authors used Twitter data and applied a combination of bi-directional gated recurrent unit (BiGRU), the transformer block, and the convolutional neural network (CNN) deep learning models. The framework demonstrated a commendable 88% accuracy rate in identifying cyberbullying content within social media platforms. However, the work used only textual data for identifying and for binary classification (aggressive and non-aggressive) of cyberbullying.

Kumar *et al* [28] worked on multi-modal cyberbullying identification and classification by utilizing capsule network with dynamic routing and deep convolutional neural networks. The authors used a dataset contained 10,000 comments which were collected from YouTube, Instagram, and Twitter. A combination of Convolutional Neural Network (CNN) and Capsule Network (CapsNet) had been used allowing them to create a hybrid deep learning model called "CapsNet-ConvNet". This model was created to identify cyberbullying from multi-modal data i.e, text, images, and info-graphic data. The findings of their study shows that, the proposed hybrid model, CapsNet-ConvNet achieve accurate and thorough results with 98% AUC-ROC.

Kumari *et al* [93] identified and classified cyberbullying from social media posts. The goal of this study was to collect data from social media posts contained both textual and image data, and to identify cyberbullying. In their research, they attempted to extract combined text and image features to identify various cases of cyberbullying. To extract features from images and text, they used a pre-trained VGG-16 network and a convolutional neural network. These features are further optimized using genetic algorithms to improve the overall system's efficiency to identify cyberbullying, and the study results showed that the model achieved a F1-score of 0.78.

Kumari *et al* [94] worked on cyberbullying free social media in smart cities. The study used a similar dataset to [93] for their work. They proposed a unified representation of text and images to eliminate the need for separate learning modules for graphics and text. The unified representation was implemented using a single - layered Convolutional Neural Network (CNN) model. The main findings of this study were that text presented as visuals proved to be a better model for encoding information. They also discovered that a single - layered CNN model produced superior results using a two-dimensional representation with 2048 filters of one-gram Term Frequency-Inverse Document Frequency alone.

In the given context, they used three layers of text and three layers of a color image to represent the input, resulting in a 74% recall of the bullying class with just one layer of CNN model. However, the weakness of this study is that the study aimed to identify whether the posts' comment is bullying or non-bullying, without categorizing the bullying into specific classes.

Singh and Sharma [95] worked on multimodal cyberbullying identification. The study employed audio, visual, and textual data from Twitter, ADIMA. A hybrid Bi-directional Long Short-Term Memory assisted Attention Hierarchical Capsule Network (BiLSTM-AHCNet) model was utilized for textual analysis. This model combined the dynamic routing of a capsule network, the nuanced detection capabilities of an attention mechanism, and the advantages of BiLSTM for comprehending the context in textual material. The Tuned Aquila EfficientNetBo (Tuned AEBo) model was used for image data analysis. They used the Librosa library, a well-liked Python tool for audio and music analysis, and also for audio feature extraction. Following their extraction, an Attention Convolutional Neural Network (ACNN) model was applied to the audio features. ACNNs were designed to handle the spatial hierarchy of sounds while concentrating on significant aspects of the audio data. The accuracy, F1-measure, specificity, and AUC of their outcome was 98.23%, 98.22%, 98.47%, and 0.982 respectively.

Ilavarasan *et al* [96] conducted the research on identifying the cyberbullying from multi-modal data by applying the pre-trained deep learning models. This study utilized the same dataset that was used in [93]. This dataset contained text and image data which was collected from Facebook, Instagram, and Twitter posts. For classifying cyberbullying, this study employed RoBERTa model for text data and Xception model for image data. The proposed approach uses a Light Gradient Boosting Machine (LightGBM) classifier to determine whether tweets are bullying or not. The suggested method successfully identified cyberbullying, achieving a 92% recall and an 82% F1-score for bullying class. The suggested model's weighted F1-score is 80%.

Sing *et al* [97] suggested a novel hybrid methodology by applying both machine learning and deep learning methods to classify cyberbullying. The used data was collected from YouTube i.e., both textual and image data. In this regard, the study uses Natural Language Processing (NLP) to identify and address the concerning phenomenon of electronic bullying. When detecting cyberbullying on social media, the machine learning (ML) approach is moderated according to predetermined features or criteria. The K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Naive Bayes (NB), Decision Trees (DT), and Random Forest (RF) techniques were used to analyze the gathered characteristics. They assessed the models and obtained the KNN (0,90), SVM (0,92), and Deep Learning: CNN-LSTM (0,96) accuracy values.

Paul *et al* [98] developed deep learning model to identify the cyberbullying from social media posts. In this study, multi-modal data was collected from Vine social media site and named it as Vine dataset. The ResidualBiLSTM-RCNN model was employed for both the textual and visual data. This study's model achieved an F1-measure of 0.75. The limitation of the study is, the researchers only focused on binary classification i.e., bullying or non-bullying, but not on multi-class classification of cyberbullying.

Romim *et. al* [99] employed machine learning and deep learning models to identify the hate speech from the Bengali language. This study introduced a new dataset of 30,000 users' comments which was gathered from Facebook and YouTube comments' sections. This collected data was further validated by experts and tagged them using crowd-sourcing. The collected data contained seven categories i.e., *sports, entertainment, politics, religion, crime, celebrities, and TikTok and meme*. The experiments results showed that SVM produced the best results with an accuracy of 87.5%, although all the deep learning models performed well.

Idrizi and Hamiti [100] conducted a study on multi-modal data extracted from social media platforms. This study conducted an analysis of various media formats (text, photos, and videos) disseminated on social media platforms with the aim of identifying instances of cyberbullying. This study introduced a graph convolutional neural network, a pre-trained Googlenet, a Mel-scale filter bank speech spectrogram, and a CNN network model for audio post-classification. The study's primary findings indicate that employing graph convolutional neural networks and MFCCs for audio post-processing yielded superior outcomes, including one-dimensional representation, for both text and image characteristics. To accomplish this, the researchers utilized a combination of Graph Convolutional Networks (GCN) and Melfrequency cepstrum to represent text, image, and video data in this specific context. This approach resulted in an accuracy rate of 85% for identifying instances of bullying. However, the study's limitation is its dependence on a binary classification system.

Hamza *et. al* [38] classified multi-modal data consisted of religiously abusive memes. The used dataset contained textual and image data of approximately 2000 meme images from social media platforms including several social media platforms including Twitter, Instagram, Facebook, and Reddit called *religiously hateful memes dataset*. The study identified religious-based images and employed the RexNeXT-152-based Masked R-CNN model for image data and the BERT-BASE model for textual data for the purpose of classification. The performance exhibits an accuracy rate of 70.60%. However, the study's limitation is it's dependence on a binary classification scheme with poor accuracy

measure.

Jadhav and Honmane [101] contributed on classification of memes sourced from social media platforms. The study focused on using textual and image data. To classify the cyberbullying, the authors developed a CNN model for image data and the Bi-LSTM model for textual data. In order to examine the efficacy of the EX-OR method, the image and text mode are combined using the late fusion methodology. For text classification, they employed a sequential model called Bi-LSTM, which obtained an average accuracy of 87%. Additionally, a convolutional neural network achieved an average accuracy of 35% for classifying images data. Ultimately, a late fusion of memes with EX-OR prediction of both text and image forms was completed, yielding an overall accuracy of 87%.

Kiela *et. al* [102] conducted a classification analysis on memes which were obtained from Facebook. The study mainly concentrated on analyzing both textual data and image data, utilizing the Convolutional Neural Network (CNN) model for image data and the Bidirectional Long Short-Term Memory (Bi-LSTM) model for textual data in order to perform classification. The performance demonstrates a precision level of 0.87.

Fang *et. al* [103] conducted a classification analysis on multi-modal hostile memes obtained from Facebook AI. In this work, the authors focused on classifying both textual data and image data by utilizing the Inception V3 model for image data and the BERT model for textual data. This work used the text modality to enhance the semantic comprehension of the image modality. In particular, this study suggested an auxiliary approach for multi-modal hate speech identification called image caption supervision (ICS), in which the image caption was intended to supervise the feature learning of images in order to gain a deeper comprehension of the semantic content. The efficacy of ICS was demonstrated by the suggested technique outperforming several state-of-the-art uni-modal and multi-modal baselines on the Facebook Hateful Memes dataset. The performance of models demonstrated with a 72.80% accuracy.

Chhabra *et. al* [104] suggested architecture called "multi-scale kernel attentive visual" (MSKAV) module that was employed an efficient multi-branch structure to extract distinctive visual information. In addition, MSKAV employed an adaptable receptive field by utilizing multi-scale kernels. MSKAV included a multi-directional visual attention module that identified and emphasized important spatial regions. The suggested model included a unique module called "knowledge distillation-based attentional caption" (KDAC). The system used a transformer-based self-attentive block to extract distinctive elements from meme captions. The accuracy scores acquired through extensive experimentation on



the MultiOff, Hateful Memes, and MMHS150K datasets, which were multi-modal hate speech benchmarks, were 0.6250, 0.8750, and 0.8078, respectively. Furthermore, it achieved notable AUC scores of 0.6557, 0.8363, and 0.7665 on the three datasets, respectively, surpassing state-of-the-art multi-modal hate speech recognition models.

Hossain *et. al* [105] introduced a framework that employed the weighted ensemble technique to allocate weights to the visual, textual, and multimodal models involved. The framework utilized advanced visual models like as VGG19, VGG16, and ResNet50, as well as textual models like multilingual-BERT, multilingual-DistilBERT, and XLM-R. Additionally, two fusion methodologies, namely early fusion and late fusion, were employed to integrate the visual and textual characteristics in order to construct the multi-modal models. The evaluations shown that the proposed weighted ensemble technique enhanced the performance compared to the examined uni-modal, multi-modal, and ensemble models. The results demonstrated that the suggested method attains greater performance on two multilingual benchmark datasets (MultiOFF and TamilMemes), with weighted f1-scores of 66.73% and 58.59% by applying the best visual (VGG19), textual (mdistilBERT), decision fusion (VGG19 + m-distilBERT), and feature fusion (VGG19 + m-distilBERT) models respectively.

After studying on these research, we summarized them and presented them into table. The table 3 below shows the summarized current state-of-the art of binary class classification of multi-modal data.

Table 3: Summary of existing literature on social media cyberbullying binary classification

Author Name	Dataset	Model Name	Accuracy
Chandrasekaran <i>et al.</i> [31]	Benchmark dataset	SSA-DBN model	Accuracy: 99.983%
Hani <i>et al.</i> [106]	Formspring messages	Neural Network	92.8% accuracy
Dadvar <i>et al.</i> [32]	Formspring, Wikipedia, Twitter	Various DL models	0.76 discrimination score
N. K. Singh <i>et al.</i> [33]	Twitter dataset	Various ML and DL models	F-1 score of 0.92
Alotaibi <i>et al.</i> [34]	Twitter data	BiGRU, Transformer, CNN	88% accuracy

Continued on next page

Table 3: Summary of existing literature on social media cyberbullying binary classification (Continued)

Author Name	Dataset	Model Name	Accuracy
Kumari <i>et al.</i> [93]	Facebook, Twitter, Instagram	VGG-16, CNN	F1-score of 0.78
Kumari <i>et al.</i> [94]	Facebook, Twitter, Instagram	CNN	F1-score of 0.74
Singh and Sharma [95]	Twitter, ADIMA	BiLSTM-AHCNet, Tuned AEBO	Accuracy: 98.23%, F-measure: 98.22%
Ilavarasan <i>et al.</i> [96]	Facebook, Twitter, Instagram	RoBERTa, Xception	F1-Score: 80%
Sing <i>et al.</i> [97]	YouTube	SVM, CNN	85% accuracy
Paul <i>et al.</i> [98]	Vine platform	Residual Bi-LSTM	F-measure of 0.75
Romim <i>et al.</i> [99]	YouTube, Facebook comments	SVM	87.5% accuracy
Koshy and Elango [107]	Twitter	RoBERTa, ViT	94-98% accuracy
Idrizi and Hamiti [100]	Facebook, Instagram	GCN, Melfrequency cepstrum	85% accuracy
Ibanez <i>et al.</i> [54]	Tik Tok	SVM, Logistic Regression, Random Forest	78.5% accuracy
Hamza <i>et al.</i> [38]	Twitter, Instagram, Facebook, and Reddit	RexNeXT-152-based Masked R-CNN, BERT	70.60% accuracy
Kiela <i>et al.</i> [102]	Facebook AI memes	CNN, Bi-LSTM	87% precision
Fang <i>et al.</i> [103]	Facebook AI	Inception V3, BERT	72.80% accuracy
Mollas <i>et al.</i> [108]	Reddit and YouTube	DistilBERT, BiLSTM	accuracy 80.36%

Continued on next page

Table 3: Summary of existing literature on social media cyberbullying binary classification (Continued)

Author Name	Dataset	Model Name	Accuracy
Ahmadinejad <i>et al.</i> [36]	twitter	RoBERTa	99.70%

### 3.2 Existing Research on Applying DL for Multi-Classification

In this section, we present existing research on applying deep learning to cyberbullying multi-class and multi-label classification. We have tried to figure out which models used, which fusion modules had been applied, and how the data was labeled so that they worked best for multi-modal data.

Titli *et. al* [27], implemented a deep learning model named bengali BERT to classify multi-classes of the cyberbullying on bengali language data. The researchers used YouTube textual comments dataset, which was the same dataset as previous studies in [99, 109]. This dataset contained several classes such as religious, sexual, linguistic, political, personal, and crime-related content. In addition, the dataset contained data related to offensive text, including personal, geographical, religious, and crime-related offensive content, as well as content related to entertainment, sports, memes, and Tik-Tok. The results of developed bengali BERT model demonstrated the best level of accuracy, reaching 0.706, and a weighted F1-score of 0.705.

Haque *et. al* [59] performed a classification analysis on Bengali social media comments on bengali language data. In this study, the authors focused exclusively on textual data obtained from comments on Facebook and collected about 42,036 comments. The study employed the deep learning models CNN and LSTM for the purpose of multi-classification. The authors categorized the data into several classes such as Political, Religious, Sexual, Acceptable, and Combined. The performance of CNN-based LSTM network, named as: CLSTM architecture exhibits an accuracy rate of 85.8% and an F1 score of 0.86.

Maity *et. al* [37] developed a multitask deep learning framework for the identification of cyberbullying, such as sentiment, sarcasm and emotion aware cyberbullying from multi-modal memes. In their study, the authors collected images and memes from *Twitter* and *Reddit* social site's memes. To scrape images, they used hashtags like MeToo, KathuaRapeCase, Nirbhya, Rendi, Chuthiya, and Kamini on Twitter and subreddits like Desimemes, HindiMememes, and Bakchodi on Reddit, resulting in

around 25000 images or memes. Various deep learning models such as BERT, ResNET-Feedback and CLIP-CentralNet were developed and trained using textual and visual data. The task of the sentiment-emotion-sarcasm-aware multi-modal cyberbully detection in a code-mixed scenario was introduced for the first time in their paper. To tackle this challenge, they developed a novel multi-modal memes dataset called MultiBully, annotated with labels for bullies, attitude, emotion, and sarcasm. The purpose of this annotation was to determine if this information could aid in more accurate cyberbullying detection. An attention-based multi-task multi-modal framework, CLIP-CentralNet, was developed as a new architecture for sentiment, emotion, and sarcasm-assisted cyberbullying detection. Their suggested model included ResNet, mBERT, and CLIP for effective representations of many modalities and support in learning generic features across several tasks. The newly created CLIP-CentralNet framework performed noticeably better than any single task and uni-modal models in their task. For the purpose of detecting cyberbullying, they achieved accuracy of 61.14% for textual data using BERT, GRU, and a fully connected layer, and 63.36% for image data using ResNet and a fully connected layer.

Kumari *et. al* [110] proposed a model that employed a Convolutional Neural Network (CNN) and Binary Particle Swarm Optimization (BPSO) to classify social media posts from platforms like Facebook, Twitter, and Instagram. The model categorized posts containing both images and written comments into three classes: non-aggressive, medium-aggressive, and high-aggressive. A dataset comprising symbolic images and their corresponding textual comments was created to validate the proposed model. The system employed a pre-trained VGG-16 model to extract the visual features of the image, while also utilizing a three-layered CNN to extract the textual data. The hybrid feature set, consisting of both picture and text features, was optimized using the BPSO algorithm to extract the most pertinent characteristics. The enhanced model, incorporating advanced features and utilizing the Random Forest classifier, achieved a weighted F1-Score of 0.74.

Barse *et al.* [111] identified cyber-trolling from social media. The dataset was gathered from various sources, including YouTube API, Twitter API, web scraping, and government sources. Their main goal was to apply the model to both text and video datasets. They developed various machine learning and deep learning techniques, including multi-modal approaches such as logistic regression, multinomial-NB, perception, random forest, bidirectional-LSTM model. The dataset was divided into topic-specific categories such as misogyny, sexism, racism, xenophobia, and homophobia. Their obtained experimental results showed that the Random Forest model provided highest accuracy with 96.50% than other models, including Bidirectional LSTM model.

Mollas *et al.* [108] detected multi-label hate speech in their reseach. They presented "ETHOS" (multi-

labEl haTe speechH detection dataSet), a textual dataset based on comments from Reddit and YouTube that was validated through the use of the Figure-Eight crowdsourcing platform. It comes in two variants: binary and multi-label. For binary classification, they have got 80.36% accuracy from DistilBERT model as a highest accuracy, and for the accuracies of multi-label classification using BiLSTM were as follows: Violence: 50.86%, Directed versus Generalized: 55.28%, Gender: 70.34%, Race: 75.97%, National Origin: 67.88%, Disability Rate: 69.64%, Religion: 71.65%, Sexual orientation: 89.83%.

Ahmadinejad *et al.* [36], proposed machine learning and deep learning-based approaches for detecting cyberbullies on social media. They presented an annotated dataset containing 99,991 tweets. They showed result for both binary classification and multi-class classification. For binary classification, the classes were: cyberbully and non-cyberbully classes, and for multi-class classification, the classes were: non-cyberbullying, religion, ethnicity/race, and gender/sexual class respectively. They showed 99.70% accuracy for binary classification, and 99.80% accuracy for multi-class classification using RoBERTa model.

The table:4 contains a summary of the reviewed literature for multi-classification of cyberbullying, and the limitation of their research to identify the gap for our research.

Table 4: Summary of existing literature on social media cyberbullying classification on multi-classification

Author Name	Dataset	Model Name	Accuracy	Label Type	Limitation
Maity <i>et al.</i> [37]	Twitter and Reddit memes	BERT, ResNET, GRU	Text accuracy: 61.14% and Image accuracy 63.36%	multi-label	Performance outcome, accuracy rate is low
Tilti <i>et al.</i> [27]	YouTube comments	Bengali BERT	Accuracy: 70.6%, F1 score: 0.705	multi-class	Textual data only
Kumari <i>et al.</i> [110]	Facebook, Twitter, Instagram	CNN, BPSO	F1-Score of 0.74	multi-class	Focused on Aggression's level

Continued on next page

Table 4: Summary of existing literature on social media cyberbullying classification on multi-classification (Continued)

Author Name	Dataset	Model Name	Accuracy	Label Type	Limitation
Hossain <i>et al.</i> [105]	MultiOFF and TamilMemes	VGG19 and m-distilBERT	Weighted F1-scores of 66.73% and 58.59%	multi-class	Focused on Aggression's level
Barse <i>et al.</i> [111]	YouTube, tiktok, twitter and other social site	Random Forest	accuracy 96.50%	multi-class	Focused only textual data
Mollas <i>et al.</i> [108]	Reddit and YouTube	BiLSTM	accuracy 80.36%	multi-label	Focused only textual data
Ahmadinejad <i>et al.</i> [36]	Twitter	RoBERTa	99.80% accuracy for multi-class classification	multi-class	Focused only textual data

### 3.2.1 Identifying Gap of The Research on Cyberbullying of Social Media

All the above mentioned existing literature on multi-class classification of cyberbullying with multi-modal data using various deep learning models is summarized and presented in table:4. Despite the fact that there is a lot of research has been done on cyberbullying thus far, the majority of the identified cyberbullying classified as binary (see subsection:3.1, and table:3). We also can see there are some research: [27, 36, 59, 108, 110, 111] that worked on multi-classification, but only on textual data. In [37], Maity *et al.* showed multi-labeled classification on multi-modal data, but the study did not ended up with good outcome. Whereas Hossain *et al.* [105] showed result on muti-classification on multi-modal data, the study only focused on aggression's level (see section:3.2, table:4).

Eventhough various researchers have worked in the field of social media cyberbullying for the classification of multi-modal data, the drawback is that the majority of research has focused on either detecting or classifying it using a binary classification system. Several types of cyberbullying have been identified by Van *et al.* [112] [39], but there has been no multi-class classification of cyberbul-

lying on multi-modal data on that basis. Thus, in this thesis, we will explore the useful transformer architectures with multi-modal data of social media cyberbullying of multi-class classification.

## 4 Research Methodology

This section explains the research methodology used to accomplish this thesis's objective. First, we start with describing the datasets and how data pre-processing was done in the subsection 4.1. Then, we will detail the architectures of the used deep-learning models for the multi-class cyberbullying classification in subsection:4.4, 4.5, and 4.6 for text, image, and multi-modal data respectively.

### 4.1 Datasets

In this thesis, two datasets were used. The first dataset was public dataset (see subsection:4.1.1), and the second dataset was private (see subsection:4.1.2). Public dataset contains memes (multi-modal data) and private datasets contain three varieties of data i.e., textual data, image data, and memes as multi-modal data (text inside the image data). Each dataset has been classified into four classes of cyberbullying. These classes are: non-bullying, defaming, offensive language, and aggressive. We have explained in more details about this classes and classifications in section:4.2.1

#### 4.1.1 Public Dataset Collection

The public dataset was downloaded from the existing studies on cyberbullying classification [37, 38]. In [38], the dataset used was named as "religious hateful meme" dataset. This dataset was collected from several social media platforms including Twitter, Instagram, Facebook, and Reddit and contains data related to religiously hateful memes with 2000 images. In [37], the dataset used was called as "sentiment-emotion-sarcasm-aware" dataset. This dataset was collected from social media platforms like Facebook, Twitter, and Reddit. They mainly focused on Twitter and Reddit. To obtain hashtags, the authors scraped images from Twitter and subreddits from Reddit which resulted in approximately twenty five thousands memes and images. The dataset contained multi-modal data related to sentiment, emotion and sarcasm-aware cyberbullying.

In this thesis, the datasets from [37, 38], were considered and combined into one dataset, and named as *Public Dataset*. Table:5 shows multi-modal data distribution of the public dataset. In total, the dataset contains twenty seven thousands data of multi-modal data. The reason of working with this dataset was because the "religiously hateful memes" dataset was readily accessible bt public and includes hateful content. Whereas, another dataset "Religiously Hateful Memes" offers a valuable opportunity to analyze and comprehend the scope of hatred on online platforms. This dataset enables us to look at the content and context of hateful memes. The dataset allows for the identification of patterns and trends in the creation and spread of hateful cyberbullying. On the other side,



the “sentiment-emotion-sarcasm-aware” dataset was selected due to its multi-label structure and public availability. Dataset includes sentiment, emotion, and sarcasm-related data with mentioning the level of each features. One column, for instance, is labeled "Sarcasm," indicating that the dataset may be used to identify subtle linguistic usage. With a range of "Harmless" to "Partially-Harmful" to "Very-Harmful," the Harmful Score appears to assess the possible harm of the content, while target designates whether the harm is intended for a specific person, group, or society. This will make it easier to classify our public dataset into the different classes, because the combined dataset had been labeled into four classes which is defined in Section:4.2.1.

Table 5: Summary of Publicly Collected Dataset

Dataset Name	Total Multi-modal Data
Religiously Hateful Memes [38]	2000
Sentiment-Emotion-Sarcasm-Aware [37]	25000

The figure 11 shows an example of the multi-modal data from public dataset related to cyberbullying, where meme image is showing offensive behavior known as cyberbullying.



Figure 11: An example of meme (multi-modal data) from Public Dataset

#### 4.1.2 Private Dataset Collection

Private dataset contains around twelve thousand textual data, around one thousand image data that contains both image and multi-modal data content (such as: memes), which were self collected in this thesis. The dataset was collected from Facebook<sup>ix</sup>, Instagram<sup>x</sup>, YouTube<sup>xi</sup>, and TikTok<sup>xii</sup> short videos' comments. Comments from these above mentioned platforms had been extracted by using a tool called APIFY<sup>xiii</sup> for Facebook and YouTube short video's comments, TKCommentExport<sup>xiv</sup> tool for TikTok's comments, IGCommentExporter<sup>xv</sup> tool for Instagram reels comments. The dataset contained

<sup>ix</sup><https://www.facebook.com/>

<sup>x</sup><https://www.instagram.com/>

<sup>xi</sup><https://www.youtube.com/>

<sup>xii</sup><https://www.tiktok.com/>

<sup>xiii</sup><https://apify.com/>

<sup>xiv</sup><https://tkcommentexport.extensionsbox.com/>

<sup>xv</sup><https://chromewebstore.google.com/detail/igcommentexporter-export/ehaaocfdhppmemaeeedemaokjooldgm>

text, images and memes (multi-modal) data related to cyberbullying. Figure:12 refers the private dataset collection process, and figure:13 shows an example of the meme (multi-modal data) from private dataset related to cyberbullying.

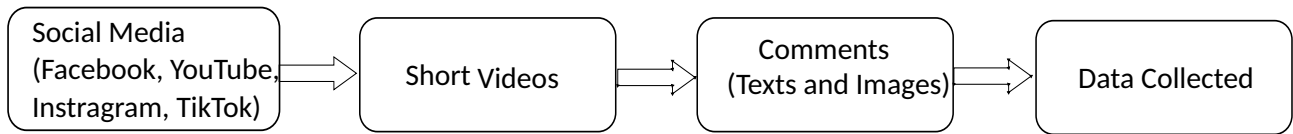


Figure 12: Private Data Collection Process

Table:6 contains total data distribution of the private dataset. There are almost twelve thousand text data and one thousand image and memes data. The reason of having only one thousand visual data is that, in each comment section, there had more textual comments than the visual (image and memes) comments.

Table 6: Total data distribution of private dataset

Dataset Name	Total Text Data	Total Images and Memes Data
Private Dataset	12000	1000



Figure 13: An example meme data from collected private dataset

## 4.2 Data pre-processing

The data preprocessing techniques for text, image, and meme (multi-modal) data for both public and private datasets are covered in this section. We describe our method for preprocessing the data and how we used feature extraction to build the model.

Data preprocessing is necessary because raw data frequently contains errors, inconsistencies, and missing values, which can result in inaccurate predictions when used directly in machine learning models [113]. Data preprocessing aims to convert raw data into a clean and organized format that is better suited for analysis. This process ensures that the model is trained on high-quality data, resulting in more consistent and accurate results [114]. However, these steps have been followed for preprocessing the data: Data categorization (classify data into multiple classes), data cleaning (dataset

cleaning, text cleaning, image cleaning), data augmentation ( image data augmentation), and data sampling (text data sampling, image data sampling). Each of these pre-processing steps helps to refine the dataset, ensuring that the data fed into the machine learning model is of the highest quality. Cleaned, integrated, and appropriately transformed data can reveal underlying patterns more clearly to algorithms, resulting in more accurate predictions. Data pre-processing can improve model efficiency and interpretability by reducing redundancy and simplifying features.

#### 4.2.1 Data Preprocessing for Textual Data

In text data preprocessing, several critical processes were involved to improve the quality and relevancy of textual information. Figure:14 shows all the steps involved in data pre-processing for textual data. We have categorized the data into four classes, cleaned the data and dataset, using augmentation and sampling method in the following way.

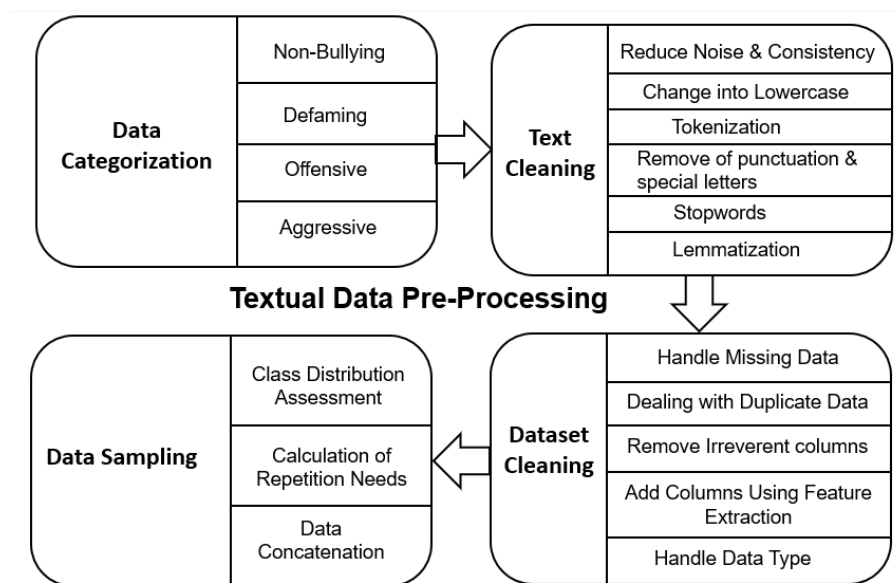


Figure 14: The Overview of the pre-processing pipeline for the both public and private datasets' text data

- **Data Categorization:**

Initially, each dataset (both public and private dataset) has been classified into four classes for cyberbullying. These classes are: non-bullying, defaming, offensive language, and aggressive. The explanation of these classes are given below.

1. Non-Bullying (class-o): The data does not include any content that is insulting, defamatory, offensive, or contains threatening or aggressive language [25].

2. Defaming Cyberbullying (class-1): Defaming cyber-bullying refers to behaviors in which individuals insult or defame another. It encompasses the act of expressing offensive comments, disseminating untrue or harmful information about an individual, or participating in the defamation of someone's character. This type of cyber-bullying is specifically targeted at damaging an individual's reputation and self-worth [39, 112].
  3. Offensive Language Cyber-bullying (class-2): Offensive language cyber-bullying include situations in which individuals employ derogatory language such as "f\*cker," "bitch," and "dog" to target someone. This form of cyber-bullying is distinguished by the utilization of offensive, derogatory, and demeaning words, specifically targeted at undermining the person's honor and self-esteem [39, 112].
  4. Aggressive Cyber-bullying (class-3): Such kinds of data refers to the act of making direct threats and displaying abusive conduct towards an individual. It encompasses explicit expressions of hurt, hateful comment, violent comment, threatening comment , and aggressive comment. This type of cyber-bullying is characterized by its confrontational and intimidating demeanor, with the intention of inducing fear in the target [112, 115].
- **Data Cleaning:** The next step in the data pre-processing is data cleaning. In this step, we cleaned the textual data. First, we reduced all text to lowercase, removed leading and trailing spaces, and replaced newline characters with spaces. Then it removed non-alphabetic and non-ASCII characters. Then it removed the URLs from the text. A regular expression tokenizer was used to break down the text into individual words. Common stopwords were then removed, but a custom list was created to exclude specific words from the default English stopword list. Single-character words were also eliminated. The remaining words were rejoined into a single string. Finally, each word was lemmatized to its root form. Furthermore, dealing with numerical values and rectifying duplicate, missing, noise, irreverent data resulted in a cleaner and more comprehensive dataset. Figure:15 shows original the public dataset's first few rows and it's preprocess data has shown in figure:16.

	Img_Name	text	label
0	2857.jpg	-5CR RELIEF FUND TO MAHARASTRA GOVT. - 1ST FU...	0
1	3983.jpg	a Russian businessman who earns more than \$20...	0
2	709.jpg	beingmoron@beingmoron Someone trying to build...	0
3	226.png	i thought i you said netflix & chill no i sai...	0
4	2245.jpg	instead of buying children all the things you...	0

Figure 15: Text Data before Pre-Processing

Img_Name	text	label	lower_case	alphanumeric	without-link	Special_word	stop_words	short_word	string	Text
0 2857.jpg	-SCR RELIEF FUND TO MAHARASTRA GOVT. -1ST FU...	0	-scr relief fund to maharashtra govt. - 1st ful...	cr relief fund to maharashtra govt st ful...	cr relief fund to maharashtra govt st ful...	[scr, relief, fund, to, maharashtra, govt, 1st, ful...	['scr', 'relief', 'fund', 'maharashtra', 'govt'...	[scr, relief, fund, maharashtra, govt, 1st, ful...	5cr relief fund maharashtra govt 1st fully covi...	5cr relief fund maharashtra govt 1st fully covi...
1 3983.jpg	a Russian businessman who earns more than \$20...	0	a russian businessman who earns more than \$20 ...	a russian businessman who earns more than ...	a russian businessman who earns more than ...	[a, russian, businessman, who, earns, more, th...	['russian', 'businessman', 'earns', '20', 'mil...	[russian, businessman, earns, 20, million, per...	russian businessman earns 20 million per year ...	russian businessman earns 20 million per year ...
2 709.jpg	beingmoron@beingmoron Someone trying to build...	0	beingmoron@beingmoron someone trying to build ...	beingmoron beingmoron someone trying to build ...	beingmoron beingmoron someone trying to build ...	[beingmoron, beingmoron, someone, trying, to, ...	['beingmoron', 'beingmoron', 'someone', 'tryin...	[beingmoron, beingmoron, someone, trying, buil...	beingmoron beingmoron someone trying build con...	beingmoron beingmoron someone trying build con...
3 226.png	i thought i you said netflix & chill no i sai...	0	i thought i you said netflix & chill no i said...	i thought i you said netflix chill no i said...	i thought i you said netflix chill no i said...	[i, thought, i, you, said, netflix, chill, no, ...	['thought', 'you', 'said', 'netflix', 'chill'...	[thought, you, said, netflix, chill, no, said, ...	thought you said netflix chill no said here ta...	thought you said netflix chill no said here ta...
4 2245.jpg	instead of buying children all the things you...	0	instead of buying children all the things you ...	instead of buying children all the things you ...	instead of buying children all the things you ...	[instead, of, buying, children, all, the, thin...	['instead', 'buying', 'children', 'things', 'y...	[instead, buying, children, things, you, never...	instead buying children things you never you s...	instead buying child thing you never you shoul...

Figure 16: Text Data after Pre-Processing

- Data Sampling:** After categorized the data into four classes, in our both datasets, we had imbalanced classes in text data, which could lead to biased models favoring the majority classes, so a targeted *sampling* strategy was required. We used *oversampling the minority class* method to balance the class distribution. The steps used were as follows:
  - *Class Distribution Assessment:* First, the dataset's existing class distributions were evaluated to determine the maximum size.
  - *Calculation of Repetition Needs:* For each class, the augmentation function computes the number of repetitions required for each data point in order to approximate the size of the largest class. This was accomplished by dividing the maximum class size by the size of the current class, then subtracting one to account for the original set of samples.
  - *Data Concatenation:* Each class was then augmented to its calculated size, and the original and augmented datasets were combined to create a balanced dataset.
- Feature Engineering:** For the text data, the column containing the "text" has been designated as the independent variable, while the "label" column has been used as the dependent variable on column that containing the text. The categories *Non-Bullying* have been assigned a score of class 0, *Defaming* a score of class 1, *Offensive* a score of class 2, and 'Aggressive' a score of class 3 using one-hot encoding process.

#### 4.2.2 Data preprocessing for Images

In image data preprocessing, several critical phases were involved in the process to improve the quality and relevancy of the images, preparing them for effective model training. Figure:17 outlines all the steps involved in image data pre-processing.

- Images data categorization:** Initially, Public dataset and Private datasets' images have been categorized into four different classes i.e., Non-Bullying (class-0), Defaming (class-1), Offensive

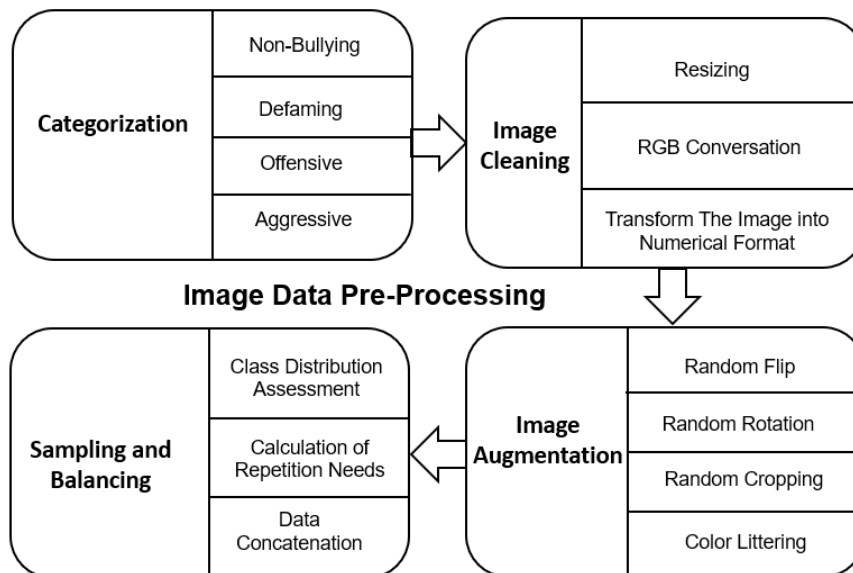


Figure 17: The overview of the pre-processing pipeline for both public and private datasets' images

(class-2), and Aggressive (class-3) by studying from following research: [116–118], and AI tools: ChatGPT.

- Categorization of images from public dataset:
  - Non-Bullying (class-0): Normal image, which does not contains any defaming, sexual, offensive, aggressive content (figure:18).
  - Defaming (class-1): Contains sexual, nudity content (figure:19).
  - Offensive (class-2): Showing middle finger (figure:11).
  - Aggressive (class-3): Beating someone, showing weapon to someone (figure:13).
- Categorization of images from private dataset:
  - Non-Bullying (class-0): Normal image, which does not contains any defaming, sexual, offensive, aggressive content (figure:18).
  - Defaming (class-1): Contains sexual, nudity content (figure:19).
  - Offensive (class-2): Showing middle finger, mixing other creature's face into people's face (figure:11, 18).
  - Aggressive (class-3): Beating someone, showing weapon to someone (figure:13, 21).

2. **Scaling and Normalization:** In the next step, *scaling* and *normalization* were used to guarantee that all photos have uniform dimensions and pixel intensity values, decreasing computational complexity and enhancing training convergence. So, all images were resized to 224x224 pixels via a transformation pipeline. This resizing not only preserves consistency but also reduces



Figure 18: An example of Class-0 image

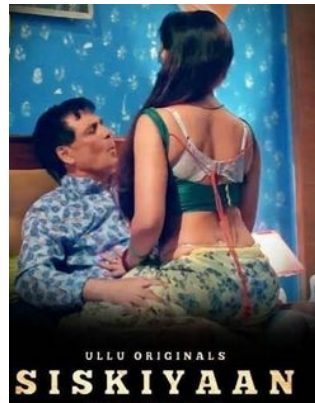


Figure 19: An example of Class-1 image



Figure 20: An example of Class-2 image



Figure 21: An example of Class-3 image

computational complexity by standardizing the amount of pixel data in each image. Color channel handling, such as converting images to RGB was also used as aids in focusing on key information.

3. **Data Augmentation:** In the next process, *data augmentation* techniques were used to increase the size of the dataset artificially and improve model generalization as follows.

- *Rotation:* It entails the act of turning images at varying angles, enabling the model to identify things from diverse perspectives.
- *Horizontal and vertical flipping:* It replicated distinct viewpoints and perspectives by reflecting images.
- *Random cropping:* It involved extracting random sections of photos, which compels the model to learn from various spatial contexts.
- *Color littering:* It involved introducing random variations to color attributes such as hue and saturation.

4. **Sampling:** The dataset *sampling* was done using a methodical approach which is *oversampling the minority class*, in which the class with the most images sets the target for all other classes. If any class had fewer images than this target, additional images were generated using augmentation techniques until all classes have an equal number of examples. This process not only balances the dataset but also enriches it with diverse examples, assisting in the development of a strong model. Effective sampling techniques, such as using random splits to create training, validation, and test sets, ensure that the model can be evaluated on various subsets of data, reflecting its expected performance on unseen real-world data.

5. **Feature Engineering for image data:** For the image data, separate folder has been created for each classes, and images have been organized according to their respective labels as class 0, 1, 2, 3 for Non-bullying, Defaming, Offensive, and Aggressive cyberbullying. We have also used scaling and normalization process as described in *Scaling and Normalization: 2*.

### 4.2.3 Multi-Modal Data Preprocessing

Figure:13, and 18 presenting two example of multi-modal data, where image contains text. For the multi-modal data, at first, we needed to reduce the images' three color channels to one before utilizing `cv2.cvtColor`<sup>xvi</sup> method to convert it to grayscale as shows in figure:22 in preparation for Optical Character Recognition as OCR<sup>xvii</sup> method. By improving the contrast between the text and the background, this step helps to better distinguish the text. Next, `cv2.threshold`<sup>xviii</sup> method was used to apply binary thresholding, transforming the grayscale image into a binary image. Here, a threshold value of 240 determines which pixels were set to white (255) and black (0). This stage was essential for lowering background noise and raising text detection phase accuracy.



Figure 22: Greyscale image

Using an image path as input, the second step involved reading the image with OpenCV, applying the preprocessing function, and finally performing OCR with *Tesseract*<sup>xix</sup>. Tesseract's configuration parameters are designed to maximize the accuracy of recognition. In addition to a character whitelist, it specifies the OCR Engine Mode (`-oem`) and Page Segmentation Mode (`-psm`), which limit the characters Tesseract attempts to recognize. This improved speed and accuracy by limiting the OCR process to only take into account the alphanumeric characters listed. After that, the extracted text was changed to lowercase to preserve consistency and perhaps make other text processing tasks easier.

<sup>xvi</sup><https://pyimagesearch.com/2021/04/28/opencv-color-spaces-cv2-cvtColor/>

<sup>xvii</sup><https://pypi.org/project/pytesseract/>

<sup>xviii</sup><https://pyimagesearch.com/2021/04/28/opencv-thresholding-cv2-threshold/>

<sup>xix</sup><https://en.wikipedia.org/wiki/Tesseract>



Finally text can be extracted by using Tesseract's *image\_to\_string* function. Figure:23 shows the overall process of extracting text from images.

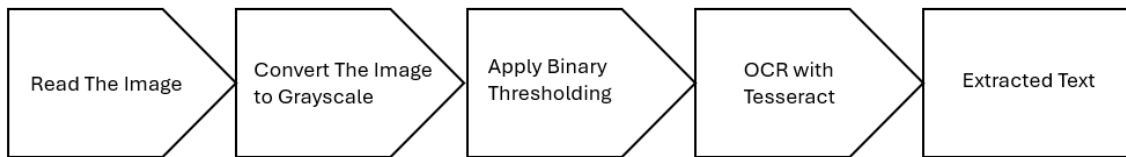


Figure 23: Process of Extracting Text from Images

After extracting the text from images, the extracted text's had been saved with text dataset, and pre-processed according to section:4.2.1, and the images also saved with image data and pre-processed according to section:4.2.2 as shown in figure:24, where it has been showed that how the multi-modal data preprocessed. From the figure, it can be understood that initially text was extracted from image and extracted text pre-processed with other text data and image preprocessed with other image data. However, figure:25 representing the pipeline of data-preprocessing for both public and private dataset, which was followed in this research.

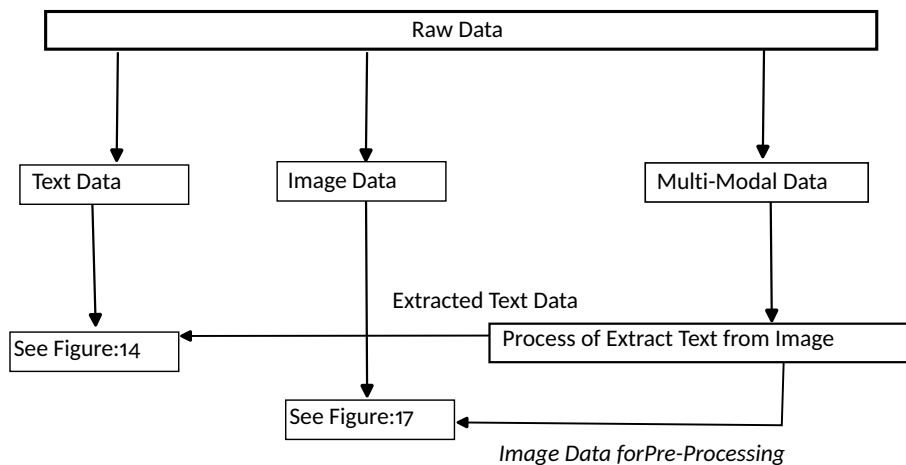


Figure 24: The of Multi-modal Data Preprocessing

**Data of each classes after pre-processing:** Table:7 explain the class distributions of both public and private dataset after pre-processing the text, image and multi-modal data. Since the public dataset contains memes data, which is multi-modal, we had to extract the text from the images according to subsection:4.2.3 and then need to do text pre-processing as described in subsection:4.2.1 and image pre-processing as described in subsection:4.2.2. Private data, contains both text, image and memes data. For the memes data, we extracted the text from images and processes according to subsection:4.2.3. Although initially public data had 25000 mutli-modal data, we ignored extreme suxiality and violent content for the privacy issue from the public dataset, and avoid the data which

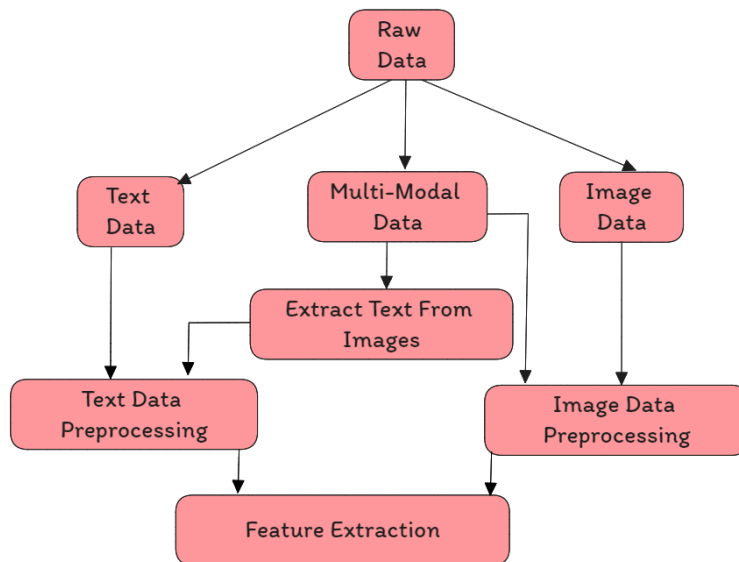


Figure 25: Complete Pipeline for Data Pre-processing.

was not in english language data, which was not related with our research and data categorization. After pre-processing, we ended up with 25811 public data and 32773 private data. As we can see from the table that, now every classes are almost balanced after data pre-processing.

Table 7: Total data distribution of dataset for each classes

Classes	Public Dataset		Private Dataset	
	Textual	Image ( image and memes )	Textual	Image ( image and memes )
Class 0	4439	2044	8130	257
Class 1	4418	2044	8128	257
Class 2	4410	2024	7952	257
Class 3	4368	2024	7536	256

#### 4.2.4 Feature Extraction

In this section, we have discussed that how we have used feature extraction method in the public and private dataset:

##### Feature Extraction for Text data:

The RoBERTa, DistilBERT, BERT tokenizer converted texts into a model-suitable format. Each text transformed into a sequence of tokens, which are then encoded as input IDs and attention masks. To ensure consistent input size, the tokenizer pads or truncates texts to a fixed length (in this case, 256 tokens). Along with input IDs, attention masks were generated to inform the model which parts of the token sequence should be addressed and which were simply padding.

### Feature Extraction for Image data:

Images first go through preprocessing to prepare them for processing. This includes resizing them to a standard size (224x224 pixels) and converting them into tensor format. After splitting each image into fixed-size patches, such as 16 by 16 pixels, the ViT model linearly converted these patches into patch embeddings. Within each patch, these embeddings capture local visual features.

### Feature Extraction for Multi-modal data:

The feature extraction process was used to extract and classify features from both textual and visual data using advanced deep learning algorithms as shown in figure:26.

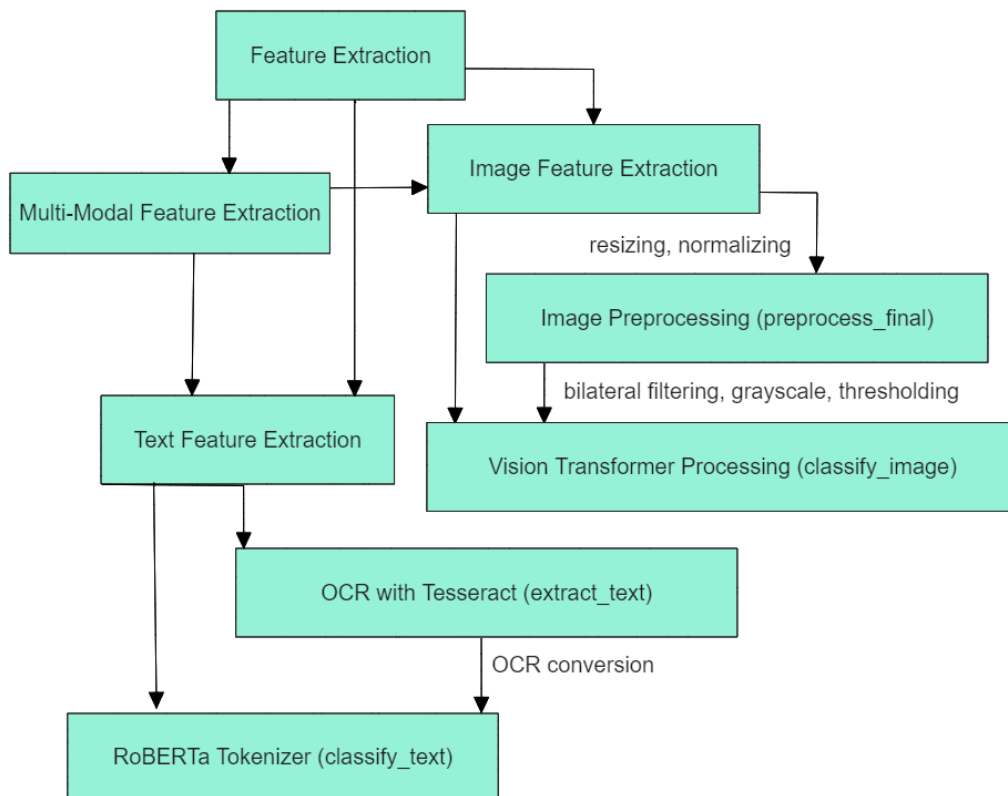


Figure 26: Pipeline for Feature Extraction.

Initially, we installed installs Tesseract-OCR, an open-source optical character recognition tool that extracted text from images. Pytesseract, a Python wrapper for Tesseract, also installed to make text extraction easier within the Python environment. To improve text visibility during preprocessing, images transformed using OpenCV, which included bilateral filtering, grayscale conversion, and thresholding. These steps were necessary for getting images ready for effective text extraction. The extracted text was then passed through a RoBERTa tokenizer, which converted it into a series of tokens that the RoBERTa model used for classification. Parallel to text processing, the script uses a Vision Transformer (ViT) for image classification. The ViT model, which was pre-trained on a large dataset, analyzed the images directly. The feature extractor converted the images into a tensor of pixel values,

which the ViT model then used to classify the image content.

#### **4.2.5 Split Data into Training and Validation Datasets**

We have taken independent column that contains text data, and label column as dependent data for training and validation the model for classifying text. Similarly, for the image data, we have taken the images folder's name:0, 1, 2, 3 as dependent variable that has been named according to it's label as mentioned in section:4.2.4 and taking the images inside these folder's as dependent variable for training, and validation the model.

To create an unbiased and generalizable cyberbullying classification model, the datasets were carefully divided into subsets for training, validation, and testing. We used a standard split ratio, allocating 80% of the data for training, allowing the model to learn and recognize patterns in cyberbullying comments. The remaining data was divided equally into two groups: 10% for validation, which is used to fine-tune model hyperparameters and prevent overfitting, and 10% for testing, which provides an unbiased evaluation of the model's performance.

Both the public and private datasets, which included textual, image and memes data, were pre-processed to ensure the input data's consistency and quality. The splitting was done at random to ensure that the data was diverse across all sets and to avoid any bias that could skew the model's learning.

### **4.3 Proposed Solution**

For classifying multi-class cyberbullying, we proposed solutions for both public and private dataset in this section. The first step is multi-modal data collection and which is explained in subsection:4.1, the next stage involves pre-processing of the multi-modal data and explained in subsection:4.2. Feature extraction is a necessary step that applied text, image, and multi-modal data for converting the data into machine readable format as described in subsection:4.2.4. Next, the subsequent stage involves developing deep learning models to train multi-modal data. The model is then tested to obtain the desired output i.e., multi-class cyberbullying classification.

The figure 27 shows proposed solution for the public dataset. The diagram illustrates an architectural design that exhibits a well-defined and sequential progression of data from input to classification. At the beginning, input receives three types of data: text, images, and text derived from images. After doing data preprocessing and feature extraction, the next step will be to apply the transformer mod-

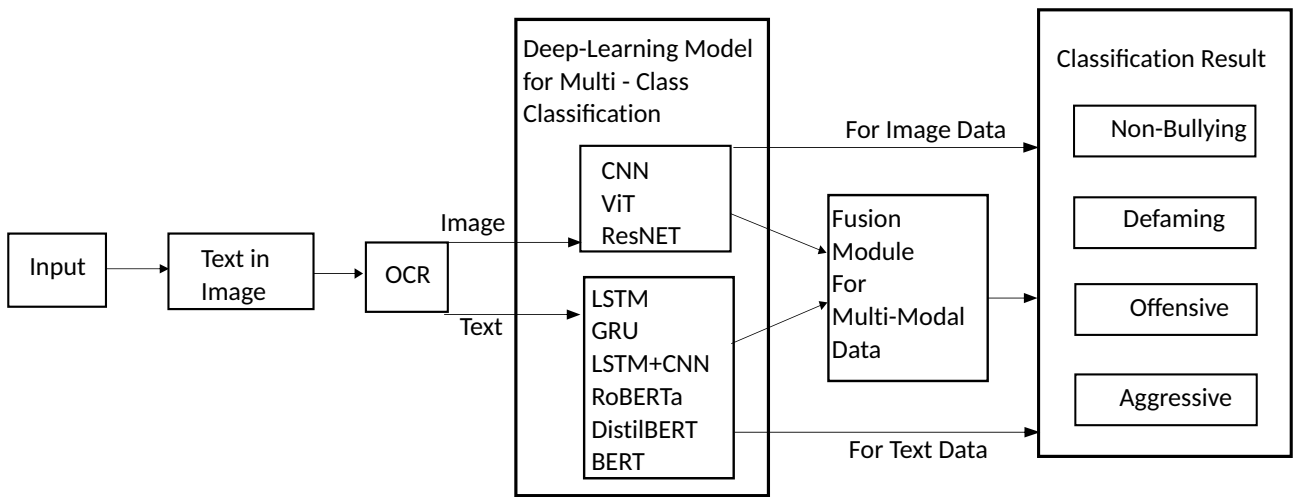


Figure 27: The proposed solution for public dataset

els for both textual and image data. For the text data in the public dataset, the four deep-learning models such as: *LSTM*, *Hybrid model(CNN+LSTM)*, *GRU*, *BERT*, *DistilBERT*, and *RoBERTa* models will be applied for analyzing the data. Further, image data will be used as input to the collection of image processing models, including the ResNet-50, *Convolution Neural Network (CNN)*, and *Vision Transformer (ViT)*. OCR is a technique used to extract text from the images which contain text. This extracted text data will be then trained using the RoBERTa model, and the images data will be trained using the ViT model. Later Hybrid (RoBERTa+ViT) model used for multi-modal data using the late fusion module.

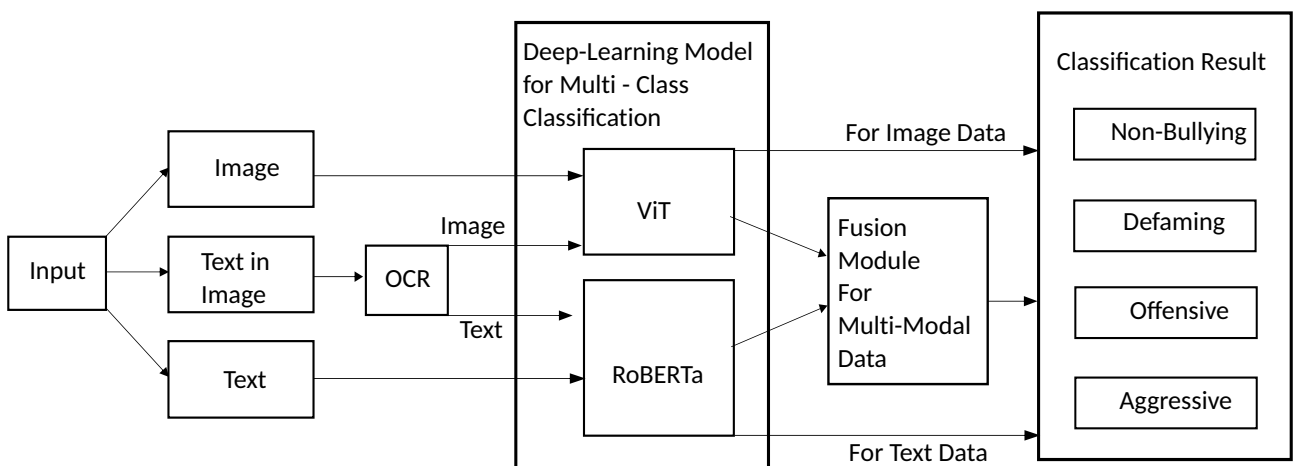


Figure 28: The proposed solution for private dataset

Figure:28 represent the proposed solution for the private dataset. for the private dataset, text data was processed using a RoBERTa model for text analysis. Whereas, input images were analyzed using image processing model i.e.,Vision Transformer (ViT) 4.5.3. For the multi-modal data, text was extracted from image, and extracted text data preprocessed with other text data and multi-modal image data preprocessed with image data. Later Hybrid (RoBERTa+ViT) model used for multi-modal

data using the late fusion module. The reason for not using same models as public data's has explained in the following chapter:5.

## 4.4 Network Architectures for Text Data Classification

For classifying multi-class cyberbullying using text data, we employed six deep learning models. These choices were made based on their popularity and use in the current state-of-the-art on the cyberbullying classification.

### 4.4.1 Hybrid(CNN+LSTM) Model

The first considered model was Hybrid(CNN+LSTM) model. Figure:29 shows the used hybrid model's (CNN+LSTM model) architecture for text data classification. For text classification tasks, the hybrid model presented combines layers of a convolutional neural network (CNN) with long short-term memory (LSTM). With the architecture's fast handling of sequential data, text inputs can be used to extract both local and global properties. Initially, the input words were converted into dense vector representations using an embedding layer. Capturing the semantic links between words requires the use of this layer. Next, we applied SpatialDropout1D, which provides regularization by dropping complete 1D feature maps at random.

After that, feature extraction was performed using a 1D convolutional layer (Conv1D) with ReLU activation. The most crucial information was preserved while the dimensionality of the features were decreased in the next MaxPooling1D layer. Bidirectional LSTM layers, which process the input sequences both forward and backward, were then incorporated into the model. Because it is bidirectional, the model was able to accurately represent long-term interdependence. After every LSTM layer, dropout layers were added to avoid over-fitting. In order to force the model to learn more resilient representations, these layers randomly discard a portion of the units during training. The output of the LSTM layers was reshaped into a format appropriate for Dense layers by a Flatten layer.

The last classification operation was then carried out by Dense layers with ReLU activation. To predict probabilities across the classes, the output layer makes use of a softmax activation function. The *Sparse Categorical Crossentropy* loss function and adam optimizer were used to compile the model during training. *ModelCheckpoint*, *EarlyStopping*, *ReduceLROnPlateau*, and other callbacks were introduced to track validation performance and modify the training procedure as necessary. All things considered, the hybrid model combines the advantages of LSTMs for capturing sequential dependencies and CNNs for feature extraction, making it an excellent choice for text classification problems.

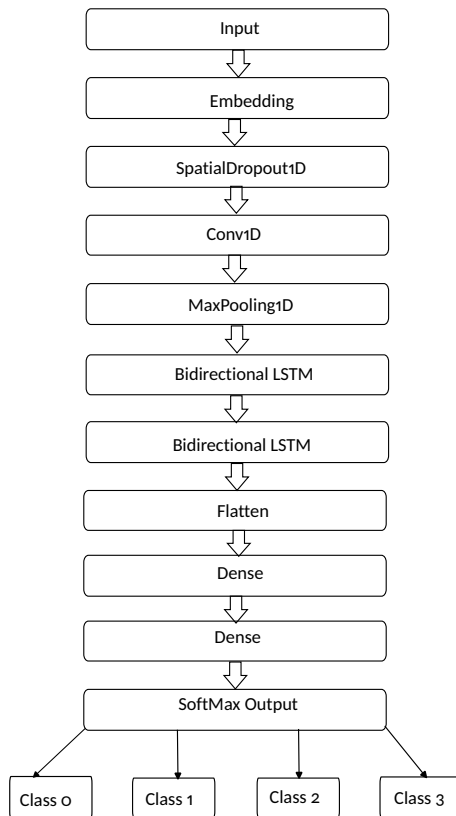


Figure 29: The Used CNN+LSTM Architecture for Text Data Classification

#### 4.4.2 LSTM Model

The second considered model was hybrid(CNN+LSTM) model. Figure:30 depicted the architecture of LSTM model for text data classification. Two layer LSTM model was used for public text data multi-class classification. The model's initial layer, called the embedding layer transforms the language input into a continuous, lower-dimensional vector space. In this design, a 128-dimensional vector represents each word. In order to help neural networks comprehend textual data, this layer was essential for capturing the semantic links between words. After the Embedding layer, a dropout layer with a dropout rate of 0.5 was added to reduce over-fitting. Dropout reduces the model's inclination to rely on particular features by arbitrarily removing entire 1D feature maps. Then, two LSTM layers were used by the model. With 50 units, the first LSTM layer gives sequences back to the next layer so that the latter can get all of the sequence information.

The goal of the second LSTM layer, which has 50 units as well, was to capture higher-level temporal representations devoid of sequence returns. To further minimize over-fitting, Dropout layers with a dropout rate of 0.5 were introduced between the LSTM layers. These layers helped the model to build more resilient representations by randomly setting some of the input units to zero during training. The model had a dense layer with 32 neurons and ReLU activation for classification after the

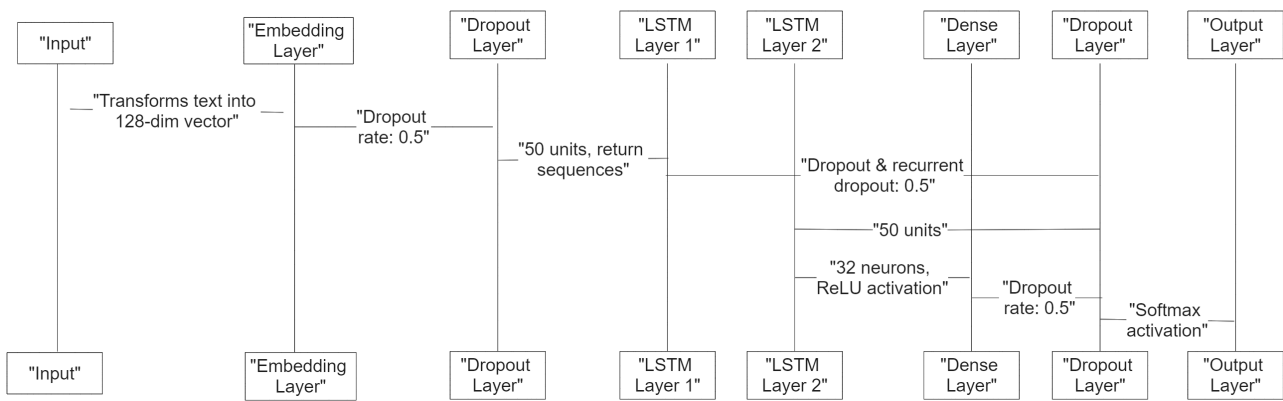


Figure 30: The used LSTM Architecture for Text Data Classification

LSTM layers. The learned features gain a degree of abstraction from this layer. To add even more regularization, a final Dropout layer with a dropout rate of 0.5 was placed before the output Dense layer.

A dense layer with a softmax activation function makes up the output layer, which makes it easier to categorize the dataset's multiple classes. The Sparse Categorical Crossentropy loss function and Adam optimizer were used to optimize the model during compilation. An *EarlyStopping* callback was utilized to restore the optimal weights based on validation loss and to monitor the overfitting risk. Overall, this model architecture efficiently handles multi-class classification tasks on textual data by combining dropout regularization with the ability of LSTM networks to capture temporal dependencies. Adding callbacks, like *EarlyStopping*, improves the model's performance and training-time generalization abilities.

#### 4.4.3 GRU Model

In the used GRU model as third considered model for text data, an embedding layer was the first layer in the model, and it transforms the input vocabulary into dense vectors with a fixed size. In this setup, a 128-dimensional vector represents each word. In order to analyze textual data in neural networks, this layer aids in capturing the semantic links between words. After the Embedding layer, a dropout layer with a dropout rate of 0.5 was added to reduce over-fitting. During training, dropout randomly removes neurons, which lessens the model's dependence on certain features and improves generalization. Two GRU layers are then used to process the sequence. With 50 units, the first GRU layer provides sequences to the next layer so it can get all of the sequence information.

The goal of the second GRU layer, which has 50 units as well, was to capture higher-level temporal representations devoid of sequence returns. To further prevent over-fitting, dropout layers with a dropout rate of 0.5 were introduced between the GRU layers. In order to encourage strong fea-



ture learning, these layers randomly change a portion of the input units to zero during training. The model had a Dense layer with 32 neurons and ReLU activation for classification after the GRU layers. The learned features gain a degree of abstraction from this layer. To add even more regularization, a final dropout layer with a dropout rate of 0.5 was placed before the output Dense layer. A dense layer with a softmax activation function makes up the output layer, which makes it easier to categorize the dataset's multiple classes. Figure:31 depicted the architecture of LSTM model for text data classification.

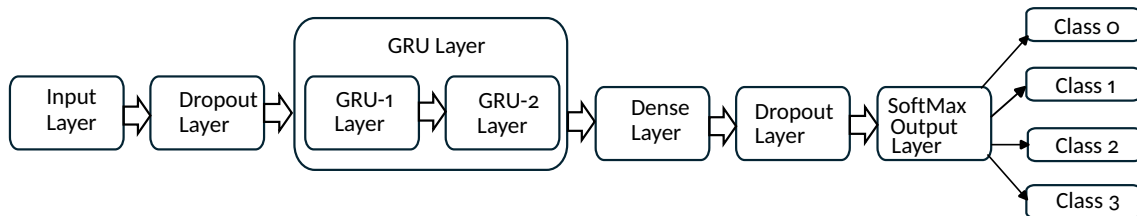


Figure 31: The Used GRU Architecture for Text Data Classification

#### 4.4.4 BERT Model

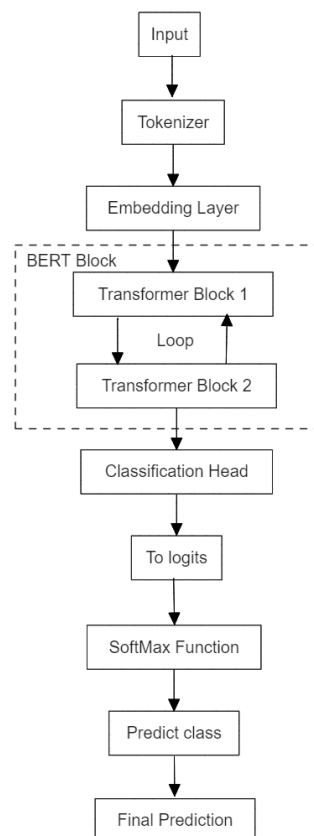


Figure 32: The Used BERT Architecture for Text Data Classification

The BERT model was the fourth considered model. In the BERT model architecture as shows in figure:32, raw text data was introduced into the system at the first layer. To process this data, a tok-

tokenizer was first used, which divided it into meaningful segments known as tokens, which were typically words or subwords. After passing through an embedding layer, each token is converted into a high-dimensional vector that includes both context and semantic meaning. The BERT model used a multi-layered structure, with each layer designed to iteratively refine the information. As the data flows through these transformer blocks, it was constantly processed, with each block contributing to a more refined understanding of the textual input based on both the context provided by surrounding words and the inherent meaning of each word.

The data arrived at the classification head after passing through all of the transformer blocks. This model component was responsible for translating the transformer blocks' complex representations into a simpler format known as logits. These logits were the unprocessed, raw predictions produced by the deep learning model's final layers. These logits were then converted to probabilities using the SoftMax function. This function was critical in classification tasks because it converted logits into probability distributions across predicted classes, allowing the most likely class to be determined for each input.

#### 4.4.5 DistilBERT Model

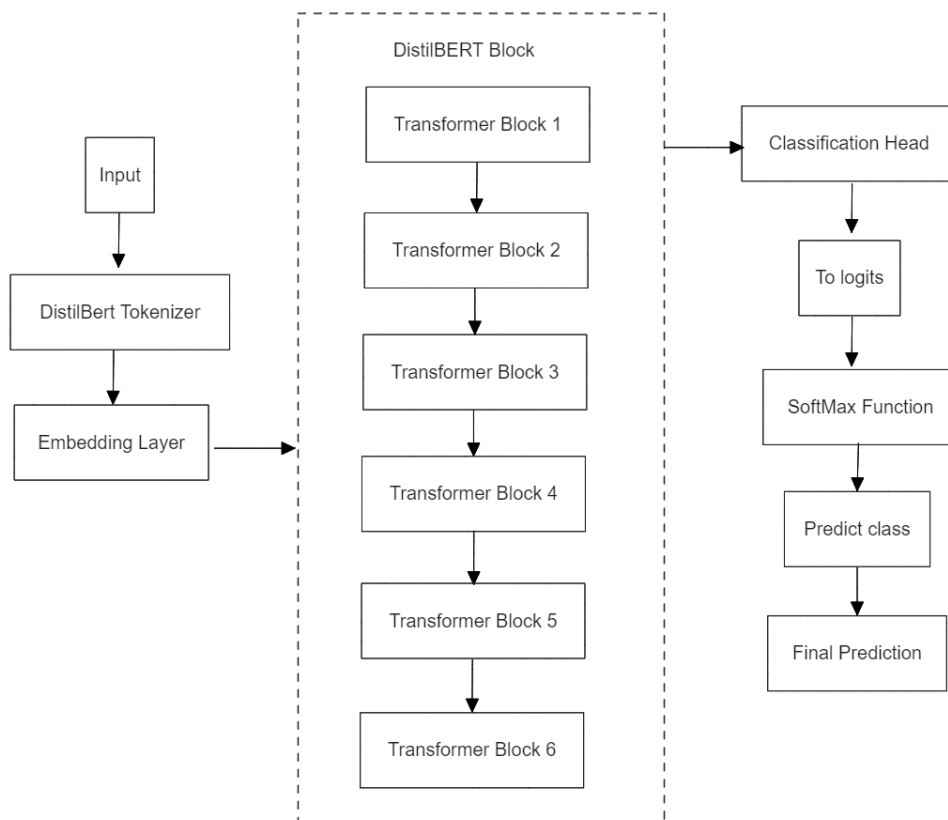


Figure 33: The Used DistilBERT Architecture for Text Data Classification

In the current architecture of DistilBERT model as fifth considered model for textual data as shows in figure:33, raw text data enters the system at the first layer. To process this data, a tokenizer is first used to separate it into meaningful segments known as tokens, which are usually words or sub-words. An embedding layer transforms each token into a high-dimensional vector that contains both context and semantic meaning. The model used in this setup was the DistilBERT, used six transformer blocks. As the tokenized data passes through these transformer blocks, it was continuously processed, with each block designed to iteratively refine the information. This refinement process improved the model's understanding of the textual input by leveraging both the contextual relationships established by adjacent words and each word's intrinsic semantic properties.

The data is routed through the transformer blocks before arriving at the model's classification head. This model component is critical because it converts the complex representations created by the transformer blocks into a simpler and more understandable format known as logits. These logits represent the unprocessed predictions made by the deep learning model's final layers. To complete the classification process, these logits were converted into probabilities using the SoftMax function. This function was important in classification tasks because it converts logits into a probability distribution across predicted classes, making it easier to determine the most likely class for each input based on the calculated probabilities. This enables the model to predict the class with the highest probability as the output for the given input text, which was an important step in determining the final prediction.

#### **4.4.6 RoBERTa Model**

The used RoBERTa model for textual data classification was sixth and last considered model, it is depicted in the figure:34. In this architecture, raw text data was added to the system at the bottom. A tokenizer was used to first process this data, dividing the text into meaningful chunks called tokens, which were typically words or subwords. After passing through an embedding layer, each token was transformed into a high-dimensional vector that includes context and semantic meaning. Twelve blocks are stacked on top of one another in RoBERTa models. These blocks refine information iteratively by passing on its output from one block to the next.

The classification head receives the data after it passes through the transformer blocks. This portion of the model was in charge of translating the intricate representations that the Transformer Blocks produce into a more straightforward format known as logits, which were unprocessed, raw predictions that a deep learning model produces in its last layers. The SoftMax Function was then used to convert these logits into probabilities. For tasks like classification, where we want to know

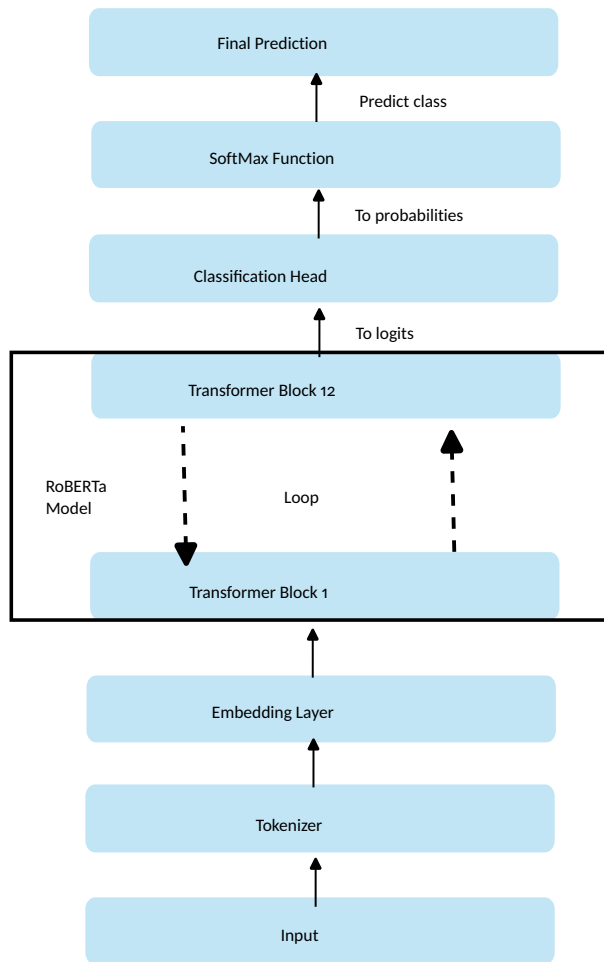


Figure 34: The Used RoBERTa Architecture for Text Data Classification

the probability that a given input belongs to a certain class, this function transforms the logits into a probability distribution over the predicted classes. And lastly, in the Final Prediction, the model predicts the class with the highest probability as the output for the given input text based on the probability distribution.

## 4.5 Network Architectures for Image Data Classification

Three deep learning models were utilized to classify cyberbullying based on multiple classes for image data.

### 4.5.1 ResNet-50 Model

ResNet-50 model was the first considered model for image data. In the ResNet-50 architecture ( see figure:35), as processed image data passes through the convolutional layers, each layer applied a set of filters to extract various image features, such as edges, textures, and complex patterns. These layers were intended to iteratively refine the information, improving the model's understanding of

the visual content by taking advantage of both the hierarchical nature of features in images and the spatial relationships between different objects within them. Once the image data passed through the convolutional layers, it reached the model's classification head. This component was critical because it converted the high-level feature representations generated by the convolutional layers into a simpler and more understandable format known as logits. These logits are raw, unprocessed predictions made by the deep learning model's final layers.

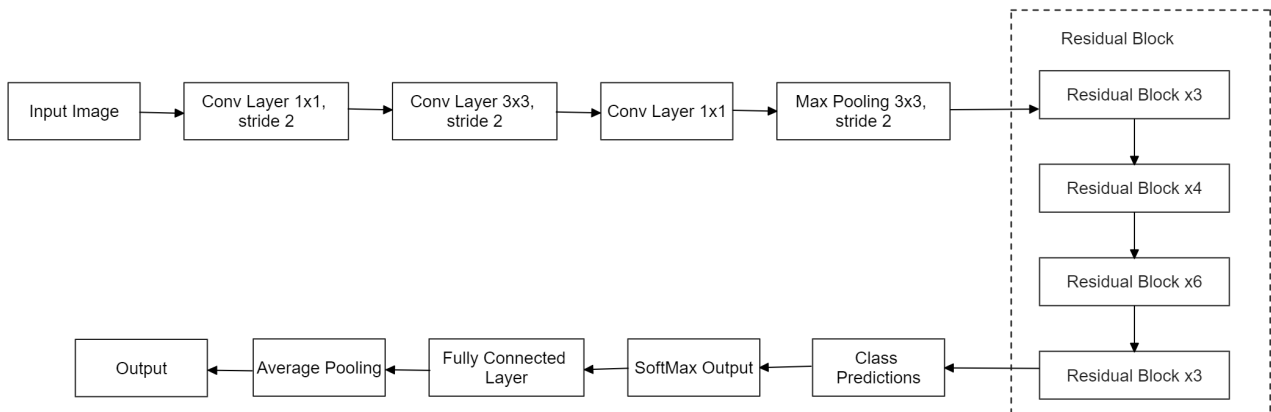


Figure 35: The Used ResNet-50 Architecture for Image Data Classification

To finish the classification process, these logits were converted into probabilities using the SoftMax function. This function was essential for classification tasks because it converts logits into a probability distribution across predicted classes, making it easier to determine the most likely class for each input based on calculated probabilities. As a result, the model predicts the class with the highest probability as the output for the given input image, which was an important step in determining the final prediction. This methodical approach enables the ResNet50 model to effectively classify images, demonstrating the power of convolutional neural networks in handling complex image data.

#### 4.5.2 CNN Model

The Convolutional Neural Network (CNN) architecture was the second considered model for image classification tasks, which is depicted in the figure:36. An image was first input into the network to start the process. The convolutional layer (Conv2d-1) was the first layer. It uses a number of filters to identify fundamental features like edges and textures. The Rectified Linear Unit (ReLU-1), a nonlinear activation function, was subsequently applied to the feature maps produced by the convolution. This adds non-linearity to the model, enabling it to learn increasingly intricate patterns.

Following the activation function, a Max Pooling layer (MaxPool2d-1) was employed to shrink the representation's spatial size. This reduces the number of parameters and computation in the net-

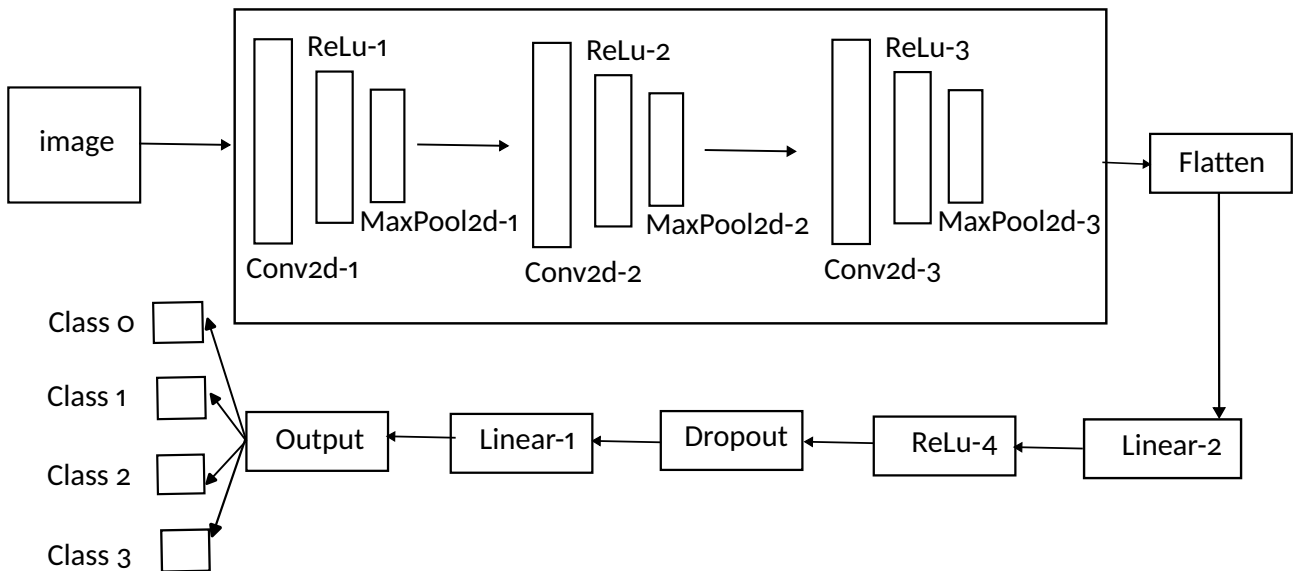


Figure 36: The Used CNN Architecture for Image Data Classification

work and aids in the management of over-fitting. Following processing, the data was run through two additional convolutional, ReLU activation, and max pooling layer sequences (Conv2d-3, ReLU-3, MaxPool2d-3 and Conv2d-2, ReLU-2, MaxPool2d-2). In order to identify more complex features in the image, these layers function gradually.

There were two linear (fully connected) layers in the network. The first, called Linear-2, reduces over-fitting by transforming the high-dimensional vector to an intermediate dimension (512 units). This was followed by ReLU-4, another activation, and a dropout layer with a dropout rate of 0.5. These features were mapped to the output classes, which correspond to the various dataset categories, by the last linear layer, Linear-1.

### 4.5.3 ViT Model

ViT model was the third and last considered model for image data. The model architecture that is offered makes use of the Vision Transformer (ViT) to enhance picture classification capabilities. The number of classes in the training dataset used to define the model's configuration at first, and the pre-trained ViT configuration *'google/vit-base-patch16-224-in21k'* was used to initialize the ViT model in particular for image classification. The model is then transferred, if possible, to the GPU to maximize computational efficiency. A Cross-Entropy Loss function was used for training, which is perfect for multi-class classification.

The Vision Transformer (ViT) neural network architecture for image classification is depicted in the figure:37. An input image was first split into patches, which were discrete, uniformly-sized portions of

the image. In the "Patch Extraction" stage, these patches were then flattened into one-dimensional vectors. In order to preserve the sequence order—which was essential for the model to comprehend the image layout—these flattened patches were then provided with positional information. "Add Position Embeddings" was the process of providing each patch with additional information to indicate its position in the original image. This embedding helped the Vision Transformer model understand the spatial relationships between patches, allowing it to reconstruct the image's original layout and structure during analysis. By combining positional embeddings with flattened patch vectors, the model processed the patches in a meaningful order, capturing spatial patterns required for accurate image classification.

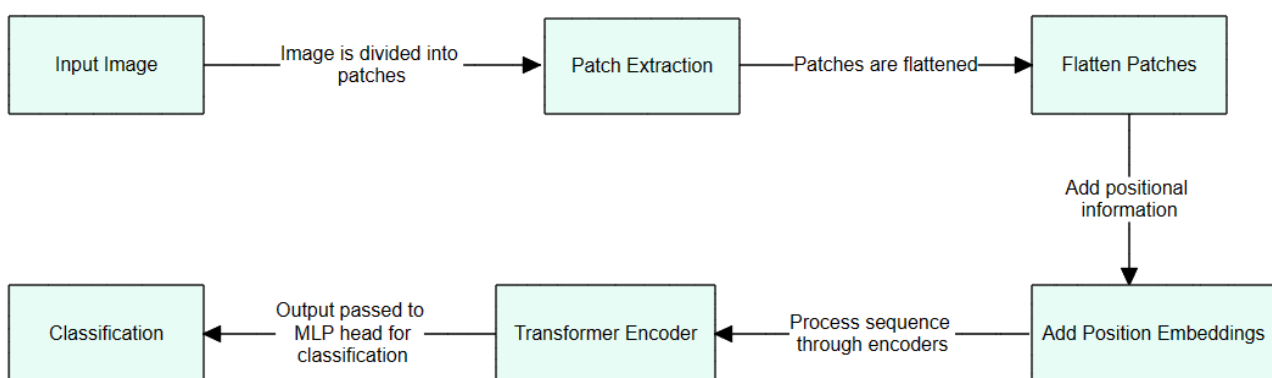


Figure 37: The Used ViT Architecture for Image Data Classification

Afterwards, the positionally embedded patches were fed into a stack of encoders from the transformer model called a Transformer Encoder. By processing these sequences using its self-attention mechanisms, the Transformer Encoder allows the model to focus on different areas of the image during prediction. Ultimately, an MLP (multi-layer perceptron) head receives the output from the Transformer Encoder and used it to classify the image into different categories by interpreting the encoded features. Fine-grained understanding and recognition are made possible by this architecture, which was taken advantage of the transformer's sequence handling capability to process images in segments rather than as a whole.

## 4.6 Network Architecture for Multi-Modal Data Classification

A hybrid (RoBERTa+CNN) model was used for multimodal data. If the input was received text data, RoBERTa model was applied on the given text data to classify the data following the section:4.4.6. If the input was received image data, ViT (section:4.5.3) model was applied on the given image data. However, when an image contained multi-modal data ( known as memes data), the first step was to extract the text as mentioned in section:4.2.3. This could include recognizing and isolating written

words or numbers from the visual background, allowing the textual content to be analyzed independently of the image. Then the extracted text was analyzed to determine its meaning using RoBERTa model of section:4.4.6. This analysis may focus on understanding the text’s language, context, or sentiment, allowing for more information about what the image depicts or was associated with. After that, image processing involved analyzing both the image and the text. This stage of the process focused on interpreting visual elements like objects, colors, and spatial relationships in the image using ViT model according to section:4.5.3. This analysis seeks to comprehend what the image represents on a visual level. To be more specific, we were using *Hybrid Model (RoBERTa + ViT)* to classify multi-modal data.

The process used late fusion module ( explained in section:4.6.1) in which the features of text and images were processed independently and then combined for prediction at a later time. This method uses separate models—RoBERTa (see section:4.4.6) for text and ViT ( see section:4.5.3) for images to process inputs that were text and image-based. The final prediction was then produced by combining the predictions from the two models. The process concluded once both the text and the image had been classified. Figure:38 shows the architectural in flowchart diagram of multi-modal data with hybrid (RoBERTa and ViT) model and late fusion for getting the final result for text and image label's cyberbullying.

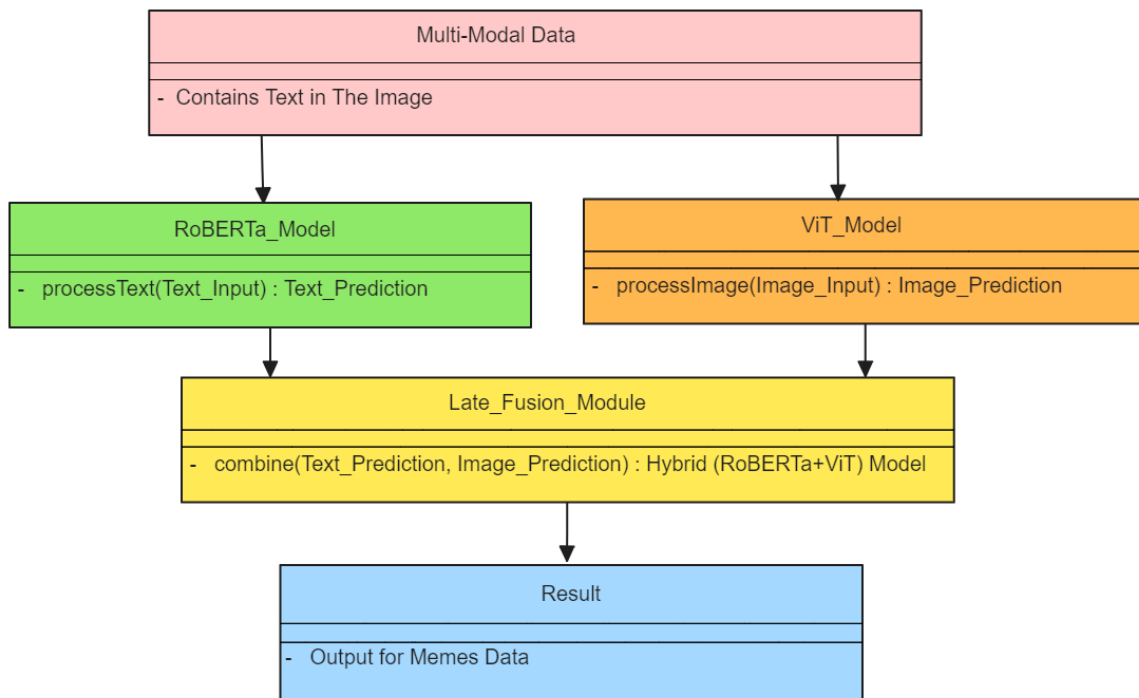


Figure 38: Architecture of Multi-Modal Data with Hybrid Model on Late Fusion Module



### 4.6.1 Late Fusion Module

We used the late fusion module in our thesis for multi-modal data. For classifying the multi-modal data, the late fusion module combined text and image data processing using separate hybrid Model (RoBERTa+ViT). This module performs decision-level fusion by combining the final predictions from two different models (RoBERTa for text and Vision Transformer for images) to provide a comprehensive assessment of the presence of cyberbullying. The decision to label the fusion as "late" stems from the fact that data integration occurs at the end of the process, after each input type was individually analyzed and classified.

Initially, the text input was processed using a pre-trained RoBERTa model, which allowed the system to extract semantic features and predict text labels. Images were processed simultaneously using a Vision Transformer (ViT) model, which extracted visual features and predicts image outcomes. Late fusion occurred when the predictions from both modalities were combined at the end to produce the final result. This combined approach enabled the system to capture information from both text and images which is depicted in figure:39.

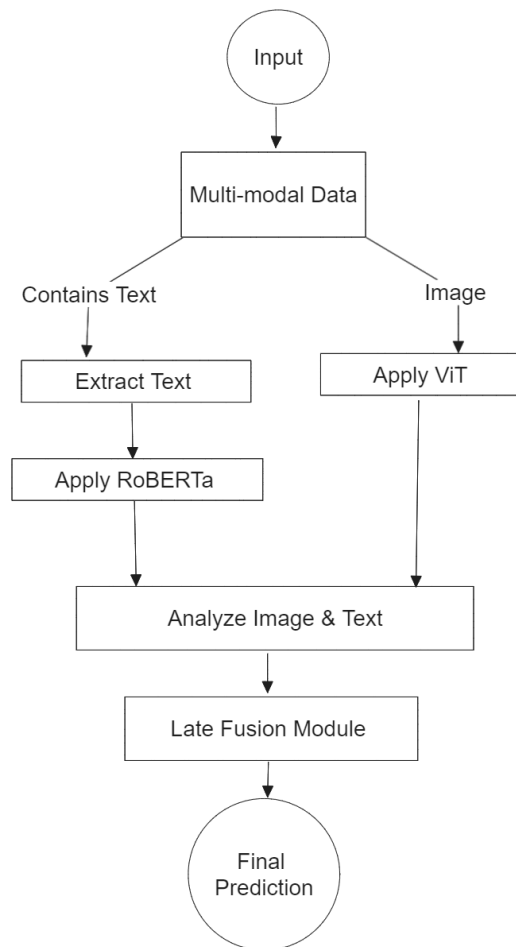


Figure 39: Process of Late Fusion Module

They key features of late fusion module for our work were as follows:

- *Integration*: Combines results from independent analyses on various types of data (text and images).
- *Decision-Making*: Decision-Making the final output based on the results of these individual analyses, using logic to resolve any discrepancies between them as pseudocode:1. The process begins with determining whether both text and image classifications (*text\_class* and *image\_class*) are available. This indicates that the input data contains multiple modalities. If both classifications are present, the process then determines whether the classifications from the text and image are the same. If both modalities agree and classify the content as non-cyberbullying (*text\_class == 0* and *image\_class == 0*), the process displays the message that input does not contain any cyber-bullying. If both modalities agree and identify the same type of cyberbullying, a message is displayed indicating that the input contains a specific class of cyberbullying, with the class identified. If the classifications disagree (i.e., *text\_class* does not equal *image\_class*), the process displays a message indicating that cyberbullying exists but that the text and image have different labels (classes). If either the text or the image classification is missing, the process concludes that the input is not multi-modal data and displays an appropriate message. The process ends once the appropriate *fusion\_message* is set based on the conditions listed above.

---

**Algorithm 1** Fusion Logic Decision Making Logic for Multi-Modal Data

---

```
1: Begin
2: if text_class is not None and image_class is not None then
3:   if text_class == image_class then
4:     if text_class == 0 then
5:       fusion_message ← "Input does not contain any Cyber-bullying."
6:     else
7:       fusion_message ← "Input contains this class {text_class} of cyberbullying."
8:     end if
9:   else
10:    fusion_message ← "Input contains cyberbullying. Text label is: {text_label} and Image
    label is: {image_label}"
11:   end if
12: else
13:   fusion_message ← "This is not Multi-Modal Data!"
14: end if
15: End
```

---

- *Output Synthesis*: It creates a cohesive response that is presented to the user, effectively communicating the findings of the cyberbullying detection analysis.

## 4.7 Models Deployment and Evaluation

In this section, we have discussed about fusion module, model's testing by performance evaluation, and model deployment. In subsection:4.6.1, we have discussed that which fusion model we have used for making decision for multi-modal data. We have discussed about testing and evaluation for the model in subsection:4.7.1. We have described the procedure of model deployment's in the subsection:4.7.2.

### 4.7.1 Models Testing

Ten percent of the data had been reserved for validation to prevent over-fitting, and rest ten percent data has been used for testing the result. By employing early stopping in our model with the patience set at three epochs, we will be able to prevent over-fitting and evaluate the model's performance during the training set by analyzing this unique set of data. We evaluated the test set's performance after the model has been trained. Next, we will compute the performance measures such as accuracy, precision, recall, ROC curve for each class, AUC for each class, and a confusion matrix to evaluate the models' performance. Subsection:2.3.10 has detailed description of each performance metric.

### 4.7.2 Models Deployment

After developing the deep learning models, these models were deployed on a graphical user-interface (GUI) using Flask Python Web Framework<sup>xx</sup>. Figure:40 shows the use-case diagram for the system process of GUI, it defines the process by which a GUI receives inputs in the form of text, images, or multi-modal data. After deployment, the GUI enables the users to choose between public and private dataset types. Then, users can enter text data, and upload an image to see which cyberbullying classification the text and image belong to.

Algorithm:2 describes the full procedure of model deployment, and how the models are responding to multi-modal data. The GUI was developed using tools such as HTML, CSS, and JavaScript. The Flask application, which was the foundation for managing incoming requests and coordinating model predictions using python program. The program carefully pre-processes the data after receiving user inputs to make sure that text and image inputs were formatted correctly for the model to use. The deep learning model's tokenizer (for RoBERTa and ViT) encrypts the input text for text data, while image preprocessing does binary thresholding and grayscale conversion to improve readability. The associated classification model for text classification and for image classification—were then fed the

---

<sup>xx</sup><https://flask.palletsprojects.com/>

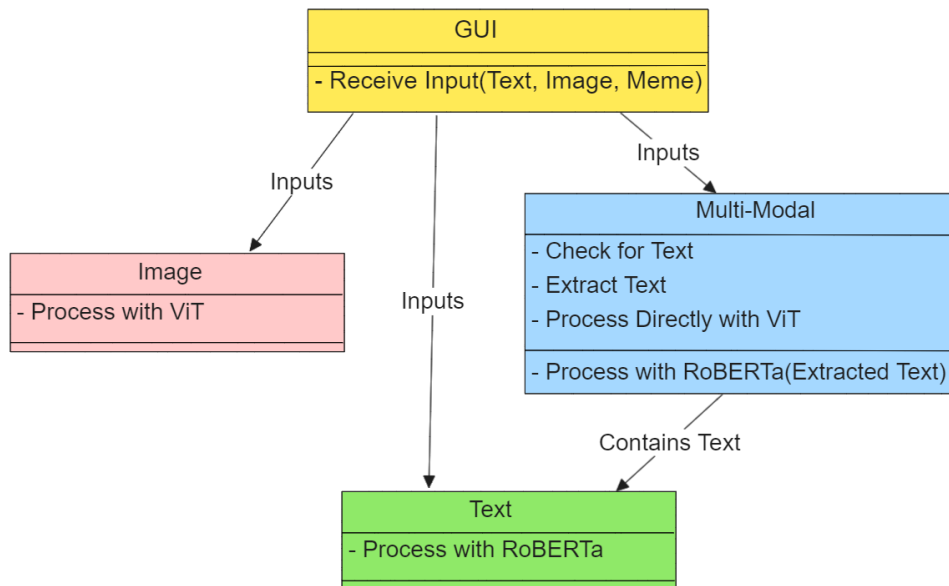


Figure 40: Use Case Diagram for Model Deployment Process

---

**Algorithm 2** Process Input through GUI

---

```

1: Start
2: Input: Receive an input through the GUI (Text or Image)
3: if the input is Text then
4:   Use the RoBERTa model to process the text.
5:   End process
6: else if the input is an Image then
7:   Check if the image contains any text.
8:   if the image contains text then
9:     Apply the Text Extracting procedure to extract text.
10:    Use the RoBERTa model to process the extracted text.
11:    Use the ViT (Vision Transformer) model to process the image.
12:    Compare text and image processing results
13:    Generate fusion message based on comparison
14:    End process
15:   else
16:     Use the ViT model to process the image directly.
17:     End process
18:   end if
19: end if
20: Render results on the GUI using a template
21: End
  
```

---

pre-processed data.

Next, RoBERTa model was used for text classification. For direct image classification, the script used a Vision Transformer (ViT) model in tandem. Then, we loaded RoBERTa and ViT models as Hybrid Model for multi-modal data to generate prediction for unseen data applying late fusion module. After

generating predictions, the program compiled the findings and showed them to the user on a results page. Users can examine any text extracted from uploaded photos as well as the expected text label and predicted image label here. Class descriptions were given to enhance the user's comprehension by providing an explanation of each predicted label. The program has strong error handling features to strengthen user experience.

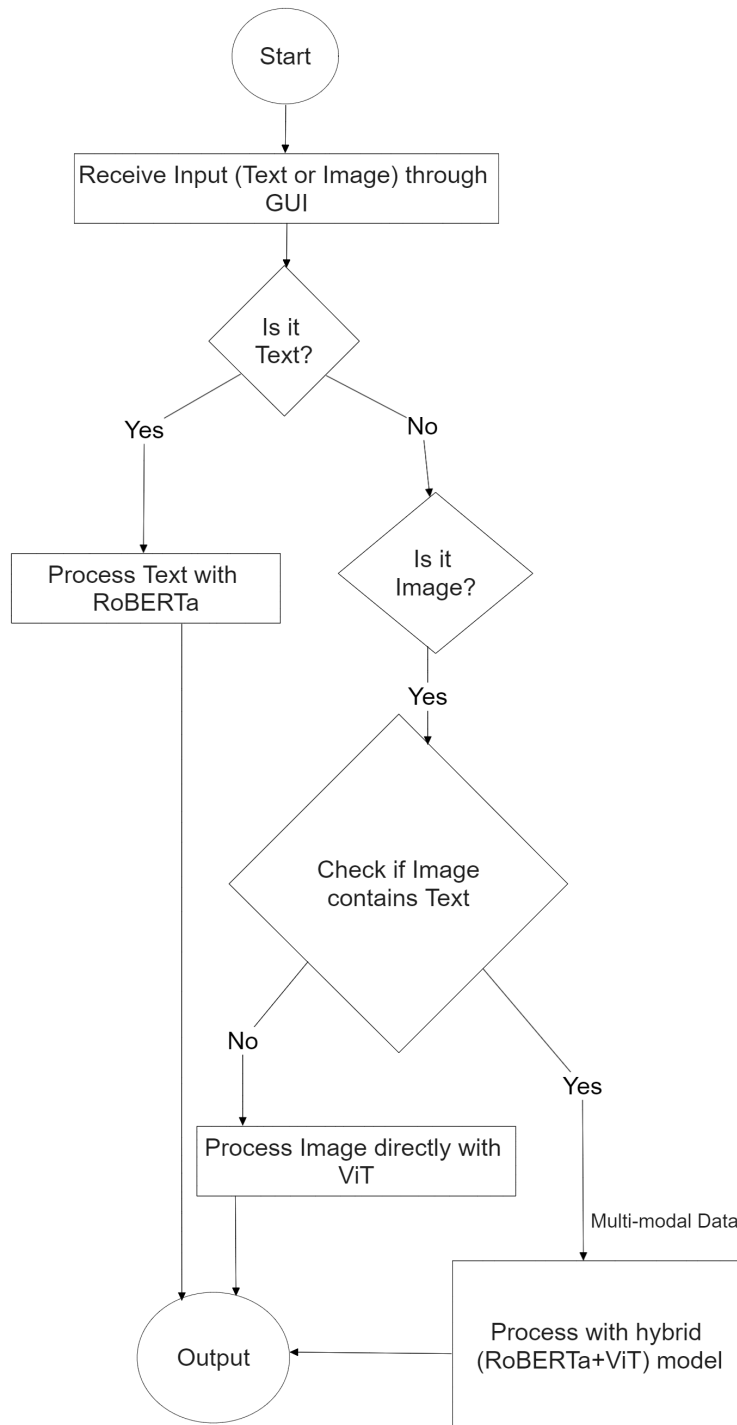


Figure 41: Flow Chart Diagram for Model Deployment Process

Figure:41 represents the flow-chart diagram to represent the model-deployment process more clearly. RoBERTa processes text inputs, and checks images for embedded text. Before ViT processes the image, RoBERTa extracts and analyzes any text that has been detected. Pictures without any text are processed directly by ViT. Every path concludes with the corresponding processing step.

## 4.8 Computing Resources

To perform the experiments, the following configurations were taken into consideration:

### 4.8.1 Hardware Configuration

The generated models' performance and efficiency are closely correlated with the underlying infrastructure. A high-performance Intel(R) Core(TM) i7-10750H processor with a base clock frequency of 2.60GHz was used to train our models. Additionally, an NVIDIA GeForce GTX 1650 Ti Discrete 4GB GDDR6 DirectX 12 GPU was installed on this system. With 4GB of RAM and support for DirectX 12, this specific GPU type offers superior graphical performance.

### 4.8.2 Software and Libraries

We have used a Windows 11 Pro operating system. Block and Arrows<sup>xxi</sup>, and Inkscape<sup>xxii</sup> software were used for drawing the figures. Google colab pro+<sup>xxiii</sup>, and Jupyter Notebook<sup>xxiv</sup> has used to run the deep-learning models. For the model deployment, Visual Studio Code<sup>xxv</sup> has been used.

The Python programming language was used for the data preprocessing, and model development. As a result, there need to install several libraries, such as: PyTorch<sup>xxvi</sup>, TensorFlow<sup>xxvii</sup>, nltk<sup>xxviii</sup>, Numpy<sup>xxix</sup>, Pandas<sup>xxx</sup>, Matplotlib<sup>xxxi</sup>, sns barplot<sup>xxxii</sup>, cv2<sup>xxxiii</sup>, Scikit-Learn<sup>xxxiv</sup>, Flask<sup>xxxv</sup> and other.

---

<sup>xxi</sup><https://www.blocksandarrows.com/>

<sup>xxii</sup><https://inkscape.org/>

<sup>xxiii</sup><https://colab.research.google.com/>

<sup>xxiv</sup><https://jupyter.org/>

<sup>xxv</sup><https://code.visualstudio.com/>

<sup>xxvi</sup><https://pytorch.org/>

<sup>xxvii</sup><https://www.tensorflow.org/>

<sup>xxviii</sup><https://www.nltk.org/>

<sup>xxix</sup><https://numpy.org/>

<sup>xxx</sup><https://pandas.pydata.org/>

<sup>xxxi</sup><https://matplotlib.org/>

<sup>xxxii</sup><https://seaborn.pydata.org/generated/seaborn.barplot.html>

<sup>xxxiii</sup><https://pypi.org/project/opencv-python/>

<sup>xxxiv</sup><https://scikit-learn.org/stable/>

<sup>xxxv</sup><https://flask.palletsprojects.com/en/3.0.x/>

## 5 Experiments and Results

This chapter begins with outlining the experimental configuration used to conduct the experiments for classifying the cyberbullying. Then the obtained results from each experiments will be presented.

### 5.1 Experimental Setup for Multi-class Classification

We used the following experimental setup and the hyper-parameter tuning process in order to conduct multiple experiments:

#### 5.1.1 For text data classification

To classify cyberbullying using text data of public dataset, models such as Hybrid (LSTM+CNN), LSTM, GRU, BERT, DistilBERT, and RoBERTa models (see subsection: 4.4.1, 4.4.2, 4.4.3, 4.4.4, 4.4.5 and 4.4.6) were used. However, for private dataset's text data, only RoBERTa model was used as mentioned in subsection:4.4.6.

#### 5.1.2 Hyperparameter tuning for text data

To optimize the model performance, we used the same hyper-parameter tuning approach for all deep-learning models to classify using both datasets' text data.

For each model, we tuned the hyperparameters using the validation dataset with batch sizes of 20, epoch size of 20, and *Adam* optimizer. During the training phase, we also employed early stopping with patience three to prevent needless runs after no improvements within three epochs. Since we needed to achieve multi-class classification, we decided to utilize *SparseCategoricalCrossentropy*<sup>xxxvi</sup> entropy as our loss function for textual data since our label's are in integer format. Since accuracy, f1-score, precision, and recall showed us about the model's efficacy, we had chosen it as our performance statistic.

#### 5.1.3 For image data classification

Using ResNet, CNN and ViT model (see subsection:4.5.1, 4.5.2, and 4.5.3), multiple experiments were conducted on image data from public datasets to categorize cyberbullying. On the another hand, we used ViT model as mentioned in subsection:4.5.3 for classifying the private dataset's image data.

---

<sup>xxxvi</sup>[https://www.tensorflow.org/api\\_docs/python/tf/keras/losses/SparseCategoricalCrossentropy](https://www.tensorflow.org/api_docs/python/tf/keras/losses/SparseCategoricalCrossentropy)

#### 5.1.4 Hyperparameter tuning for image data

To optimize the model structure, we used the same hyper-parameter tuning approach for all deep-learning models for public dataset. For the public dataset, we used the validation dataset to modify the hyperparameters for each tested model configuration. The CNN and ViT models underwent 20 epochs of training with a batch size of 20. Three epochs of early stopping were allowed for in order to prevent overfitting and pointless training after the model stops improving. The *CrossEntropyLoss*<sup>xxxvii</sup> algorithm was applied to the image classification task. This algorithm was appropriate for multi-class classification problems in which the class labels are given as integers. We used accuracy as a measure of model performance since it offers a simple way to assess how well the model classified the images.

On the another hand, for the private data, we used random search method (see subsection: 2.3.9) for both training and validation datasets in a thorough hyper-parameter tuning process as part of our ViT model. A thorough evaluation of the model's performance was conducted through a series of trials with various combinations of batch sizes and learning rates. In particular, batch sizes varied between 8, 16, 32, and 64. We ran ten different trials, each consisting of twenty epochs, in order to identify the optimal model parameters based on the observed validation loss and accuracy. Every configuration underwent training and validation stages during these trials, which allowed us to systematically evaluate the model's functionality and generalizability. Three epochs of patience were used to incorporate an early stopping mechanism into the training process.

In order to save computational resources and avoid over-fitting, this strategy was created to end training early if improvements in validation loss were not observed. We had chosen to use *nn.CrossEntropyLoss* entropy as our loss function for textual data because we need to achieve multi-class classification and our labels are in integer format. Based on its superior validation performance, the best model configuration was ultimately chosen, and its efficacy was then thoroughly assessed on the test dataset. We selected accuracy as our performance statistic because it provides insight into the effectiveness of the model.

#### 5.1.5 For multi-modal data classification

We used hybrid(RoBERTa+ViT) model for multi modal classification as described in subsection:4.6 for both public dataset and the private dataset.

---

<sup>xxxvii</sup><https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html>



### 5.1.6 Hyperparameter tuning for multi-modal data

In the subsection:38, we described that, how we extracted the text from images within the multi-modal data, and combined the extracted text data's with other text data, and memes image with other images. So, for the extracted text data, we followed subsection:5.1.2, and for the image data, we followed the subsection:5.1.4. Later, we combined RoBERTa model, and ViT model together as hybrid(RoBERTa+ViT) model.

## 5.2 Results

In this section, we have presented and have discussed the obtained results of a series of experiments conducted on both datasets.

### 5.2.1 Experimental Results on The Public Dataset

As mentioned before, to classify the cyberbullying, we used memes (multi-modal) data from public dataset. After extracting text from the memes, text data was trained by experimenting with six models such as Hybrid(CNN+LSTM), LSTM-2, GRU, BERT, DistilBERT, and RoBERTa models. Whereas, image data was trained by utilizing CNN and Vit model (see subsection:4.4, and 4.5). Later Hybrid model was used for multi-modal data, which described in subsection:38. All the obtained experimental results of these models can be seen in Table 8 for text data, and for image data in table:9. Table:10 shows the result of hybrid model for multi-modal data.

### Results of Textual Data

Table:8 shows the obtained experiment results by using several deep learning models on textual data.

Model Name	Test Accuracy	Recall	F1-Score	Precision
Hybrid (CNN+LSTM)	0.490	0.492	0.363	0.316
LSTM-2	0.477	0.48	0.39	0.39
GRU	0.506	0.49	0.37	0.32
BERT	0.977	0.977	0.977	0.977
DistilBERT	0.991	0.991	0.991	0.991
<b>RoBERTa</b>	<b>0.992</b>	<b>0.992</b>	<b>0.992</b>	<b>0.992</b>

Table 8: Performance of various Deep-Learning Models for Textual Data of Public dataset

#### Experiment - 1 for text data using hybrid (CNN+LSTM) model

In this first experiment, we trained the hybrid (CNN+LSTM) model (see subsection:4.4.1) in classifying multi-classes cyberbullying using the textual data and the obtained results can be seen in the table:8.

This model demonstrated a moderate level of performance, as indicated by a test accuracy of 0.490. The Recall score of 0.492 indicated that it had the ability to correctly identify almost half of the relevant cases. Nevertheless, the f1-score and precision of the model were relatively low, measuring at 0.363 and 0.316, respectively. This indicated that there were concern regarding the accuracy and balance of its predictions. The reason of not performing well could be, the limited use of twenty epochs, due to the absence of a supercomputer, may had led to inadequate training of the model, thus causing inaccurate classification of the data. However, The performance for each classes has shown in appendix:A.1.1.1, in table:14.

**ROC-Curve:** To test the hybrid model's performance graphically in the first experiment, we measured the ROC-AUC curve and can be seen in figure:42. From the figure, it can be seen that the model performs exceptionally well for class 0, but its effectiveness noticeably declines for the other classes, as indicated by the ROC curves for classes 1, 2, and 3, even though they are significantly lower than Class 0.

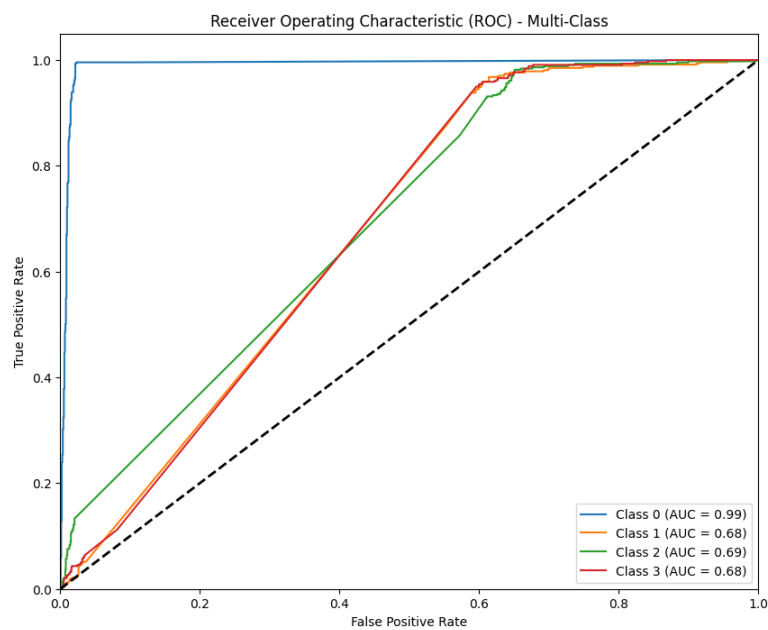


Figure 42: ROC-AUC of Hybrid model for Textual Data

**Confusion Matrix:** The confusion matrix was computed for hybrid model (CNN+LSTM) and can be seen in figure:43. The matrix demonstrates how well the model performed in classifying four distinct classes, primarily correctly identifying the intended classes. Class 0 demonstrated its ability to accurately identify this class with 455 correct predictions and very few errors. Class 1 had 455 accurate predictions, making it a flawless classification. With 428 accurate predictions for classes 3 likewise produced impressive results; however, class 1 and 2 were mistakenly classified, indicating potential feature overlap or sensitivity problems in these classes. Confusion matrix for each classes for hybrid

model (CNN+LSTM) has shown in appendix: A.1.1.1.

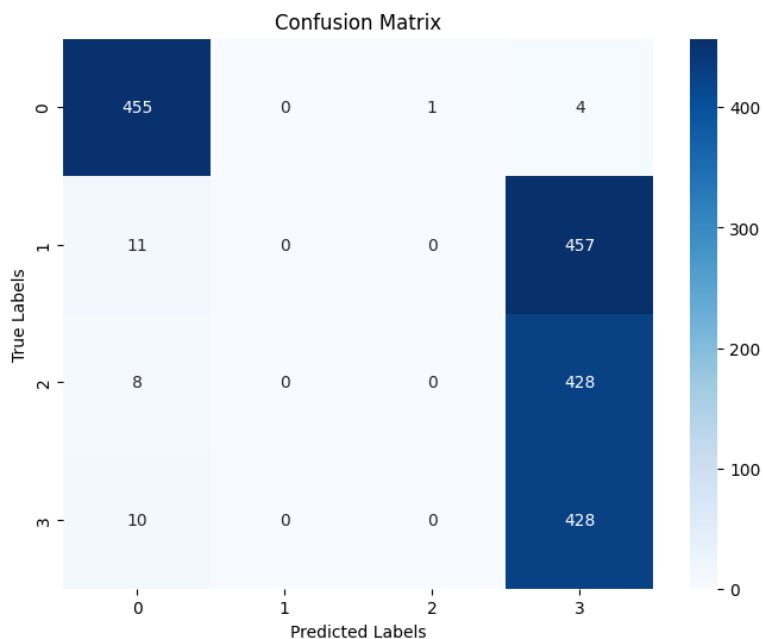


Figure 43: Confusion Matrix of Hybrid(CNN+LSTM) model on Public Data

After experimenting with confusion matrix, the performance indicating that the system’s ability to classified actual cases of cyberbullying was not good. There might occur resemblance between classes 1, 2 and 3, making it difficult to accurately categorize the data, resulting in potential misclassifications.

### Experiment - 2 for text data using LSTM-2 model

we trained the LSTM-2 model (see subsection:4.4.2) as second experiment to classify cyberbullying using textual data, and the results are shown in the table:8. The table shows that the model achieved a test accuracy of 47.7% and a f1-score of 39.0% , marginally worse than those of the CNN+LSTM model. The reason could be, because we lacked a supercomputer and only used twenty epochs, as a result the model may not have been trained enough, and leading to incorrect data classification. The performance for each classes has shown in appendix:A.1.1.2, in table:15.

**ROC-Curve:** The Figure:44, shows the overall ROC-AUC curves of LSTM model, which provide additional insight into the model’s discriminative ability across these classes. The ROC curves showed that class 0 performed well, with an AUC of 0.99, indicating that the model is extremely accurate in this class. However, performance in the other classes falls significantly, with class 1 at 0.71, class 2 at 0.51, and class 3 at 0.76, highlighting the model’s difficulty in effectively distinguishing between these types of cyberbullying.

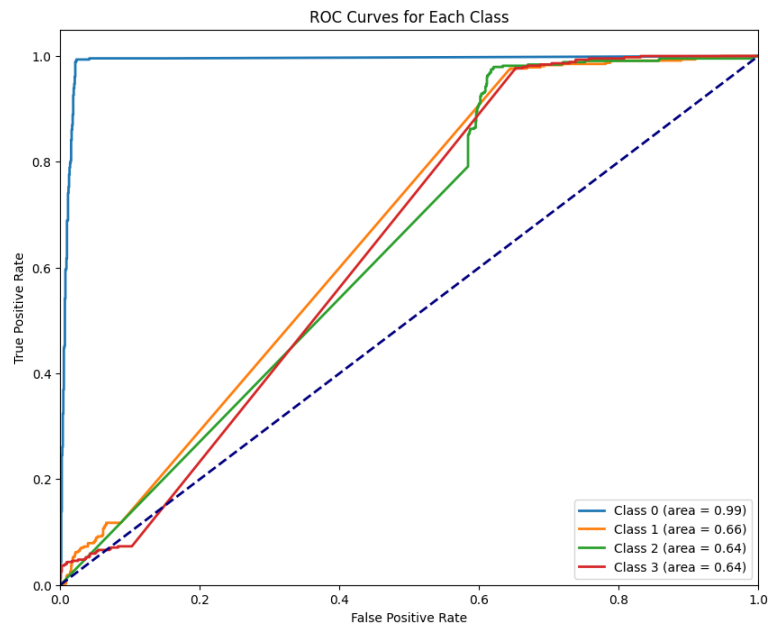


Figure 44: ROC-AUC of LSTM model for Textual Data

**Confusion Matrix:** In the figure:45, the confusion matrix for the LSTM model display a mixed performance on confusion matrix. While class 0 was handled effectively by classifying 453 data accurately by the model, class 2 also classified 351 data accurately, there were pronounced deficiencies in recognizing classes 1, and 3, necessitating further refinement and training. The confusion matrix for each class can be found in A.1.1.2.

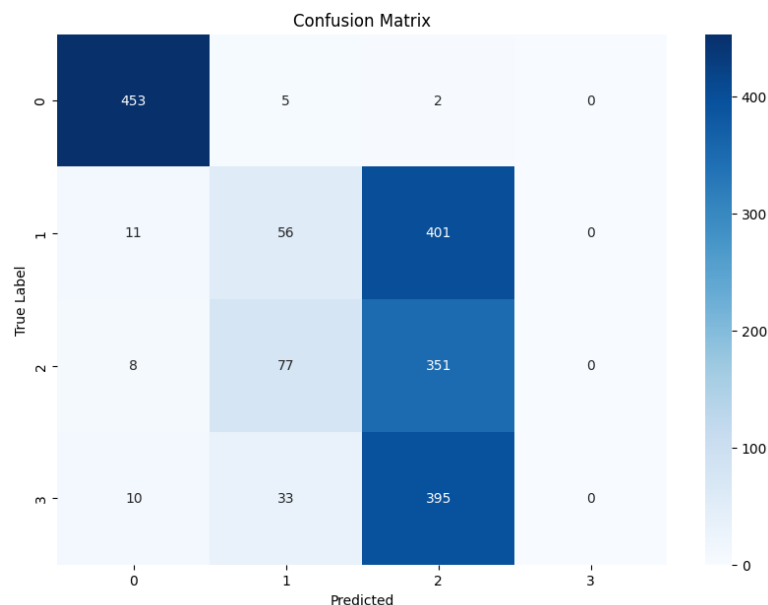


Figure 45: Confusion Matrix of LSTM model for Textual Data

After experimenting with text data using a LSTM model, the results showed that the system's ability to identify actual cases of cyberbullying was poor. Misclassifications can result from the difficulty to

correctly classify the data because classes 1, 2 and 3 can seem quite similar in some case.

### Experiment - 3 for text data using GRU model

With a test accuracy of 50.6% and 0.37 f1-score, the GRU model outperformed LSTM-2 by a small margin, suggesting small variations in model efficiencies due to architectural differences as depicted in table:8. However, the performance for each classes has shown in appendix:A.1.1.3, in table:16.

**ROC-Curve:** The ROC-AUC graph reveals information about the effectiveness of the GRU model in figure:46, which demonstrate its ability to discriminate between various classes, offered additional insights. Class 0 demonstrated an exceptional AUC of 0.99, signifying nearly flawless classification ability. In contrast, class 1, class 2, and class 3 encountered notable difficulties, as evidenced by their lower AUC values, implying that these classes were not as well classified by the model.

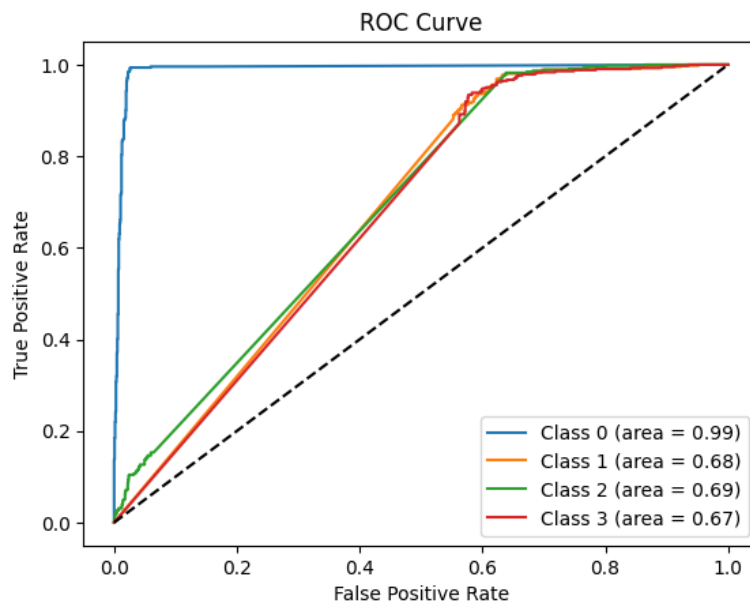


Figure 46: ROC-AUC of GRU model for Textual Data

**Confusion Matrix:** The GRU model accurately classified class 0 for 466 instances, class 1 for 445 instances, class 2 for 441 instances, and class 3 for 410 instances as shown in figure:47. The off-diagonal cells indicate misclassifications, such as 6 instances where class 0 was incorrectly predicted as class 1. Confusion matrix for each classes separately presented in appendix:A.1.1.3.

In the public dataset, classes 1, 2 and 3 can appear very similar some times, making accurate data classification difficult and potentially leading to misclassifications.

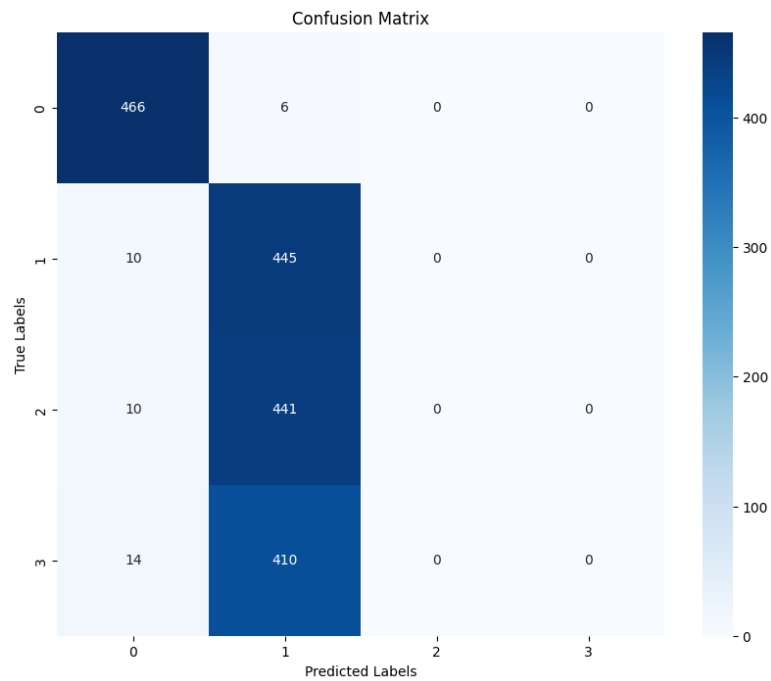


Figure 47: Confusion Matrix of GRU model for Textual Data

#### Experiment - 4 for text data using BERT model

The BERT model performed better than the hybrid(CNN+LSTM), LSTM, GRU model for the textual data with 97.7% test accuracy and f1-Score of 0.977 as shown in table:8. This indicates an excellent capability in accurately classifying relevant classes. The performance for each classes has shown in appendix:A.1.1.4, in table:17.

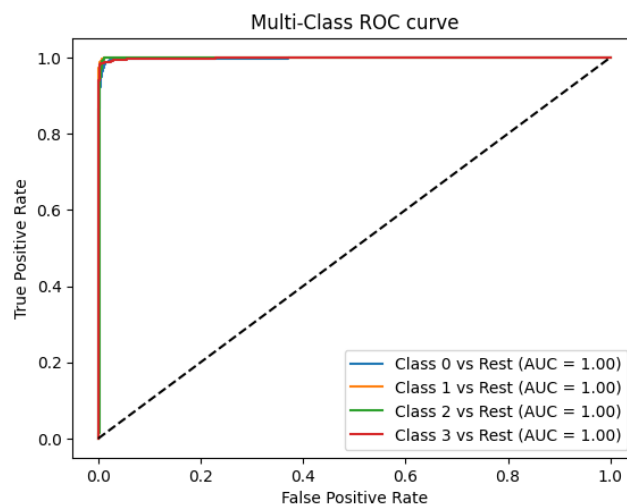


Figure 48: ROC-AUC for BERT model

**ROC-Curve:** The model's ROC-AUC graph in figure:48 shows outstanding performance across multiple classes for the BERT model, with each class outperforming the others by achieving a perfect Area

Under Curve (AUC) score of 1.00. This showed that the model can distinguish each class from the others with high accuracy and consistency.

**Confusion Matrix:** Figure:49 represents the confusion matrix for the performance evaluation of BERT model. The matrix shows that the model was very effective at classification, with significant correct predictions for each class: class 0 had 404 correct predictions, class 1 had 460, class 2 had 435, and class 3 had 424. However, performance with confusion matrices for four classes separately had represented in appendices:A.1.1.4.

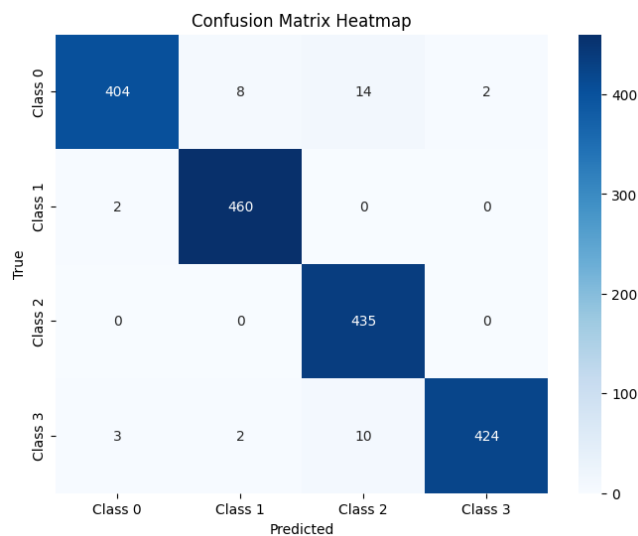


Figure 49: Confusion matrix for BERT model

However, the matrix contains some misclassifications. There are times when classes 1, 2 and 3 look a lot alike, which makes it hard to sort the data correctly and can cause mistakes.

### Experiment - 5 for text data using DistilBERT model

DistilBERT obtained 99.1% accuracy, f1-Score, recall, and precision individually(see table:8). This model provided better performance than former four experiment. which indicate that DistilBERT model perform well for multi-class classification for textual data. The performance for each classes has shown in appendix:A.1.1.5, in table:18.

**ROC-Curve:** To observe the effectiveness of DistilBERT model, we used ROC-AUC curve. The Multi-Class ROC Curve figure:50 shows that the model had high discriminatory power, with AUC values above 1.00 for all four classes.

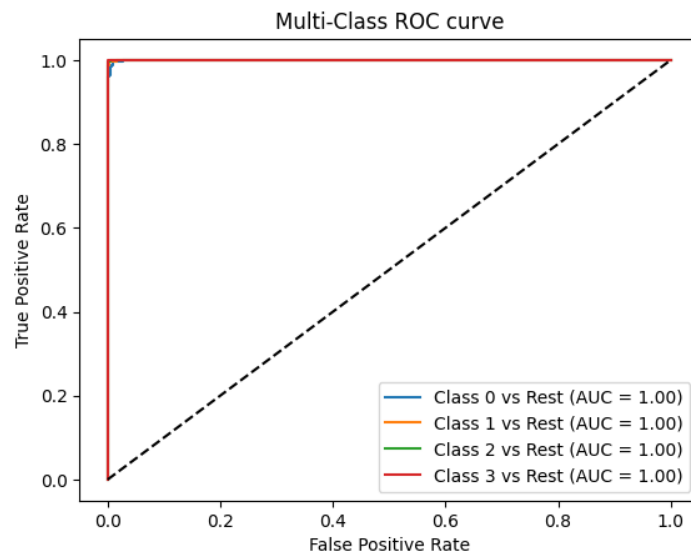


Figure 50: ROC-AUC for DistilBERT model

**Confusion Matrix:** The confusion matrices for the DistilBERT model clearly showed its high accuracy and precision in classifying textual data across multiple categories, as shown in figure:51. The model classified 424 cases for class 0 correctly, but it made a few mistakes—three times it mistook class 0 for class 2 and once for class 3. It accurately classified 453 cases for class 1, but misclassified 9 instances of class 1 as class 0. Class 2: 434 instances were correctly classified by the model, and one instance was incorrectly classified as class 0. Lastly, class 3 showed that the model was largely accurate, with 437 correct classified; however, it incorrectly classified two instances as class 0. Furthermore, performance with confusion matrices for four classes separately has represented in appendix:A.1.1.5.

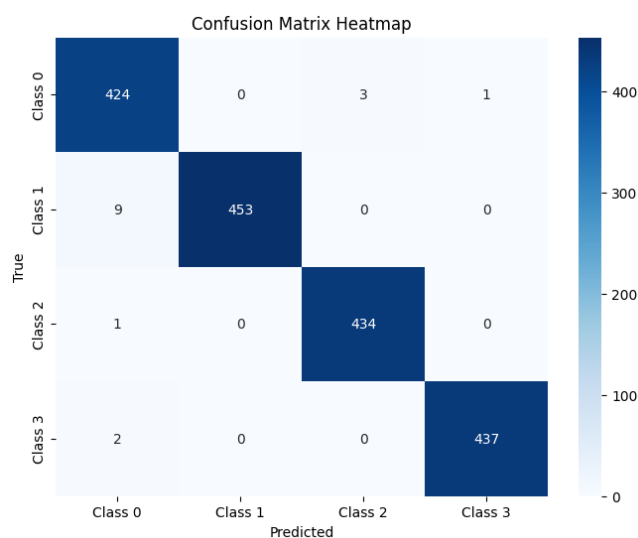


Figure 51: Confusion matrix for DistilBERT model



In the confusion matrix misclassifications occurred when classes 1, 2, and 3 appear to be very similar in some case, making it difficult to sort the data correctly and potentially resulting in errors.

### Experiment - 6 for text data using RoBERTa model

For textual data, in every statistic in table:8, the RoBERTa model performed better than the others for the textual data with 99.2% test accuracy. With a 99.2% recall, F1-score, and precision individually, it demonstrated an extraordinary capacity to categorize textual material reliably and to generalize effectively to new data. The performance for each classes has shown in appendix:A.1.1.6, in table:19.

**ROC-Curve:** The ROC curves presented in the Multi-Class ROC Curve graph in the figure:52, each class's curve rises vertically near the Y-axis before running along the top to the right, indicating optimal performance. This pattern demonstrated how well the model distinguishes each class from the others. The Area Under the Curve (AUC) for each class was 1.00, indicating that the model's predictions for each class were highly accurated and caused minimal confusion between classes. This shows that the RoBERTa model did an excellent job of classifying all of the different classes without mistaking one for another.

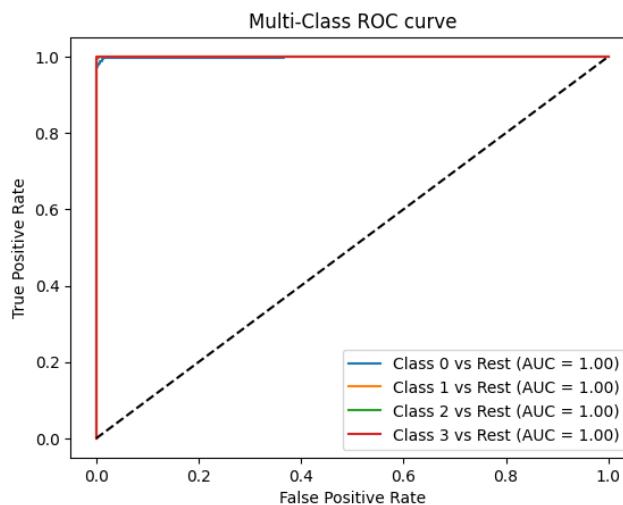


Figure 52: ROC-AUC for RoBERTa model

**Confusion Matrix:** The confusion matrices for the RoBERTa model clearly demonstrate its high accuracy and precision in classifying textual data across multiple categories in figure:53. For class 0, the model correctly predicted 418 instances, but there were some errors: one instance was incorrectly predicted as class 1, seven as class 2, and two as class 3. Class 1 had 458 correct predictions, with minor errors resulting in three instances being labeled as class 0 and one as class 2. Class 2 was perfectly predicted, with all 435 instances correctly classified, demonstrating the model's high performance in

this category. Finally, class 3 demonstrated high accuracy, with 439 instances correctly identified and no misclassifications.

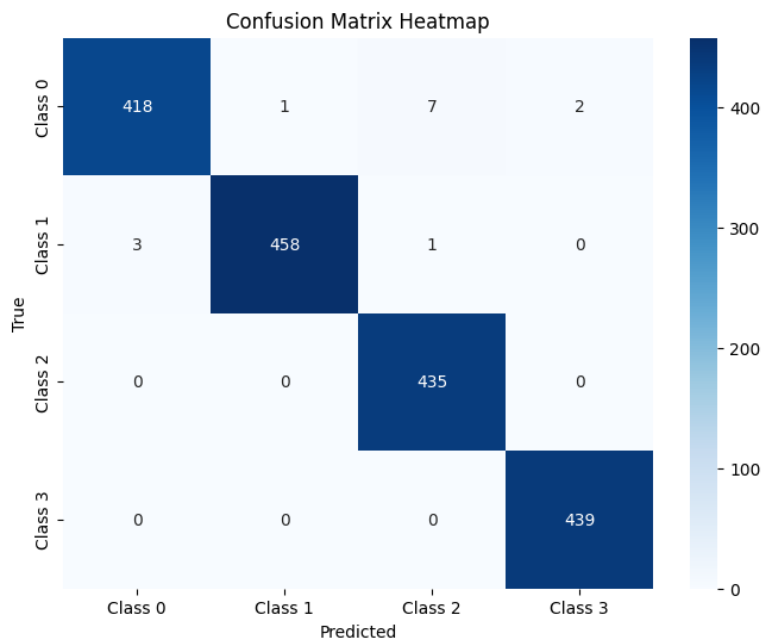


Figure 53: Confusion matrix for RoBERTa model

The confusion matrices for each class—class 0, class 1, class 2, and Class 3—consistently showed a high number of correct predictions and few misclassifications in appendix:A.1.1.6. Matrix misclassifications occurred when classes 1, 2, and 3 appear to be very similar, making it difficult to sort the data correctly and potentially leading to errors.

After evaluating textual data which was gathered by existing study, it is clearly shows that the transformer model performed well and providing better accuracy among other deep-learning models for the multi-class classification for textual data. Although BERT, DistilBERT, and RoBERTa model gave excellent performance, RoBERTa model showed the best performance among all models with 99.2% accuracy.

## Results of Image Data

Table:9 presents the obtained results by using ResNet, CNN, and ViT models on the image dataset.

Model Name	Test Accuracy	Precision	Recall	F1-Score
ResNet	0.94	0.94	0.94	0.94
CNN	0.98	0.98	0.98	0.98
<b>ViT</b>	<b>0.995</b>	<b>0.995</b>	<b>0.995</b>	<b>0.995</b>

Table 9: Performance Evaluation of Image Dataset

## Experiment - 1 for image data using ResNet model

For the first experiment of image data, from table:9, we can see ResNet model performed well with 94% test accuracy and 0.94 f1-score. This indicates that model performed well for image data classification. The performance for each classes has shown in appendix:A.1.2.1, in table:20.

**ROC-Curve:** The ROC-AUC curves shows in figure:54, high discriminative performance for all classes, with Area Under the Curve (AUC) values reflecting excellent classification capabilities: class 0 has an AUC of 0.94, class 1 had a slightly higher AUC of 0.96, class 2 exhibits near-perfect classification with an AUC of 0.99, and class 3 also performs well with an AUC of 0.94.

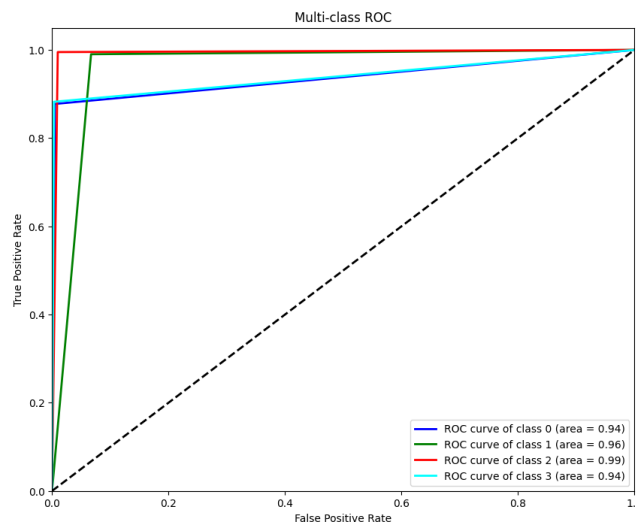


Figure 54: ROC-AUC of ResNet model on Public Data

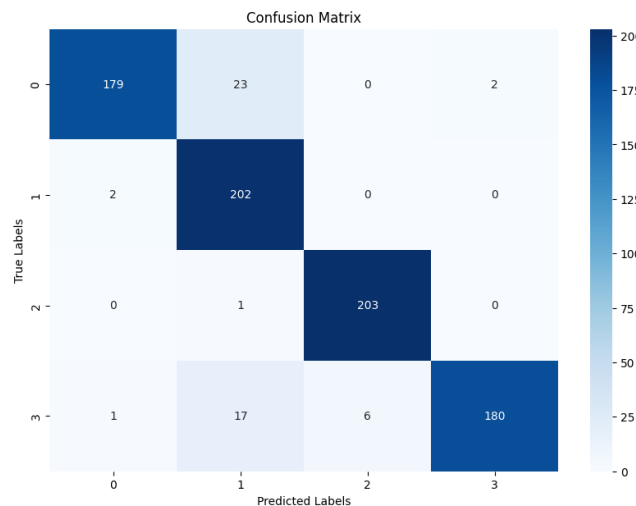


Figure 55: Confusion Matrix of ResNet model on Public Data

**Confusion Matrix:** The ResNet model's confusion matrices in figure:55 provide information about how well the model performed in each of the four classes. The model did well with class 0. It found

609 cases correctly and only made a few mistakes. Class 1 also did very well; they identified 202 objects correctly and only made two mistakes. Class 2 almost got it right, with 203 right guesses and only one wrong one. Also, class 3 did pretty well. It made 180 correct guesses, but a few more mistakes than the other classes.

In the confusion matrix, misclassifications occurred when classes 1, 2, and 3 appeared to be very similar, making it difficult to sort the data correctly and potentially leading to errors.

### Experiment - 2 for image data using CNN model

For the second experiment for image data, From table:9, we can see, CNN Model performed exceptionally well, with a 98.0% test accuracy with 0.98 of f1-score, 0.98 of precision, and 0.98 of recall. CNN model performed nicely for image data multi-class classification. The performance for each classes has represent in appendix:A.1.2.2, in table:21.

**ROC-Curve:** The CNN model's ROC-AUC graph shows a high AUC value, demonstrating its effective classifying ability in figure:56. Class 0 has an AUC of 0.99, indicating almost perfect distinction capability. Class 1 followed closely with an AUC of 0.97, while class 2 achieved a perfect AUC score of 1.00, indicating absolute accuracy in identifying this class. Class 3 has a high AUC of 0.99, almost matching class 0.

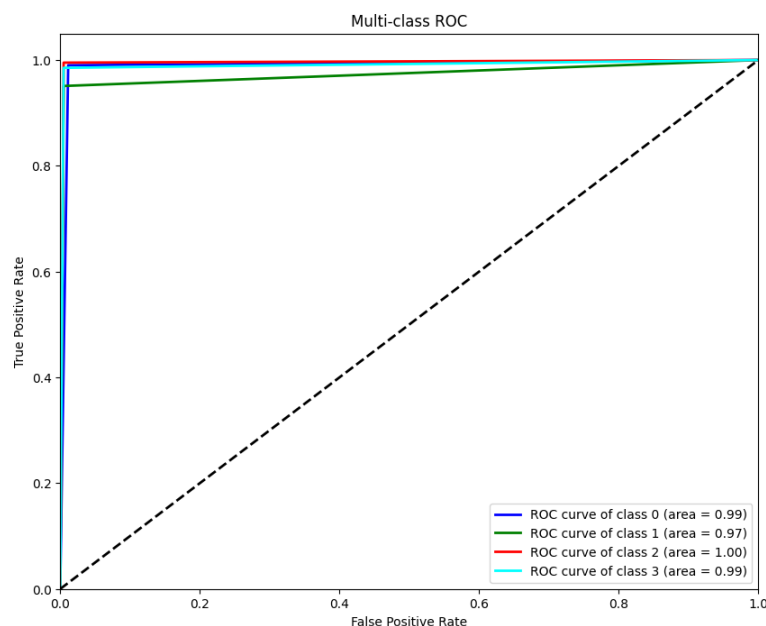


Figure 56: ROC-AUC of CNN model on Public Data

**Confusion Matrix:** The confusion matrices in figure:57 demonstrate the classification model's robust performance across four distinct classes:0, 1, 2, and 3. For class 0, the model correctly predicted

202 instances with only a few errors, misclassifying two instances as class 1 and three as class 3. Class 1 showed 194 correct predictions. Class 2 had the highest accuracy, with 203 correct predictions and one instance misclassified as Class 1. Finally, Class 3 demonstrated strong performance, with 201 correct predictions and only three instances misclassified as Class 0. Overall, the model showed strong predictive capabilities, especially in Classes 2 and 3, where it achieved high accuracy.

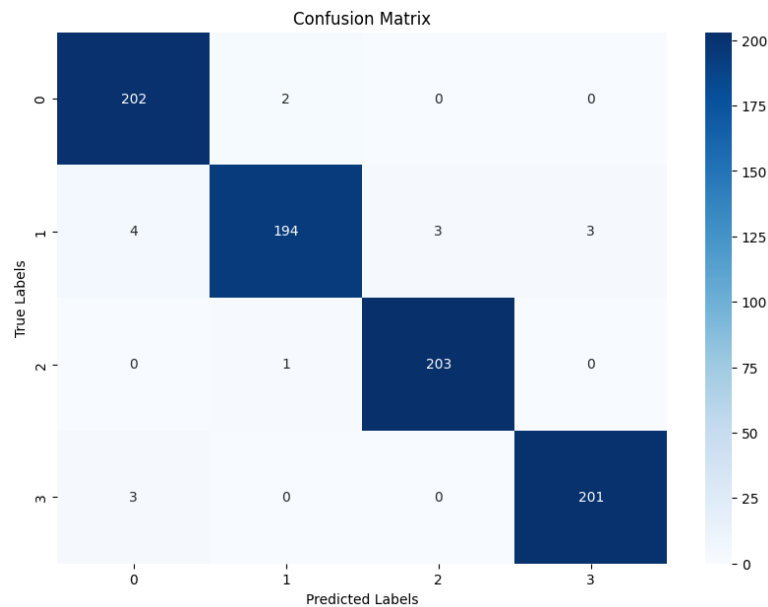


Figure 57: Confusion Matrix of CNN model on Public Data

When Classes 1, 2, and 3 seem to be quite similar, misclassifications in the confusion matrix happen, which makes it challenging to sort the data accurately and maybe leads to mistakes.

### Experiment - 3 for image data using ViT model

In the third experiment for image data, the performance profile of the ViT Model was even more remarkable in table:9. With a test accuracy, precision, recall, and F1-score of 99.5% of all mirrored this high accuracy than ResNet and CNN model, and demonstrating the model's remarkable capacity to recognize real instances of cyberbullying without misclassified them. The performance for each classes has shown in appendix:A.1.2.3, in table:22.

**ROC-Curve:** The ROC-AUC graph in figure:58, for the ViT model is quite near to 1 for four classes 0, 1, 2, 3. The graph highlighting the ability to discriminate between the various classes. Which indicated, ViT model performed well for the image data.

**Confusion Matrix:** The provided classification model's confusion matrices in figure:59, provide information about its performance across four distinct classes. For class 0, the model correctly iden-

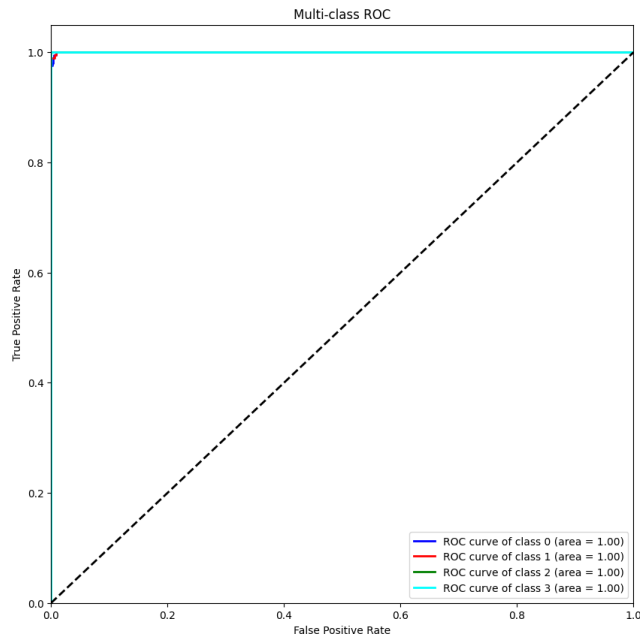


Figure 58: ROC-AUC for ViT model on Public Data

tified 202 instances with a low number of misclassifications, demonstrating high precision and accuracy . Similarly, class 1 performed well, with 202 correct classifications and few errors, indicating that the model is reliable. Class 2 and class 3 both show perfect identification, with 204 correct predictions each and no instances misclassified as other classes, demonstrating the model's exceptional ability to distinguish between these categories accurately. Confusion matrix for each four classes has represented in appendix:A.1.2.3.

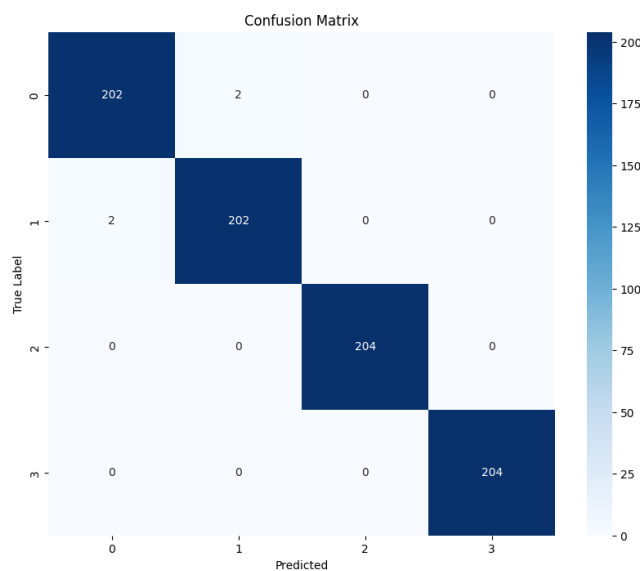


Figure 59: Confusion Matrix of ViT model on Public Data

Misclassifications in the confusion matrix happened when there seemed to be a lot of similarities

between Classes 1, 2, and 3, which made it challenging to sort the data accurately and might have led to mistakes.

In conclusion, the ViT model marginally outperformed the ResNet, and CNN model overall with 99.5% accuracy. This is especially true given that the ViT model consistently scored highly in every class. This highlights the ViT model’s exceptional capacity to accurately and consistently handle a wide range of complex cyberbullying cases in image data.

## Result of Multi-Modal Data

From the table:8, and table:9, we can see that, for the textual data, RoBERTa model performed best accuracy among other deep learning models for textual data with 99.2% accuracy and ViT model performed best accuracy than other deep learning models for image data with 99.5% accuracy. So we chosed this two models for experimenting with text and image data, and hybrid(RoBERTA+ViT) model used to classify the multi-modal data. We merged the accuracy of RoBERTa and ViT model and calculate their average performance for getting hybrid(RoBERTA+ViT) model’s accuracy to classify multi-modal data into multi-classes. Table:10 depicted the performance outcome of hybrid(RoBERTA+ViT) model, where accuracy and f1-score is 99.2% and ROC-AUC value is 0.999.

Model Name	Accuracy	Recall	F1-Score	Precision	ROC-AUC
Hybrid(RoBERTa+ViT)	0.992	0.992	0.992	0.992	0.999

Table 10: Performace Evaluation for Multi-modal data’s Model for Public Data

To classify the multi-modal data, we used a late fusion module (refer to subsection 4.6.1). Figure:60 displays the result of multi-modal data by showing how it was processed. This depicts a system that makes predictions or decisions based on text and images. It processed text and images separately, using two different models, RoBERTa and ViT. RoBERTa, which handles the text, and the ViT model examined the images. Before making the final decision, the fusion module combines data from both text and images. The model has an accuracy of 99.2%.

**Confusion Matrix:** Figure:61 depicted the confusion matrix hybrid(RoBERTA+ViT) model which shows the performance for each classes. class 0 had the most correct predictions 308. Class 1 showed high accuracy, with 332 correct predictions and few errors—only two instances were misclassified as class 0. Classes 2 and 3 both demonstrate flawless predictive accuracy, with 319 and 321 correct predictions, respectively, and no instances incorrectly classified into any other class. This matrix highlights the model’s robust ability to accurately identify classes 2 and 3, as well as its performance with class

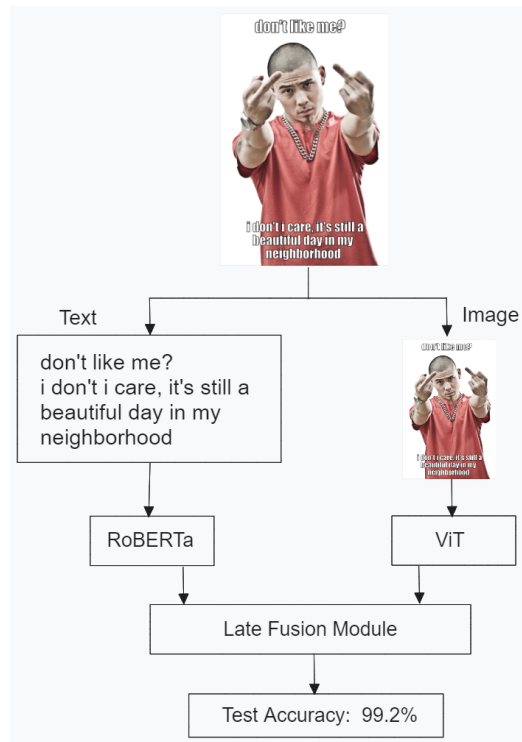


Figure 60: Result of Multi-modal Data for Public Dataset

1.

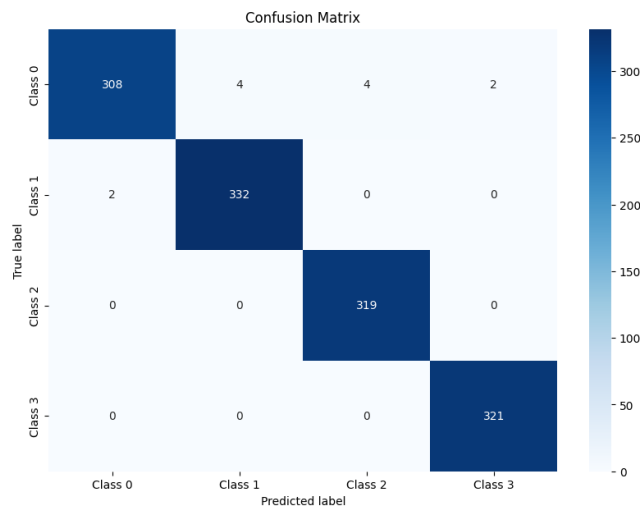


Figure 61: Confusion Matrix of Hybrid(RoBERTa+ViT) model

## 5.2.2 Experimental Results on Private Dataset

Table:11 shows the performance result for the RoBERTa, ViT, and hybrid(RoBERTa+ViT) models on a dataset that includes text, image, and multimodal (memes) data. The performance of three models, RoBERTa, ViT, and hybrid(RoBERTa+ViT), was critically examined in the Self-Collected Dataset, i.e private dataset.



Model Name	Accuracy	Recall	F1-Score	Precision
RoBERTa for Text Data	0.982	0.982	0.982	0.982
ViT for Image Data	0.932	0.932	0.932	0.933
Hybrid(RoBERTa+ViT) for Multi-modal Data	0.961	0.9599	0.9599	0.960

Table 11: Performance Evaluation for Private Data

## Result of Textual Data

The RoBERTa model showed remarkable performance with 98.2% accuracy, recall, F1-score, and precision score individually in table:11. With scores of 0.986 for accuracy, recall, f1-score, and precision, the model demonstrated exceptional performance in all classes. The performance for each classes has depicted in appendix:A.1.1, in table:23.

**ROC-Curve:** The ROC curve, in the figure:62 demonstrates a classification model's exceptional performance across multiple categories. Interestingly, each class (0, 1, 2, and 3) had an Area Under the Curve (AUC) of 1.00, indicating perfect classification accuracy.

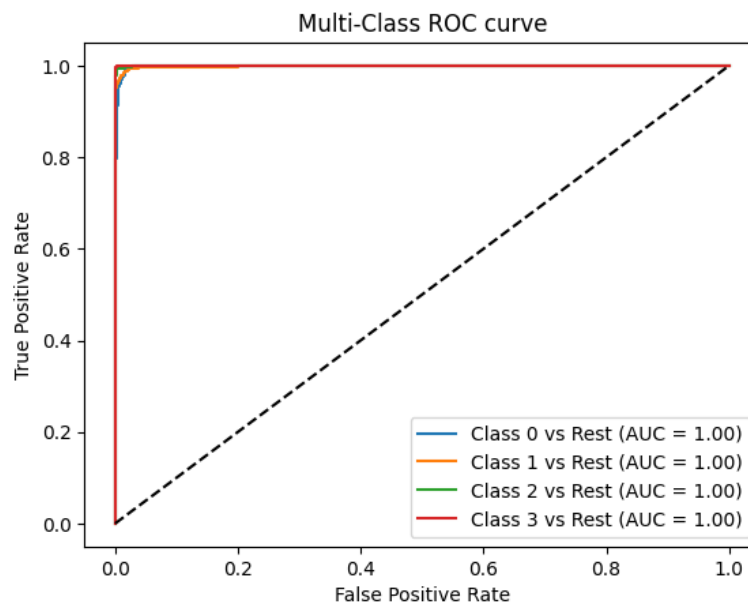


Figure 62: ROC-AUC of RoBERTa model on Private Data

**Confusion Matrix:** Figure: 63 shows the confusion matrix of RoBERTa model. The matrix for class 0 classified 772 true positives. Class 1 contains 756 true positives and 15 classes was failed to classified correctly in total. Class 2 shows 794 true positives missclassified with eight classes. Class 3 Like Class 2, it has an excellent prediction rate, with 795 true positives, 2 false negatives, and 2 false positives. The performance matrix for individual four classes has shown in appendix:A.2.1.

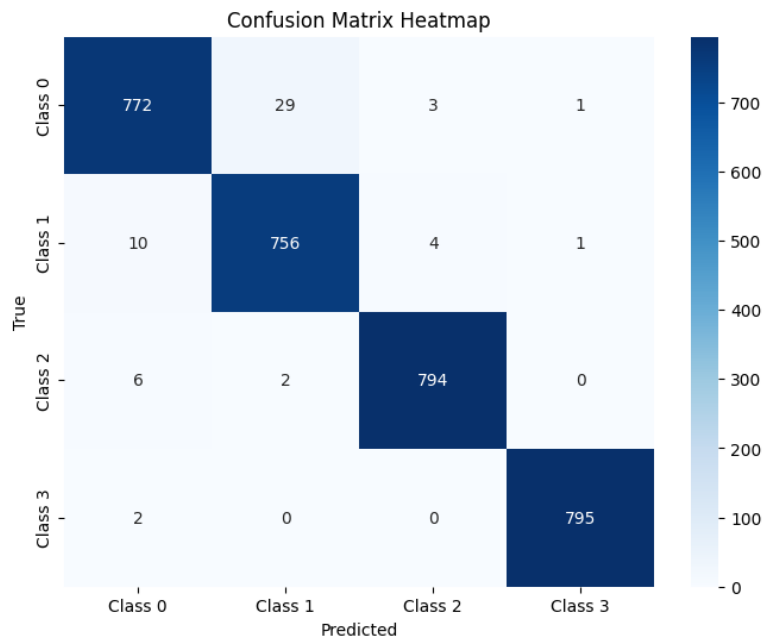


Figure 63: Confusion Matrix of RoBERTa model on Private Data

When there appeared to be a lot of similarities between Classes 1, 2, and 3, it was difficult to classify the data accurately and may have resulted in errors. This led to misclassifications in the confusion matrix.

### Result of Image Data

For the private image data experiment, the ViT model, tailored for image data, with an accuracy, precision, recall, and F1-score of 0.932, and a precision of 0.933 as depicted in table:11. Which indicated an excellent capability in accurately classifying the image data. The performance for each classes has shown in appendix:A.2.2, in table:24.

**ROC-Curve:** Class 0, Class 1, Class 2, and Class 3 ROC curves displayed in blue, red, green, and cyan, correspondingly in figure:64. With an AUC of 0.99, Class 0 performed excellently, albeit just short of perfection. With an AUC of 0.98, Class 1 came in close second, indicating excellent performance but a marginally higher false positive rate than Class 0. Both Class 2 and Class 3 exhibit exceptional classifier accuracy with no false positives, achieving perfect AUC scores of 1.00.

**Confusion Matrix:** The presented figure:65 comprise a set of confusion matrices that show how well a classifier performs on a dataset consisting of four classes. Class 0 had 25 correct predictions, Class 1 had 22, Class 2 had 24, and Class 3 had 25, demonstrating the classifier’s strong performance across all classes with high true positive rates, according to the overall confusion matrix. The performance matrix for each classes separately has shown in appendix:A.2.2.

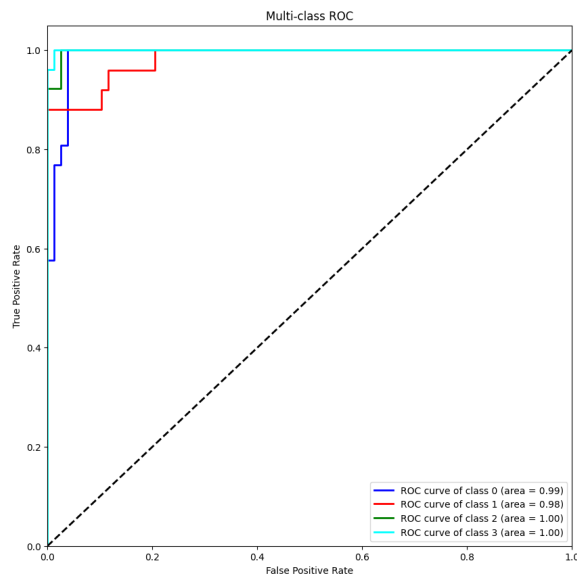


Figure 64: ROC-AUC for ViT model on Private Data

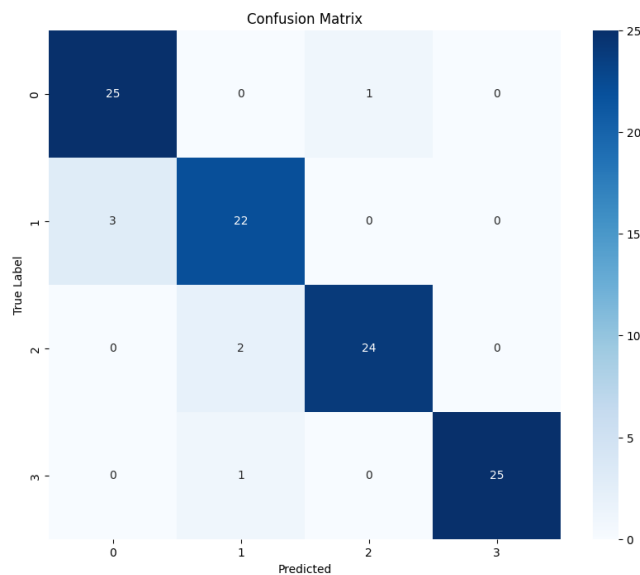


Figure 65: Confusion Matrix of ViT model for Private Data

When Classes 1, 2, and 3 appeared together into single data, it was difficult to accurately classify the data and may have resulted in errors. This led to misclassifications in the confusion matrix.

### Result of Multi-Modal Data

To experiment the multi-modal data, we combined two models RoBERTa and ViT as a hybrid (RoBERTa+ViT) model to classify multi-modal data. We combined the accuracy of the RoBERTa and ViT models and calculated the average performance of two models to obtain the hybrid (RoBERTa+ViT) model's accu-

racy using the late fusion module (see subsection:4.6.1) in classifying multi-modal data into multiple classes. Table:11 shows the performance outcome of the hybrid (RoBERTA+ViT) model, with accuracy of 96.1%, and f1-score of 95.99% and ROC-AUC value of 0.99.

**Confusion Matrix:** Figure: 66 depicts the confusion matrix hybrid (RoBERTA+ViT) model, which shows the performance for each classes.

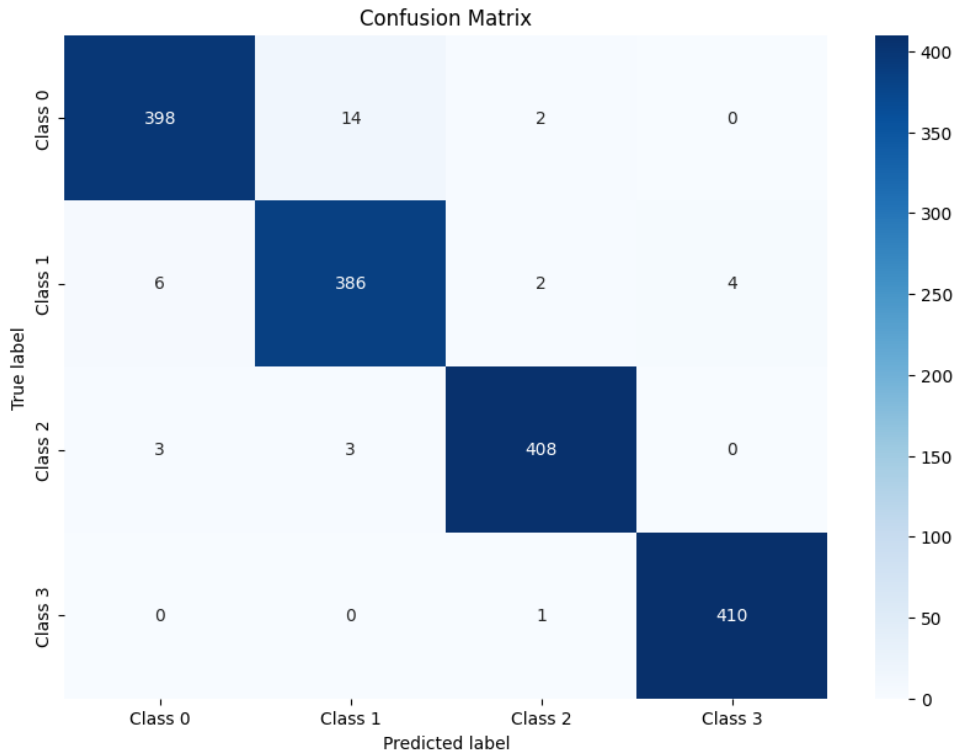


Figure 66: Confusion Matrix of Hybrid(RoBERTA+ViT) Model for Private Data

The model correctly classified 398 instance for Class 0. Similarly, in Class 1, it mostly got it right with 386 correct predictions, but it made a few mistakes, mislabeling 6 items as Class 0, 2 as Class 2, and 4 as Class 3. Class 2 and Class 3 had very high correct predictions (408 and 410, respectively), with few items mislabeled, this is because, when Classes 1, 2, and 3 in the confusion matrix seem to be very similar, misclassifications happen, making it challenging to classify the data accurately and possibly leading to mistakes.

Figure:67 which displayed the result of multi-modal data by showing the procedure of how multi-modal data was processed. This depicts a system that uses text and images to make predictions or decisions. It processed text and images separately, using two distinct models. RoBERTa, which handles the text, is built on technology that reads and understands words. The Vision Transformer (ViT) model examined the images. After the fusion module combines information from both text and image just before making the final decision. The model shows the accuracy of 96.1%.

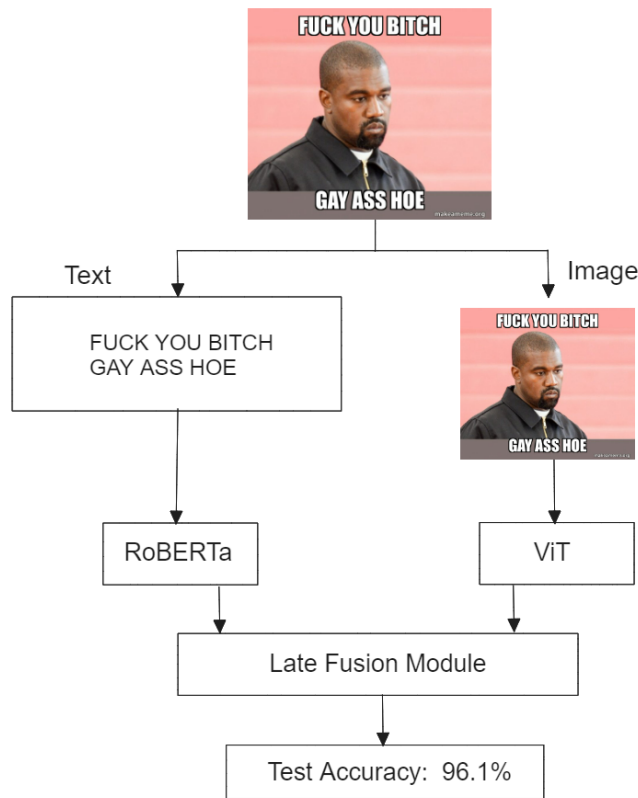


Figure 67: Result of Multi-modal Data for Private Dataset

### 5.3 Model Deployment

We have deployment the model for showing our result in graphical user interface (GUI). The processing of generating the graphical user interface described in the subsection:4.7.2. In this section, we showed experiment result of model deployment. The result of inputs such as class label 0, 1, 2, 3's explanation was described in subsection:4.2.

**Cyberbullying Multi-class Classification for Multi-modal Data**

**Dataset:**

**Text:**

**Image:**  
 No file chosen

Figure 68: Cyberbullying Classification Web-Page

Figure:68 displays the webpage's user interface. The system can take text data, image data, and multi-modal data as input, and can display result of the input into multi-class cyberbullying. Image input can take both image and image that contains text such as memes (multi-modal) data, and can generate prediction result according to image and multi-modal data. It is evident from the figure:69 that the model is capable of predicting outcomes for both public and private datasets.

The screenshot shows a web interface titled "Cyberbullying Multi-class Classification for Multi-modal Data". It features a "Dataset:" dropdown menu with "Private Dataset" selected. Below the dropdown are two buttons: "Submit Text" and "Upload Image". The "Image:" section shows a "Choose File" button and the text "No file chosen".

Figure 69: Cyberbullying Classification Using The Model of Private and Public Data

The screenshot displays three sections of class descriptions:

- Class-Description for Text Data (Public and Private Dataset)**
  - Class 0:** The data does not include any content that is insulting, defamatory, offensive, or contains threatening or aggressive language
  - Class 1:** Defaming cyber-bullying refers to behaviors in which individuals insult or defame another. It encompasses the act of expressing offensive comments, disseminating untrue or harmful information about an individual, or participating in the defamation of someone's character.
  - Class 2:** Offensive language cyber-bullying include situations in which individuals employ derogatory language such as "F\*cker," "bitch," and "dog" to target someone
  - Class 3:** Threatening, hateful, aggressive comments
- Class-Description for Image (Public Dataset)**
  - Class 0:** Non-Bullying (class-0): Normal image, which does not contains any defaming, sexual, offensive, aggressive content
  - Class 1:** Contains sexual, nudity content
  - Class 2:** Showing middle finger
  - Class 3:** Beating someone, showing weapon to someone
- Class-Description for Image (Private Dataset)**
  - Class 0:** Normal image, which does not contains any defaming, sexual, offensive, aggressive conten
  - Class 1:** Contains sexual, nudity content
  - Class 2:** Showing middle finger, mixing other creature's face into people's face
  - Class 3:** Beating someone, showing weapon to someone

At the bottom of the page is a "Back to Home" button.

Figure 70: Showing The Result Description for text, image, and multi-modal data in the GUI

Figure:70 shows the class description note, which will help user to understand the prediction result

for text, image and multi-modal memes data.

Text data was tested as showed in figure:71, and figure:72. The text in figure:71 was cyber-bullying which was containing to insult to someone, as a result, the output shows that the class label was 1 (see section:4.2.1) in figure:72 . The result had been tested for both public and private dataset.

**Dataset:**

**Text:**

Figure 71: Testing Text For Class label: 1

### Prediction Results

**Extracted Text:**

No text extracted

**Text Prediction:**

Text class label: 1

**Image Prediction:**

No image prediction

**Multi-Modal Data Prediction:**

This is not Multi-Modal Data!

Figure 72: Showing The Result after Uploading Input as Text of Figure:71

As seen in figures 73 and 74, text data was tested. Because the text in figure:73 contains defamatory language intended to harm someone, the output indicated that the class label in figure:74 is 3 (see section:4.2.1). Both public and private datasets was undergone testing to verify the outcome.

**Dataset:**

**Text:**

Figure 73: Testing Text For Class label: 3

### Prediction Results

**Extracted Text:**

No text extracted

**Text Prediction:**

Text class label: 3

**Image Prediction:**

No image prediction

**Multi-Modal Data Prediction:**

This is not Multi-Modal Data!

Figure 74: Showing The Result after uploading input Image of figure:73



Figure 75: Test Image For Private Data

## Prediction Results

### Extracted Text:

### Text Prediction:

No text prediction

### Image Prediction:

Image class label: 2

### Multi-Modal Data Prediction:

This is not Multi-Modal Data!

Figure 76: Showing The Result after uploading input Image of figure:75

In the figure:75, only image data was tested. The image in figure:75 contained offensive image of someone, where it was trying to use animal's face into someone's face for bullying that person, as a result the output in figure:76 indicated that the class label of figure:75 is 2 (see section:1). Private datasets undergone testing to verify the outcome. Since the image did not contain any text, so the system identified that the image was not multi-modal data, as a result fusion module displayed as a output that the input was not multi-modal data, and the system only displayed the result for image data.

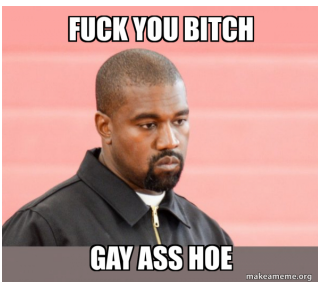


Figure 77: Testing The Multi-Modal Data For The Model

## Prediction Results

### Extracted Text:

fuck you bitch w j gay ass hoe

### Text Prediction:

Extracted text class label: 3

### Image Prediction:

Image class label: 0

### Multi-Modal Data Prediction:

Input contains cyberbullying. Extracted text class label: 3 and Image class label: 0

Figure 78: Showing The Result after uploading input Image of figure:77

The multi-modal data was tested using figure:77 to determine whether or not the text and image constituted cyberbullying. If it did, its label would be displayed. Thus, the outcome of figure:77 was displayed in figure:78. Since the image did not depict any cyberbullying material, the image label is 0 (see section:1). The extracted text was aggressive based text, hence the text label is 3 (see sec-



tion:4.2.1). Both public and private datasets undergone testing to verify the outcome. Since data was multi-modal, the fusion module showed the result for extracted text and images, and generated decision for multi-modal data.



Figure 79: Test Image For The Model

### Prediction Results

#### Extracted Text:

you are a stupid bitch

#### Text Prediction:

Extracted text class label: 2

#### Image Prediction:

Image class label: 2

#### Multi-Modal Data Prediction:

Input contains cyberbullying of class label: 2.

Figure 80: Showing The Result after uploading input Image of figure:79

Figure:79 was used to test the multi-modal data in order to ascertain whether the text and image qualify as cyberbullying. Its label will appear if it does. Consequently, figure:80 displayed the result of figure:79. The image label was 2 because there was showing middle finger which was evidence of cyberbullying in it (see section:1). Since the extracted text was offensive word in nature, class label: 2 (see section:4.2.1) was the text label. Testing had been done on both public and private datasets to confirm the results. The fusion module produced a decision for the multi-modal data and displayed the results for the extracted text and images because the data was multi-modal.



Figure 81: Testing the Multi-modal data For The Model

### Prediction Results

#### Extracted Text:

so cute

#### Text Prediction:

Extracted text class label: 0

#### Image Prediction:

Image class label: 0

#### Multi-Modal Data Prediction:

Input does not contain any Cyber-bullying. Extracted text class label: 0 and Image class label: 0

Figure 82: Showing The Result after uploading input multi-modal data of figure:81

The multi-modal data was tested using figure:81 to determine whether the text and image qualify as

cyberbullying. If so, its label will show up. As a result, the outcome of figure:81 is shown in figure:82. The figure is displayed, a cartoon, which was not containing any cyberbullying, hence the image label was 0 (see section:1). As the extracted text did not contain any bullying language, the text label was class label: 0 (see section:4.2.1). To verify the findings, testing had been done on both public and private datasets. Because the data was multi-modal, the fusion module generated a decision for it and showed the outcomes for the extracted text and images.



Figure 83: Test Meme For The Model

## Prediction Results

### Extracted Text:

you should die, understand???

### Text Prediction:

Extracted text class label: 3

### Image Prediction:

Image class label: 3

### Multi-Modal Data Prediction:

Input contains cyberbullying of class label: 3.

Figure 84: Showing The Result after uploading input Image of figure:83

The multi-modal data was tested using Figure:83 to determine whether the text and image data as cyberbullying. As a result, the outcome of figure:83 was shown in figure:84. The figure was displayed, a man was trying to kill a woman , which was containing aggressive based cyberbullying (see section:1), hence the image label was 3. As the extracted text contains aggressive based bullying language (see subsection:4.2.1), the text label was class label: 3. Testing has been done on both public and private datasets in order to confirm the results. The fusion module produced a decision for the multi-modal data and displayed the results for the extracted text and images.

This is how the GUI performed to classify multi-class cyberbullying for muti-modal data.

## 6 Discussion

This section will discuss our results against research questions outlined in subsection:1.3, our result has been verified with private data and discussed in subsection:6.2, finally we compared our result with existing literature in subsection:6.3.

### 6.1 Discussing Results for each Research Question

To achieve the main objective of this thesis, the following research questions have been outlined in this subsection:

1. How to collect, label and pre-process a multi-modal cyberbullying dataset from various social media platforms?

This research question was addressed by collecting, labelling, and pre-processing two multi-modal cyberbullying datasets obtained from various social media platforms. Public dataset used in thesis was collected by Hamza *et al.* [38], and Maity *et al.* [37]. The dataset provided by Hamza *et al.* [38] was labelled for binary classification. On the other hand, the dataset provided by Maity *et al.* [37] was labelled as multi-label for textual data and binary for image data. Private data collection entails extracting text and image content from platforms such as Tiktok, Instagram, and Facebook, TikTok via APIFY and other web scraping techniques (see subsection:4.1). Next, the collected data was categorized to classify instances of cyberbullying as mentioned in subsection:4.2.1, 1. After that, text data was pre-processed by cleaning, text data preprocessing, augmenting, and sampling which is described in subsection:4.2.1, and image data was pre-processed by re-sizing, coloring, augmenting and sampling as mentioned in subsection:4.2.2. Text has been extracted from images and preprocessed for multi-modal data as described in subsection:4.2.3. The detailed answer to this question has been presented in Section 4.1.

2. Which deep learning models are best suitable for multi-class classification of cyberbullying using text data?

We answered this research question by applying six deep learning models on public dataset's text data such as hybrid(CNN+LSTM) model, GRU model, LSTM model, BERT model, DistilBERT model, and RoBERTa model for multi-class classification. From these applied models, the experimental results showed that Roberta model obtained an accuracy, recall, f1-score and precision of 99.2% respectively (see Table:8) when compared with other models. For private dataset's text data, we used RoBERTa model and this model achieved 98.2% accuracy, recall, f1-score and

precision respectively. The detailed answer to this question has been presented in Sub-section 5.2.1.

3. Which deep learning models are best suitable for multi-class classification of cyberbullying using image data?

We have developed three deep learning models, such as: ResNet model, ViT model, and CNN model for image data classification. After applying these three models, we have got best suitable result with ViT model with a 99.5% accuracy, recall, f1-score and precision respectively which has shown in the table:9. So, it can be said that, ViT model is the best suitable among other deep learning models for multi-class classification of cyberbullying using image data. As a result we applied ViT model in private data also, and got 93.2% accuracy, recall, f1-score and precision respectively. The detailed answer to this question has been presented in the table:11.

4. Which deep learning models are best suitable for multi-class classification of cyberbullying using multi-modal data?

After getting suitable model for text data which was RoBERTa (see table:8) and for image data which was ViT (see table:9), we decided to use these two models together as hybrid model for our multi-modal data, and we have applied hybrid (RoBERTa+ViT) model using late fusion module for our multimodal data for multi-class classification for both public and private dataset. We got 99.2% accuracy, recall, f1-score and precision for public data, and 96.1% accuracy, and 0.96 recall, f1-score and precision score for private data using hybrid (RoBERTa+ViT) model for multi-modal data. The detailed answer to this question has been presented in Sub-section:5.2.1, and 5.2.2.

5. How should the results from the built deep learning models be presented on the developed GUI?

We used a graphical user interface to deploy the model, which allowed users to input text, images, or multimodal data and view the classification results. Initially, the system determines the nature of the input. If it is text, it is processed by RoBERTa, a language model designed to handle a wide variety of text-based tasks. When an image is input, the system looks for embedded text first. If text is detected within an image, RoBERTa extracts and processes it separately, while the image content is processed by Vision Transformer (ViT), an architecture that uses transformer models to analyze images. For images without text, the ViT performs the processing directly. In scenarios where both text and image processing are required, a hybrid model approach is used in late fusion module, combining the capabilities of RoBERTa and ViT to interpret multimodal data which is described in subsection4.7.2, and defined the experiment's result in subsection:5.3.

## 6. Does deep-learning perform better than the state-of-the-art algorithms?

As we mentioned earlier in subsection:4.1.1 that, our public dataset was based on the combined datasets from Maity *et al.* [37] and Hamza *et al.* [38]. So, our Results of Public Dataset appears to be based on the combined datasets. We used RoBERTa and ViT. We have significantly higher accuracy, f1-score, recall, precision, with RoBERTa scoring 99.2%, 0.992, 0.992, and 0.992 respectively and ViT 99.5%, 0.995, 0.995, and 0.995 respectively as showed in table:12, and for the multi-modal data, we got 99.2% accuracy and f1-score using Hybrid(RoBERTa+ViT) model. We have achieved higher result, and better performance than the study of Maity *et al.* [37] and Hamza *et al.* [38], where they ended up with 63.36% and 70.60% accuracy.

Author	Data Collection Site	Data Type	Model	Accuracy
Maity <i>et al.</i> [37]	Twitter and Reddit memes	Memes	BERT, ResNET, GRU	Text accuracy: 61.14% and Image accuracy 63.36%
Hamza <i>et al.</i> [38]	Twitter, Instagram, Facebook, and Reddit	Memes	RexNeXT-152-based Masked R-CNN, BERT	accuracy 70.60%
<b>Our Result (Public Dataset)</b>	From the study of Maity <i>et al.</i> [37], and Hamza <i>et al.</i> [38]	Text, Image, Memes (Multi-modal)	RoBERTa, ViT	accuracy RoBERTa: 99.2%, ViT: 99.5%, and Hybrid(RoBERTa+ViT): 99.2%

Table 12: Comparison with Applied Literature's Dataset for Public Dataset

## 6.2 Verifying our Results with the Private Dataset

During the verification process with the private dataset, each of our models was subjected to thorough evaluation to assess their efficacy in dealing with unseen data. The examination focused on the RoBERTa ( see section:4.4.6 ), ViT (see section:4.5.3) model, and Multi-modal (see section:4.6) models, with their performance evaluated comprehensively on a dataset that included both textual and image, and memes information.

The evaluation results, shown in table:11, demonstrate the RoBERTa model's proficiency, with outstanding performance metrics across all fronts. Notably, with an accuracy, recall, F1-score, and precision of **98.2%**, the model demonstrated robustness and generalization abilities, indicating its efficacy in a variety of contexts. This exceptional performance was consistent across all classes (0 through 3), with accuracies ranging from **98.4% to 99.8%**, as shown in table:23. Furthermore, the visual aids

provided in figures:63 and 62, namely the confusion matrices and ROC-AUC graphs, demonstrated the model's ability to perform classification tasks with few false positives and negatives.

In contrast, the ViT model, while slightly following the RoBERTa in terms of overall accuracy (**93.2%**), demonstrated outstanding results across key metrics. The accuracy was consistent across classes, with values of 0.961, 0.942, 0.971, and 0.990 for classes 0, 1, 2, and 3, respectively, as shown in table:23. Similarly, the associated visual representations in figures:65 and 64 provided compelling evidence of the ViT model's classification performance, with a high concentration of true positives and low rates of false positives and negatives.

In summary, both the RoBERTa and ViT models together as hybrid (RoBERTa+ViT) model demonstrate efficacy with accuracy of 0.961, and ROC-AUC value of 0.99 in addressing the multi-class cyberbullying classification task for multi-modal data, albeit with subtle differences in performance. The robustness of these models in handling textual, images, and multi-modal data highlights their potential utility in practical applications that address similar challenges.

### **6.3 Comparison with Existing Literature**

Table:13 shows a comprehensive comparison of this thesis work with the existing literature for multi-class cyberbullying for multi-modal data. Various scholars have investigated a variety of methodologies and feature sets to improve classification accuracy. Prior efforts, such as those by Maity *et al.* [37] and Titil *et al.* [27], have yielded accuracies ranging from 60-80%, utilizing diverse data from platforms such as Twitter, YouTube, and Instagram. Their techniques combine BERT, ResNET, GRU, and CNN models. Notably, Ahmadinejad *et al.* [36] achieved a high level of 99.80% accuracy on text data using the RoBERTa model, though this was limited to a text data type research. Barse *et al.* showed 96.50% accuracy on their research on text data only

In contrast, our work advances on this cyberbullying classification field by using hybrid (RoBERTa+ViT) model on both public and private datasets, resulting in near-perfect accuracy, f1-score, precision, recall of 99.2%, 0.992, 0.992, and 0.992 respectively for public data, and accuracy, f1-score, precision, recall of 96.1%, 0.959, 0.960, and 0.959 respectively for private data.

Author	Data Collection Site	Data Type	Model	Performance
Maity <i>et al.</i> [37]	Twitter and Reddit memes	Memes	BERT, ResNET, GRU	Text accuracy: 61.14% and Image accuracy 63.36%
Titli <i>et al.</i> [27]	YouTube	Text	Bangali-BERT	accuracy 70.60%
Kumari <i>et al.</i> [110]	Facebook, Twitter, Instagram	Memes	CNN, BPSO	F1-Score of 0.74
Hossain <i>et al.</i> [105]	Facebook, Twitter, Instagram	Memes	VGG19 and m-distilBERT	Weighted F1-scores of 66.73% and 58.59%
Mollas <i>et al.</i> [108]	YouTube and Reddit	Text	DistilBERT, BiLSTM	accuracy 80.36%
Ahmadinejad <i>et al.</i> [36]	Twitter	Text	RoBERTa	accuracy 99.80%
Barse <i>et al.</i> [111]	YouTube, tiktok, twitter and other social site	Text	Random Forest Classifier	accuracy 96.50%
<b>Our Result</b>	Public Dataset	Text, Image, and memes	RoBERTa, ViT, Hybrid model (RoBERTa+ViT)	RoBERTa: 99.2%, ViT: 99.5% and Hybrid (RoBERTa+ViT):99.2%
<b>Our Result</b>	Private Dataset	Text, Image and memes	RoBERTa, ViT, Hybrid model (RoBERTa + ViT)	accuracy RoBERTa: 98.6%, ViT: 96.2% and hybrid (RoBERTa+ViT) model: 96.1%

Table 13: Comparison with Existing Literature of Multi-class Classification of Cyberbullying

## 7 Conclusion and Future Work

In this section, we have defined our overall work in the thesis in subsection:7.1, and we have defined our future work in subsection:7.2.

### 7.1 Conclusion

Social media platforms such as Facebook, Twitter, and Instagram have revolutionized online interaction by enabling us to share and engage with diverse forms of content, particularly short videos. These videos frequently receive numerous comments in different formats, including text, images, and memes which is collectively referred to as multimodal data. Social media facilitates positive comments, but it also makes negative comments, that occurs cyberbullying. Serious psychological effects, such as depression and low self-esteem, can result from cyberbullying.

Although, many work was done on classifying cyberbullying to solve or reduce the cyberbullying issue from social media platform, most of the research focused on binary classification using multi-modal data, or multi-classification using textual data. There was a noticeable gap in the multi-class classification of cyberbullying using multimodal data, despite notable advances in deep learning techniques for cyberbullying classification. This thesis attempted to fill this gap by using a hybrid deep learning approach by combining models, Vision Transformer (ViT) for images and RoBERTa for text to accurately classify cyberbullying using multi-modal data types.

Both public and private datasets were used to accomplish this. The private dataset was gathered via APIFY from comments on short videos posted on a variety of social media platforms, whereas the public dataset was sourced from previously published research works. The outcomes of the experiment showed how well the models in use performed. The public dataset, the RoBERTa model outperformed other models such as hybrid (CNN+LSTM), LSTM, GRU, BERT, and DistilBERT, achieving an accuracy of 99.2% and F1-score of 0.992 for text data. With an accuracy of 99.5% and F1-score of 0.995 in classifying image data, the ViT model outperformed the CNN and ResNet models for public data. On the other side, ViT model obtained F1-score of 0.9319 and an accuracy of 93.20% for image data, while RoBERTa achieved F1-score of 0.986 and an accuracy of 98.6% for text data on the private dataset. In multimodal cyberbullying classification, the hybrid model (RoBERTa+ViT) demonstrated impressive results, reaching up to 99.2% accuracy and F1-score of 0.992 on public datasets and 96.1% accuracy and F1-score of 0.96 on private datasets.



Based on our results, we believe that, deep learning models such as RoBERTa and Vision Transformer (ViT) models are well effective at classifying multiple types of cyberbullying. RoBERTa works well with text, producing nearly flawless results, whereas ViT excels at handling images. Furthermore, when these models are combined into a hybrid (RoBERTa+ViT) model, they perform even better at multi-class classifying cyberbullying in multi-modal content, such as memes.

## 7.2 Future Work

In future work, the plan is to focus on the following work:

- We will try to use multi-label classification for representing the work more realistic. As a result, if a comment contains aggressive content with bullying, the result will be display for both aggressive and bullying classification type cyberbullying.
- In this thesis, we have only focused on English language data. We will try to collect more data on multi-languages for multi-class classification on multi-modal data, so that we can classify cyberbullying from multiple language, such as bengali<sup>xxxviii</sup>, hindi<sup>xxxix</sup>, urdhu<sup>xl</sup>, and norwegian<sup>xli</sup>.
- We only worked on text, image and multimodal like memes data for the private dataset. In future we will try to work with uploaded stickers and GIF<sup>xlii</sup> data along with our existing data to classify multi-class cyberbullying. As a result, we will be able to handle all kinds of data from the comments sections of social media short videos.

---

<sup>xxxviii</sup>[https://en.wikipedia.org/wiki/Bengali\\_language](https://en.wikipedia.org/wiki/Bengali_language)

<sup>xxxix</sup><https://en.wikipedia.org/wiki/Hindi>

<sup>xl</sup><https://en.wikipedia.org/wiki/Urdu>

<sup>xli</sup>[https://en.wikipedia.org/wiki/Norwegian\\_language](https://en.wikipedia.org/wiki/Norwegian_language)

<sup>xlii</sup><https://giphy.com/>

## References

- [1] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5929–5955, 2020.
- [2] Y. Fang, S. Yang, B. Zhao, and C. Huang, "Cyberbullying detection in social networks using bi-gru with self-attention mechanism," *Information*, vol. 12, no. 4, p. 171, 2021.
- [3] Z. Huang, C. Low, M. Teng, H. Zhang, D. E. Ho, M. S. Krass, and M. Grabmair, "Context-aware legal citation recommendation using deep learning," in *Proceedings of the eighteenth international conference on artificial intelligence and law*, pp. 79–88, 2021.
- [4] V. H. Phung and E. J. Rhee, "A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets," *Applied Sciences*, vol. 9, no. 21, p. 4500, 2019.
- [5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.
- [6] A. Tasdelen and B. Sen, "A hybrid cnn-lstm model for pre-mirna classification," *Scientific reports*, vol. 11, no. 1, p. 14125, 2021.
- [7] S. Gundapu and R. Mamidi, "Transformer based automatic covid-19 fake news detection system," *arXiv preprint arXiv:2101.00180*, 2021.
- [8] H. Adel, A. Dahou, A. Mabrouk, M. Abd Elaziz, M. Kayed, I. M. El-Henawy, S. Alshathri, and A. Amin Ali, "Improving crisis events detection using distilbert with hunger games search algorithm," *Mathematics*, vol. 10, no. 3, p. 447, 2022.
- [9] A. Mayfield, "What is social media," 2008.
- [10] D. Le Compte and D. Klug, "'it's viral!'—a study of the behaviors, practices, and motivations of tiktok users and social activism," in *Companion publication of the 2021 conference on computer supported cooperative work and social computing*, pp. 108–111, 2021.
- [11] D. B. V. Kaye, J. Zeng, and P. Wikstrom, *TikTok: Creativity and culture in short video*. John Wiley & Sons, 2022.
- [12] M. Koutamanis, H. G. Vossen, and P. M. Valkenburg, "Adolescents' comments in social media: Why do adolescents receive negative feedback and who is most at risk?," *Computers in Human Behavior*, vol. 53, pp. 486–494, 2015.

- [13] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 689–696, 2011.
- [14] L. H. Collantes, Y. Martafian, S. N. Khofifah, T. K. Fajarwati, N. T. Lassela, and M. Khairunnisa, "The impact of cyberbullying on mental health of the victims," in *2020 4th International Conference on Vocational Education and Training (ICOVET)*, pp. 30–35, IEEE, 2020.
- [15] S. Livingstone, L. Haddon, U. Hasebrink, K. Ólafsson, B. O'Neill, D. Smahel, and E. Staksrud, "Eu kids online: Findings, methods, recommendations," *LSE, London: EU Kids Online*. Available on <http://lsedesignunit.com/EUKidsOnline>, 2014.
- [16] R. S. Tokunaga, "Following you home from school: A critical review and synthesis of research on cyberbullying victimization," *Computers in human behavior*, vol. 26, no. 3, pp. 277–287, 2010.
- [17] J. Qiu, M. Moh, and T.-S. Moh, "Multi-modal detection of cyberbullying on twitter," in *Proceedings of the 2022 ACM Southeast Conference*, pp. 9–16, 2022.
- [18] Y. Chen, Y. Zhou, S. Zhu, and H. Xu, "Detecting offensive language in social media to protect adolescent online safety," in *2012 international conference on privacy, security, risk and trust and 2012 international conference on social computing*, pp. 71–80, IEEE, 2012.
- [19] J. Van der Zwaan, V. Dignum, and C. Jonker, "Simulating peer support for victims of cyberbullying," in *Proceedings of the 22st Benelux conference on artificial intelligence (BNAIC 2010)*, 2010.
- [20] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean, "A guide to deep learning in healthcare," *Nature medicine*, vol. 25, no. 1, pp. 24–29, 2019.
- [21] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [22] K. Wang, Q. Xiong, C. Wu, M. Gao, and Y. Yu, "Multi-modal cyberbullying detection on social networks," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2020.
- [23] S. Agrawal and A. Awekar, "Deep learning for detecting cyberbullying across multiple social media platforms," in *Advances in Information Retrieval: 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings*, pp. 141–153, Springer, 2018.

- [24] J. Batani, E. Mbunge, B. Muchemwa, G. Gaobotse, C. Gurajena, S. Fashoto, T. Kavuu, and K. Dandajena, "A review of deep learning models for detecting cyberbullying on social media networks," in *Computer Science On-line Conference*, pp. 528–550, Springer, 2022.
- [25] M. T. Hasan, M. A. E. Hossain, M. S. H. Mukta, A. Akter, M. Ahmed, and S. Islam, "A review on deep-learning-based cyberbullying detection," *Future Internet*, vol. 15, no. 5, p. 179, 2023.
- [26] D. Sultan, B. Omarov, Z. Kozhamkulova, G. Kazbekova, L. Alimzhanova, A. Dautbayeva, Y. Zholdassov, and R. Abdrakhmanov, "A review of machine learning techniques in cyberbullying detection," *Computers, Materials & Continua*, vol. 74, no. 3, 2023.
- [27] S. R. Titli and S. Paul, "Automated bengali abusive text classification: Using deep learning techniques," in *2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS)*, pp. 1–6, IEEE, 2023.
- [28] A. Kumar and N. Sachdeva, "Multimodal cyberbullying detection using capsule network with dynamic routing and deep convolutional neural network," *Multimedia Systems*, pp. 1–10, 2021.
- [29] R. Kumar, A. K. Ojha, S. Malmasi, and M. Zampieri, "Benchmarking aggression identification in social media," in *Proceedings of the first workshop on trolling, aggression and cyberbullying (TRAC-2018)*, pp. 1–11, 2018.
- [30] D. Chatzakou, I. Leontiadis, J. Blackburn, E. D. Cristofaro, G. Stringhini, A. Vakali, and N. Kourtellis, "Detecting cyberbullying and cyberaggression in social media," *ACM Transactions on the Web (TWEB)*, vol. 13, no. 3, pp. 1–51, 2019.
- [31] S. Chandrasekaran, A. K. Singh Pundir, T. B. Lingaiah, *et al.*, "Deep learning approaches for cyberbullying detection and classification on social media," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.
- [32] M. Dadvar and K. Eckert, "Cyberbullying detection in social networks using deep learning based models," in *Big Data Analytics and Knowledge Discovery: 22nd International Conference, DaWaK 2020, Bratislava, Slovakia, September 14–17, 2020, Proceedings 22*, pp. 245–255, Springer, 2020.
- [33] N. K. Singh, P. Singh, and S. Chand, "Deep learning based methods for cyberbullying detection on social media," in *2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pp. 521–525, IEEE, 2022.
- [34] M. Alotaibi, B. Alotaibi, and A. Razaque, "A multichannel deep learning framework for cyberbullying detection on social media," *Electronics*, vol. 10, no. 21, p. 2664, 2021.

- [35] A. Faraj and S. Utku, "Comparative analysis of word embeddings for multiclass cyberbullying detection," *UHD Journal of Science and Technology*, vol. 8, no. 1, pp. 55–63, 2024.
- [36] M. Ahmadinejad, N. Shahriar, and L. Fan, *Self-Training for Cyberbully Detection: Achieving High Accuracy with a Balanced Multi-Class Dataset*. PhD thesis, Faculty of Graduate Studies and Research, University of Regina, 2023.
- [37] K. Maity, P. Jha, S. Saha, and P. Bhattacharyya, "A multitask framework for sentiment, emotion and sarcasm aware cyberbullying detection from multi-modal code-mixed memes," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1739–1749, 2022.
- [38] A. Hamza, A. R. Javed, F. Iqbal, A. Yasin, G. Srivastava, D. Połap, T. R. Gadekallu, and Z. Jalil, "Multimodal religiously hateful social media memes classification based on textual and image data," *ACM Transactions on Asian and Low-Resource Language Information Processing*, 2023.
- [39] C. Van Hee, G. Jacobs, C. Emmery, B. Desmet, E. Lefever, B. Verhoeven, G. De Pauw, W. Daelemans, and V. Hoste, "Automatic detection of cyberbullying in social media text," *PloS one*, vol. 13, no. 10, p. e0203794, 2018.
- [40] I. Niiniluoto, "The aim and structure of applied research," *Erkenntnis*, vol. 38, no. 1, pp. 1–21, 1993.
- [41] S. B. Mishra and S. Alok, "Handbook of research methodology," 2011.
- [42] M. Baimyrzaeva, "Beginners' guide for applied research process: What is it, and why and how to do it," *University of Central Asia*, vol. 4, no. 8, pp. 1–42, 2018.
- [43] M. Drahošová and P. Balco, "The analysis of advantages and disadvantages of use of social media in european union," *Procedia Computer Science*, vol. 109, pp. 1005–1009, 2017.
- [44] P. Yi and A. Zubiaga, "Session-based cyberbullying detection in social media: A survey," *Online Social Networks and Media*, vol. 36, p. 100250, 2023.
- [45] F. W. Putra and M. Ramli, "Cognitive behavior counseling to help victims of cyberbullying: systematic review," *Konselor*, vol. 11, no. 3, pp. 98–103, 2022.
- [46] J. E. Copp, E. A. Mumford, and B. G. Taylor, "Online sexual harassment and cyberbullying in a nationally representative sample of teens: Prevalence, predictors, and consequences," *Journal of adolescence*, vol. 93, pp. 202–211, 2021.

- [47] P. Suanpang *et al.*, "Lgbtq cyberbullying on online learning platforms among university students," *International Journal of Cyber Criminology*, vol. 15, no. 2, 2022.
- [48] L. Edwards, A. E. Kontostathis, and C. Fisher, "Cyberbullying, race/ethnicity and mental health outcomes: A review of the literature," *Media and Communication*, vol. 4, no. 3, pp. 71–78, 2016.
- [49] T. T. Ojanen and R. Sittichai, "Sogie, bullying, and cyberbullying in thai schools," in *SOGI Minority and School Life in Asian Contexts*, pp. 119–134, Routledge, 2023.
- [50] K. Sulastri, "Woman and cyberbullying," *JUDIMAS*, vol. 1, no. 1, pp. 26–37, 2021.
- [51] A. F. Borges, F. J. Laurindo, M. M. Spínola, R. F. Gonçalves, and C. A. Mattos, "The strategic use of artificial intelligence in the digital era: Systematic literature review and future research directions," *International Journal of Information Management*, vol. 57, p. 102225, 2021.
- [52] J. H. Fetzer and J. H. Fetzer, *What is Artificial Intelligence?* Springer, 1990.
- [53] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR)*.*[Internet]*, vol. 9, no. 1, pp. 381–386, 2020.
- [54] M. Ibañez, R. Sapinit, L. A. Reyes, M. Hussien, J. M. Imperial, and R. Rodriguez, "Audio-based hate speech classification from online short-form videos," in *2021 International Conference on Asian Language Processing (IALP)*, pp. 72–77, IEEE, 2021.
- [55] M. Arif, "A systematic review of machine learning algorithms in cyberbullying detection: future directions and challenges," *Journal of Information Security and Cybercrimes Research*, vol. 4, no. 1, pp. 01–26, 2021.
- [56] A. C. de Carvalho and A. A. Freitas, "A tutorial on multi-label classification techniques," *Foundations of Computational Intelligence Volume 5: Function Approximation and Classification*, pp. 177–195, 2009.
- [57] M. J. Er, R. Venkatesan, and N. Wang, "An online universal classifier for binary, multi-class and multi-label classification," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 003701–003706, IEEE, 2016.
- [58] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 3, no. 3, pp. 1–13, 2007.

- [59] R. Haque, N. Islam, M. Tasneem, and A. K. Das, "Multi-class sentiment classification on bengali social media comments using machine learning," *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 21–35, 2023.
- [60] M.-L. Zhang and Z.-H. Zhou, "MI-knn: A lazy learning approach to multi-label learning," *Pattern recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.
- [61] R. Rana, "Gated recurrent unit (gru) for emotion classification from noisy speech," *arXiv preprint arXiv:1612.07778*, 2016.
- [62] G. Yiğit and M. F. Amasyali, "Simple but effective gru variants," in *2021 international conference on INnovations in intelligent SysTems and applications (INISTA)*, pp. 1–6, IEEE, 2021.
- [63] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [64] K. L. Tan, C. P. Lee, K. S. M. Anbananthen, and K. M. Lim, "Roberta-lstm: a hybrid model for sentiment analysis with transformer and recurrent neural network," *IEEE Access*, vol. 10, pp. 21517–21525, 2022.
- [65] X. Li and H. Ning, "Chinese text classification based on hybrid model of cnn and lstm," in *Proceedings of the 3rd international conference on data science and information technology*, pp. 129–134, 2020.
- [66] X. She and D. Zhang, "Text classification based on hybrid cnn-lstm hybrid model," in *2018 11th International symposium on computational intelligence and design (ISCID)*, vol. 2, pp. 185–189, IEEE, 2018.
- [67] J. Zhang, Y. Li, J. Tian, and T. Li, "Lstm-cnn hybrid model for text classification," in *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 1675–1680, IEEE, 2018.
- [68] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [69] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.
- [70] H. Alibrahim and S. A. Ludwig, "Hyperparameter optimization: Comparing genetic algorithm against grid search and bayesian optimization," in *2021 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1551–1559, IEEE, 2021.

- [71] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization.," *Journal of machine learning research*, vol. 13, no. 2, 2012.
- [72] S. R. Young, D. C. Rose, T. P. Karnowski, S.-H. Lim, and R. M. Patton, "Optimizing deep learning hyper-parameters through an evolutionary algorithm," in *Proceedings of the workshop on machine learning in high-performance computing environments*, pp. 1–5, 2015.
- [73] S. Li and W. Deng, "Blended emotion in-the-wild: Multi-label facial expression recognition using crowdsourced annotations and deep locality feature learning," *International Journal of Computer Vision*, vol. 127, no. 6, pp. 884–906, 2019.
- [74] A. Kavitha, P. Shivakumara, G. Kumar, and T. Lu, "Text segmentation in degraded historical document images," *Egyptian informatics journal*, vol. 17, no. 2, pp. 189–197, 2016.
- [75] G. A. Miller and P. E. Nicely, "An analysis of perceptual confusions among some english consonants," *The Journal of the Acoustical Society of America*, vol. 27, no. 2, pp. 338–352, 1955.
- [76] G. Canbek, S. Sagiroglu, T. Taskaya Temizel, and N. Baykal, "Binary classification performance measures/metrics: A comprehensive visualized roadmap to gain new insights," pp. 821–826, 10 2017.
- [77] V. Labatut and H. Cherifi, "Evaluation of performance measures for classifiers comparison," *arXiv preprint arXiv:1112.4133*, 2011.
- [78] T. Fawcett, "An introduction to roc analysis: Pattern recognition letter, v. 27," 2006.
- [79] S. Koço and C. Capponi, "On multi-class learning through the minimization of the confusion matrix norm," *arXiv preprint arXiv:1303.4015*, 2013.
- [80] D. Krstinić, M. Braović, L. Šerić, and D. Božić-Štulić, "Multi-label classifier performance evaluation with confusion matrix," *Computer Science & Information Technology*, vol. 1, pp. 1–14, 2020.
- [81] N. Gehlenborg and B. Wong, "Heat maps," *Nature Methods*, vol. 9, no. 3, p. 213, 2012.
- [82] E. Kaderabek and P. Suwannajang, "Confusion matrix viz,"
- [83] B. S. Nandhini and J. Sheeba, "Cyberbullying detection and classification using information retrieval algorithm," in *Proceedings of the 2015 international conference on advanced research in computer science engineering & technology (ICARCSET 2015)*, pp. 1–5, 2015.
- [84] J. Muschelli III, "Roc and auc with a binary predictor: a potentially misleading metric," *Journal of classification*, vol. 37, no. 3, pp. 696–708, 2020.



- [85] L. Prechelt, "Early stopping-but when?," in *Neural Networks: Tricks of the trade*, pp. 55–69, Springer, 2002.
- [86] S. S. McPherson, *Tim Berners-Lee: Inventor of the World Wide Web*. Twenty-First Century Books, 2009.
- [87] K. Jamsa, K. King, and A. Anderson, *HTML & Web Design*. McGraw-Hill, 2002.
- [88] A. Felt, P. Hooimeijer, D. Evans, and W. Weimer, "Talking to strangers without taking their candy: isolating proxied content," in *Proceedings of the 1st Workshop on Social Network Systems*, pp. 25–30, 2008.
- [89] G. van Rossum and F. L. Drake, *An introduction to Python*. Network Theory Limited, 2006.
- [90] D. S. R. Sukhdeve and S. S. Sukhdeve, "Google colab," in *Google Cloud Platform for Data Science: A Crash Course on Big Data, Machine Learning, and Data Analytics Services*, pp. 11–34, Springer, 2023.
- [91] E. Bisong and E. Bisong, "Google colab," *Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners*, pp. 59–64, 2019.
- [92] V. S. Code, "Visual studio code," *Recuperado el Octubre de*, 2019.
- [93] K. Kumari and J. P. Singh, "Identification of cyberbullying on multi-modal social media posts using genetic algorithm," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 2, p. e3907, 2021.
- [94] K. Kumari, J. P. Singh, Y. K. Dwivedi, and N. P. Rana, "Towards cyberbullying-free social media in smart cities: a unified multi-modal approach," *Soft computing*, vol. 24, pp. 11059–11070, 2020.
- [95] N. M. Singh and S. K. Sharma, "An efficient automated multi-modal cyberbullying detection using decision fusion classifier on social media platforms," *Multimedia Tools and Applications*, pp. 1–29, 2023.
- [96] E. ILAVARASAN *et al.*, "Cyberbullying detection on multi-modal data using pre-trained deep learning architectures," *Ingeniería Solidaria*, vol. 17, no. 3, pp. 1–20, 2021.
- [97] G. Singh, S. Kumar, S. Vijayan, T. Perumal, M. Sathiyarayanan, and R. Campus, "Cyber bullying detection using machine learning and deep learning,"
- [98] S. Paul, S. Saha, and M. Hasanuzzaman, "Identification of cyberbullying: A deep learning based multimodal approach," *Multimedia Tools and applications*, pp. 1–20, 2020.

- [99] N. Romim, M. Ahmed, H. Talukder, and M. Saiful Islam, "Hate speech detection in the bengali language: A dataset and its baseline evaluation," in *Proceedings of International Joint Conference on Advances in Computational Intelligence: IJACAI 2020*, pp. 457–468, Springer, 2021.
- [100] E. Idrizi and M. Hamiti, "Classification of text, image and audio messages used for cyberbullying on social medias," in *2023 46th MIPRO ICT and Electronics Convention (MIPRO)*, pp. 797–802, IEEE, 2023.
- [101] R. Jadhav and V. N. Honmane, "Memes classification system using computer vision and nlp techniques," *International Journal of Engineering Applied Sciences and Technology*, 2021.
- [102] D. Kiela, H. Firooz, A. Mohan, V. Goswami, A. Singh, P. Ringshia, and D. Testuggine, "The hateful memes challenge: Detecting hate speech in multimodal memes," *Advances in neural information processing systems*, vol. 33, pp. 2611–2624, 2020.
- [103] H. Fang, F. Zhu, J. Han, and S. Hu, "Multimodal hateful memes detection via image caption supervision," in *2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta)*, pp. 1530–1537, IEEE, 2022.
- [104] A. Chhabra and D. K. Vishwakarma, "Multimodal hate speech detection via multi-scale visual kernels and knowledge distillation architecture," *Engineering Applications of Artificial Intelligence*, vol. 126, p. 106991, 2023.
- [105] E. Hossain, O. Sharif, M. M. Hoque, M. A. A. Dewan, N. Siddique, and M. A. Hossain, "Identification of multilingual offense and troll from social media memes using weighted ensemble of multimodal features," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 9, pp. 6605–6623, 2022.
- [106] J. Hani, N. Mohamed, M. Ahmed, Z. Emad, E. Amer, and M. Ammar, "Social media cyberbullying detection using machine learning," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 5, 2019.
- [107] R. Koshy and S. Elango, "Multimodal tweet classification in disaster response systems using transformer-based bidirectional attention model," *Neural Computing and Applications*, vol. 35, no. 2, pp. 1607–1627, 2023.
- [108] I. Mollas, Z. Chrysopoulou, S. Karlos, and G. Tsoumakas, "Ethos: a multi-label hate speech detection dataset," *Complex & Intelligent Systems*, vol. 8, no. 6, pp. 4663–4678, 2022.

- [109] M. R. Karim, S. K. Dey, T. Islam, M. Shajalal, and B. R. Chakravarthi, "Multimodal hate speech detection from bengali memes and texts," in *International Conference on Speech and Language Technologies for Low-resource Languages*, pp. 293–308, Springer, 2022.
- [110] K. Kumari, J. P. Singh, Y. K. Dwivedi, and N. P. Rana, "Multi-modal aggression identification using convolutional neural network and binary particle swarm optimization," *Future Generation Computer Systems*, vol. 118, pp. 187–197, 2021.
- [111] S. Barse, D. Bhagat, K. Dhawale, Y. Solanke, and D. Kurve, "Cyber-trolling detection system," Available at SSRN 4340372, 2023.
- [112] C. Van Hee, E. Lefever, B. Verhoeven, J. Mennes, B. Desmet, G. De Pauw, W. Daelemans, and V. Hoste, "Detection and fine-grained classification of cyberbullying events," in *Proceedings of the international conference recent advances in natural language processing*, pp. 672–680, 2015.
- [113] N. Z. Abidin, A. R. Ismail, and N. A. Emran, "Performance analysis of machine learning algorithms for missing value imputation," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 6, 2018.
- [114] V. Gudivada, A. Apon, and J. Ding, "Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations," *International Journal on Advances in Software*, vol. 10, no. 1, pp. 1–20, 2017.
- [115] A. Dewani, M. A. Memon, and S. Bhatti, "Cyberbullying detection: advanced preprocessing techniques & deep learning architecture for roman urdu data," *Journal of big data*, vol. 8, no. 1, p. 160, 2021.
- [116] S. Ahsan, E. Hossain, O. Sharif, A. Das, M. M. Hoque, and M. Dewan, "A multimodal framework to detect target aware aggression in memes," in *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2487–2500, 2024.
- [117] M. Paciello, F. D'Errico, G. Saleri, and E. Lamponi, "Online sexist meme and its effects on moral and emotional processes in social media," *Computers in human behavior*, vol. 116, p. 106655, 2021.
- [118] S. Sharma, F. Alam, M. S. Akhtar, D. Dimitrov, G. D. S. Martino, H. Firooz, A. Halevy, F. Silvestri, P. Nakov, and T. Chakraborty, "Detecting and understanding harmful memes: A survey," *arXiv preprint arXiv:2205.04274*, 2022.

# Appendices

## List of Appendices

Appendix

131

# Appendix A Appendix

## A.1 Performance Evaluation of Public Data for Each Classes

### A.1.1 Performance Evaluation of Textual Data

In this section, we describe all the confusion matrix for each classes using textual data of public dataset.

#### A.1.1.1 Experiment with hybrid (CNN+LSTM) model for each classes

Table:14 present the performance of each classes individually hybrid (CNN+LSTM) model, to present that how the model was performed. In this table, we can observe that class 0 was well-classified, whereas Classes 1 and 2 exhibit a complete breakdown in effective prediction. Despite its high recall, Class 3 had poor precision, indicating that the model needed to be adjusted to reduce false positives and improve overall accuracy.

Class Label	Accuracy	Recall	F1-Score	Precision
0	0.98	0.989	0.964	0.940
1	0.74	0.00	0.00	0.00
2	0.76	0.00	0.00	0.00
3	0.50	0.977	0.488	0.325

Table 14: Performance of Each Classes by Hybrid(CNN+LSTM) Model

#### Confusion Matrix:

For the Hybrid (CNN+LSTM) model confusion matrix, class 0 ( see figure:85) performs reasonably well. 455 instances were correctly classified as Class 0 by the model. Nevertheless, it incorrectly classified 29 instances of other classes as Class 0 and 5 instances of Class 0 are incorrectly classified as another class. Class 1 ( figure:86) poses a serious problem since the model did not correctly classify any instance of Class 1, misidentifying all 468 genuine instances of Class 1 as non-Class 1, and it did not mistakenly identify any other class as Class 1. With 436 accurate predictions and only one instance where Class 2 (figure:87) was mistakenly identified as Class 2, Class 2 exhibits superior accuracy.

The amount of misclassification was negligible, with all inaccurate Class 2 predictions being classified as non-class 2. Class 3 (figure: 88) has a high false positive rate of 475 cases from other classes that were mistakenly classified as Class 3, despite having a good number of correct predictions (428). Ten

instances of Class 3 were misclassified as not being in Class 3.

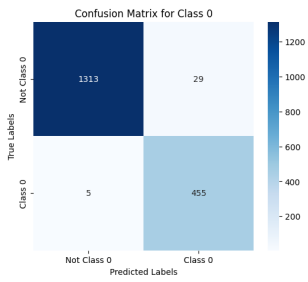


Figure 85: Confusion Matrix for Class 0 for Hybrid Model

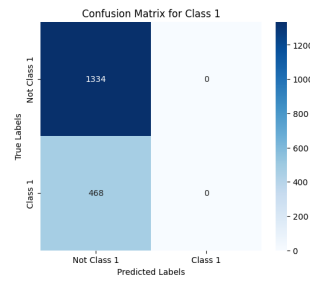


Figure 86: confusion matrix of class 1 for Hybrid model

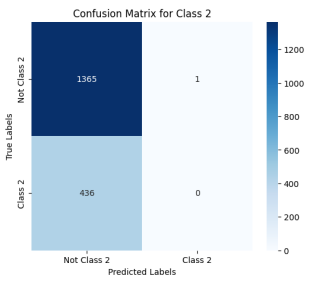


Figure 87: confusion matrix of class 2 for Hybrid model

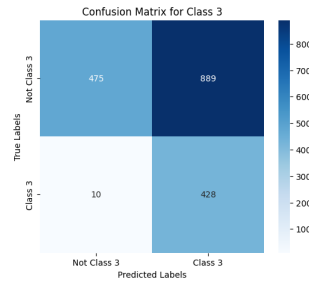


Figure 88: confusion matrix of class 3 for Hybrid model

### A.1.1.2 Experiment with LSTM model for each classes

Table:15 shows the performance for each classes by the LSTM model. While class 0 has excellent predictive performance, class 3 completely failed in prediction metrics. Class 1 and class 2 produce mixed results, with class 2 performing slightly better in recall but poorly in precision, and class 1 struggling overall, with particularly low recall.

Class Label	Accuracy	Recall	F1-Score	Precision
0	0.98	0.98	0.96	0.94
1	0.71	0.12	0.18	0.33
2	0.51	0.81	0.44	0.31
3	0.76	0.00	0.00	0.00

Table 15: Performance of Each Classes by LSTM Model

#### Confusion Matrix:

In the LSTM model, class 0, shown in figure:89, performs exceptionally well, with the model correctly predicting 453 instances as Class 0. However, there are some minor inaccuracies, such as 5 instances where Class 0 was misclassified and 29 instances where other classes were incorrectly classified as Class 0. Class 1, as shown in figure:90, has significant issues because the model fails to correctly classify any instance of this class, with all 468 instances incorrectly classified as not Class 1. This indicates a critical flaw in the model's ability to distinguish Class 1, and no instances from other classes were incorrectly classified as Class 1.

Figure:90, which depicts Class 2, shows a more positive result, with the model correctly identifying 351 instances. Despite the relatively high number of correct predictions, 85 instances were misclassified as not being in Class 2.

sified as not Class 2, with only one case of misclassification involving another class being labeled as Class 2. Class 3, as shown in figure:92, has a good recognition capability with 395 correct predictions but suffers from a high number of misclassifications. Notably, 33 instances were misidentified as not being Class 3, while a large number of other class instances (475) were incorrectly classified as Class 3, indicating a high false positive rate. Given the limitations of computational resources, as only a standard computer was available, the model was restricted to 20 epochs, which may have impeded achieving optimal performance. These factors highlight the need for additional training and potentially more computational resources to improve the model's overall efficacy.

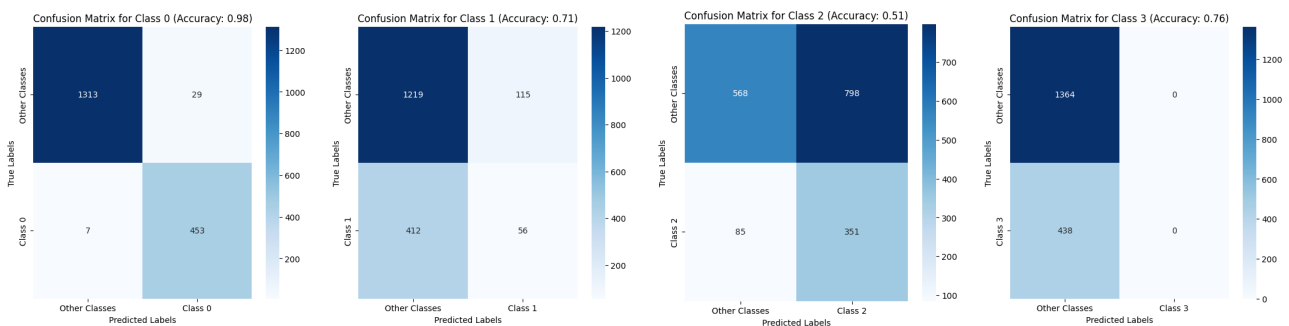


Figure 89: Confusion Matrix for Class 0 for LSTM Model  
 Figure 90: confusion matrix of class 1 for LSTM model  
 Figure 91: confusion matrix of class 2 for LSTM model  
 Figure 92: confusion matrix of class 3 for LSTM model

### A.1.1.3 Experiment with GRU model for each classes

Table:16 shows the performance for each classes by GRU model, where we can see that, the performance for four class 2 and 3 was not good. Class 0 and class 1 showed relatively high accuracy rates of 0.987 and 0.978, respectively. However, classes 2 and 3 displayed extremely poor results with no successful predictions, highlighting major deficiencies in the model's ability to recognize these classes accurately.

Class Label	Accuracy	Recall	F1-Score	Precision
0	0.987	0.99	0.96	0.93
1	0.978	0.98	0.51	0.34
2	0.0	0.00	0.00	0.00
3	0.0	0.00	0.00	0.00

Table 16: Performance of Each Classes by GRU Model

### Confusion Matrix:

In GRU model, the Class 0 matrix in figure:93 revealed a high number of correct predictions (466), but also some misclassifications, most notably mislabeling other class instances as Class 0. The matrix

for Class 1 in figure:94 presented a challenging scenario, with many instances incorrectly identified, demonstrating the model's difficulty with this class. The matrices for Classes 2 and 3 revealed a complete failure in prediction, with zero correct classifications, confirming the model's inadequacy in handling these classes in figure:95 and figure:96 respectively.

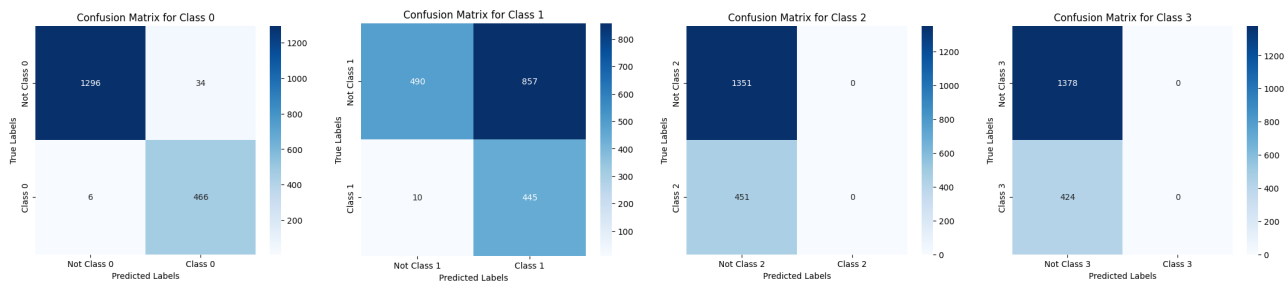


Figure 93: Confusion Matrix for Class 0 for GRU Model  
 Figure 94: confusion matrix of class 1 for GRU model  
 Figure 95: confusion matrix of class 2 for GRU model  
 Figure 96: confusion matrix of class 3 for GRU model

#### A.1.1.4 Experiment with BERT model for each classes

To be specific, table:17 represents the performance for each classes by BERT model. This model performed excellent for the four classes, to be more specific, it excelled at classifying textual data into four distinct classes 0, 1, 2, and 3 with 98.3%, 99.3%, 98.6%, 99.0% accuracy respectively and also indicating high recall, F1-score, and precision.

Class Label	Accuracy	Recall	F1-Score	Precision
0	0.983	0.944	0.965	0.988
1	0.993	0.996	0.987	0.979
2	0.986	1.00	0.973	0.948
3	0.990	0.966	0.980	0.995

Table 17: Performance of Each Classes by BERT Model

#### Confusion Matrix:

In the BERT model, Class 0 had 24 instances misclassified as not Class 0, while 5 instances from other classes were incorrectly classified as Class 0 (see figure:97). Class 1 had the fewest misclassifications, with only two instances incorrectly identified as not Class 1; however, it had the most false positives, with ten instances from other classes incorrectly labeled as Class 1 (see figure:98). Class 2 performed the best, with no misclassifications of true Class 2 instances, but it did classify 24 instances from other classes as Class 2, indicating a need for more specificity (see figure:99). Class 3 had a significant number of correct predictions, but it also had 15 instances misclassified as not Class 3 and two instances from other classes incorrectly identified as Class 3 (see figure:100).



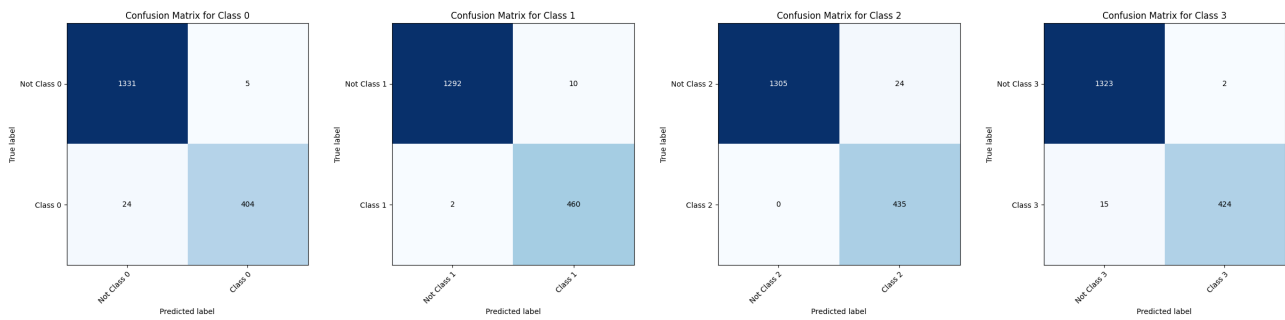


Figure 97: Confusion Matrix for Class 0 for BERT Model  
 Figure 98: confusion matrix of class 1 for BERT model  
 Figure 99: confusion matrix of class 2 for BERT model  
 Figure 100: confusion matrix of class 3 for BERT model

### A.1.1.5 Experiment with DistilBERT model for each classes

Furthermore, table:18 represents the performance for each classes by DistilBERT model, where it's clearly can see that, each classes performance was also excelent. Class 0 had an accuracy and recall of 99.1%, with a slightly lower f1-score and precision, indicating strong but not perfect predictive reliability. Class 1 has exceptionally high precision at 100% and an overall accuracy of 99.5%, demonstrating its exceptional ability to correctly identify and confirm instances of this class without error. Class 2 and class 3 both have near-perfect scores across all metrics, with class 2 achieving 99.8% accuracy and recall and class 3 also scoring 99.8% accuracy and 99.5% recall.

Class Label	Accuracy	Recall	F1-Score	Precision
0	0.991	0.991	0.981	0.972
1	0.995	0.981	0.990	1.00
2	0.998	0.998	0.995	0.993
3	0.998	0.995	0.997	0.997

Table 18: Performance of Each Classes by DistilBERT Model

**Confusion Matrix:** The DistilBERT model accurately identified 424 instances of Class 0 in figure:101, with only 4 misclassifications and 12 false positives, indicating high recall and precision. Class 1 produced excellent results, with 453 correct predictions and minimal errors, including only 9 misclassifications and no false positives, demonstrating the model's ability to accurately identify this category with high sensitivity and specificity (see figure:102. Class 2 continued to perform well, correctly classifying 434 instances and having very few misclassifications (figure:103 shows 3 instances from other classes wrongly labeled as Class 2 and only 1 instance classified as not belonging to Class 2). In the end, Class 3 achieved nearly perfect accuracy with 437 correct predictions; only 2 cases from other classes were misclassified as Class 3. This leads to an almost perfect classification record in figure:104.

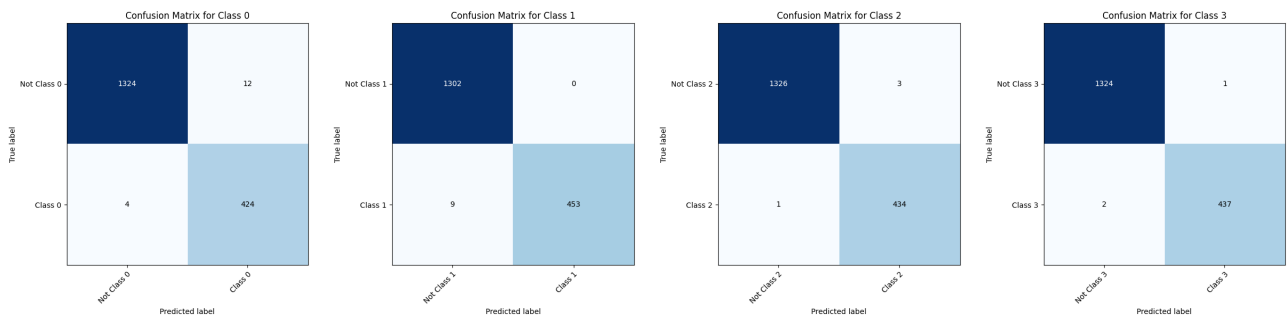


Figure 101: Confusion Matrix for Class 0 for DistilBERT Model  
 Figure 102: confusion matrix of class 1 for DistilBERT model  
 Figure 103: confusion matrix of class 2 for DistilBERT model  
 Figure 104: confusion matrix of class 3 for DistilBERT model

### A.1.1.6 Experiment with RoBERTa model for each classes

The performance for each classes by the RoBERTa model has depicted in table:19. The RoBERTa model performed admirably, scoring nearly perfect across all classes in terms of accuracy also. Class 0 had an accuracy of 0.992 and an F1-score of 0.984, while Class 1 had an even higher accuracy of 0.997 and an F1-score of 0.995%. Class 2 demonstrated a perfect recall score and an impressive F1-score of 0.991. Class 3 also performed well, with a 1.00 recall and a 0.998 F1-score, demonstrating the RoBERTa model’s ability to effectively handle a variety of textual classifications.

Class Label	Accuracy	Recall	F1-Score	Precision
0	0.992	0.977	0.984	0.993
1	0.997	0.991	0.995	0.998
2	0.995	1	0.991	0.982
3	0.998	1	0.998	0.995

Table 19: Performance of Each Classes by RoBERTa Model



Figure 105: Confusion Matrix for Class 0 for RoBERTa Model  
 Figure 106: confusion matrix of class 1 for RoBERTa model  
 Figure 107: confusion matrix of class 2 for RoBERTa model  
 Figure 108: confusion matrix of class 3 for RoBERTa model

### Confusion Matrix:

In RoBERTa model, for Class 0 in figure:105, the model correctly identified 418 instances, with only

10 misclassifications and three false positives, demonstrating high recall and precision. Class 1 also produced excellent results, with 458 correct predictions and few errors, including only four misclassifications and one false positive, demonstrating the model's ability to accurately identify this category with high sensitivity and specificity in figure:106. Class 2 maintained its strong performance, correctly identifying 435 instances and having very few misclassifications, with 8 instances classified as not belonging to Class 2 and no instances from other classes incorrectly labeled as Class 2 in figure:107. Finally, Class 3 had 439 correct predictions, with near-perfect accuracy; only two instances from other classes were incorrectly identified as Class 3, resulting in an almost flawless classification record in figure:108.

## A.1.2 Performance Evaluation of Image Data

In this section, we describe all the performance matrix for each classes using image data of public dataset.

### A.1.2.1 Experiment with ResNet model for each classes

Table:20 presents the performance for each classes by ResNet model, and the model exhibits strong performance in all four classes; in class 0, the results are 98% accurate, 88% precise, 93% recall, and 0.90 F1-score. Class 1 has an F1-score of 0.94, a high precision of 99%, a recall of 90%, and is less accurate at 83%. With 97% accuracy, 100% perfect precision, 98% recall, and an F1-score of 0.99, Class 2 performs exceptionally well. Last but not least, Class 3 attains the maximum accuracy of 99%, 88% precision, 93% recall, and an F1-score of 0.91.

Class Label	Accuracy	Precision	Recall	F1-Score
0	0.98	0.88	0.93	0.90
1	0.83	0.99	0.90	0.94
2	0.97	1.00	0.98	0.99
3	0.99	0.88	0.93	0.91

Table 20: Performance of Each Classes by ResNet Model

### Confusion Matrix:

In ResNet model, with 609 accurate classifications for Class 0 and few false positives and negatives, the model demonstrated strong identification of this class and high accuracy (see figure:109). Class 1 demonstrated exceptional performance as well, identifying 202 objects correctly and misclassifying only two, highlighting the accuracy of the model (see figure:109). With 203 accurate predictions and only one misclassification, Class 2 demonstrated nearly perfect accuracy, demonstrating its effectiveness in identifying this category (see figure:109). Last but not least, Class 3 had 180 accurate

predictions, maintaining a similar high accuracy even though it had a little bit more misclassifications than the other classes (see figure:109).

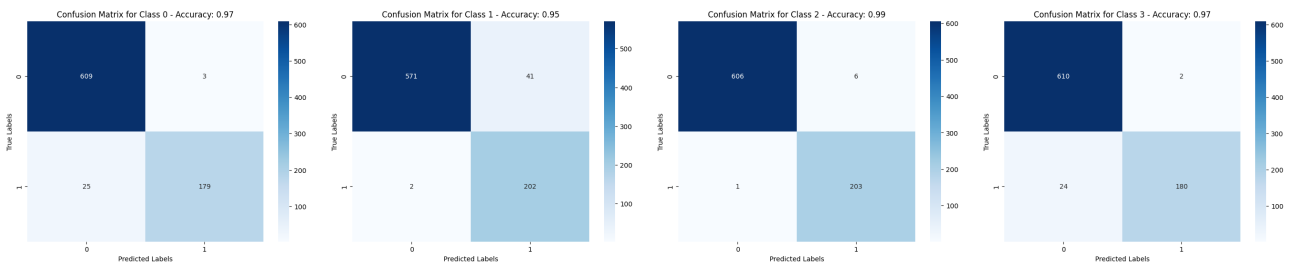


Figure 109: Confusion Matrix for Class 0 for ResNet Model  
 Figure 110: confusion matrix of class 1 for ResNet model  
 Figure 111: confusion matrix of class 2 for ResNet model  
 Figure 112: confusion matrix of class 3 for ResNet model

### A.1.2.2 Experiment with CNN model for each classes

The CNN model demonstrated impressive performance across all classes in table:21. In class 0, it scored a 99.0% accuracy, a 99.0% recall, a 98.0% F1-score, and a 97.0% precision. These scores imply that the model was especially good at correctly recognizing and categorizing examples within this class. It obtained a 99.0% accuracy, 99.0% recall, 98.0% F1-score, and 97.0% precision for class 0. These scores imply that the model was especially good at correctly recognizing and categorizing examples within this class. Class 1 also achieved high results, maintaining a high F1-score of 99.5% and precision of 98.0% while having a somewhat lower recall of 95.0%. The model performed exceptionally well in classes 2 and 3, achieving 100% accuracy, 100% recall, 100% F1-score, and 100% precision in all measures.

Class Label	Accuracy	Recall	F1-Score	Precision
0	0.99	0.99	0.98	0.97
1	0.98	0.95	0.995	0.98
2	1.00	1.00	0.99	0.99
3	0.99	0.99	0.99	0.99

Table 21: Performance of Each Classes by CNN Model

#### Confusion Matrix:

Class 0 for CNN model, as illustrated in figure:113, saw 605 correct identifications, with very few instances misclassified, demonstrating its reliable detection capabilities and high accuracy rate. In Class 1, as shown in figure:114, the model correctly identified 202 instances with only 10 inaccuracies, demonstrating its precision. Figure:115 shows that Class 2 performed exceptionally well, with 203 correct classifications and only one misclassification, demonstrating the model's nearly-perfect accuracy in this category. Class 3 (shown in figure:116) maintained high accuracy with 180 correct

predictions, but had a slight increase in misclassifications compared to other classes.

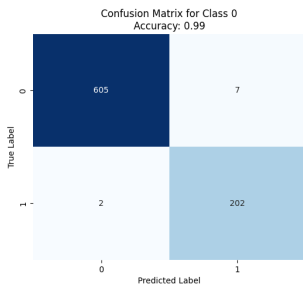


Figure 113: Confusion Matrix for Class 0 for CNN Model

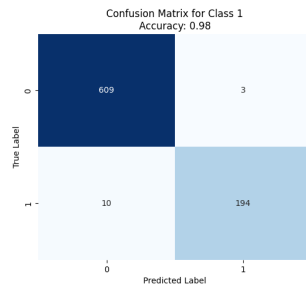


Figure 114: confusion matrix of class 1 for CNN model

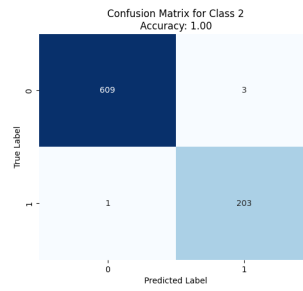


Figure 115: confusion matrix of class 2 for CNN model

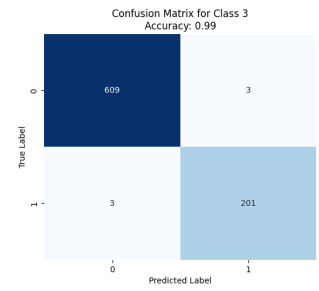


Figure 116: confusion matrix of class 3 for CNN model

### A.1.2.3 Experiment with ViT model for each classes

Class Label	Accuracy	Recall	F1-Score	Precision
0	0.99	0.99	0.99	0.99
1	0.99	0.99	0.99	0.99
2	1.00	1.00	1.00	1.00
3	1.00	1.00	1.00	1.00

Table 22: Performance of Each Classes by ViT Model

Table:22 shows the performance by ViT model for four classes. Classes 0 and 1 both had an accuracy, recall, F1-score, and precision of 0.99, indicating nearly flawless recognition and prediction abilities. Classes 2 and 3 received perfect 1.00 scores in all metrics, demonstrating the model's exceptional ability to correctly identify and predict these categories with no errors. This indicates a highly effective model that consistently outperforms across multiple classifications.

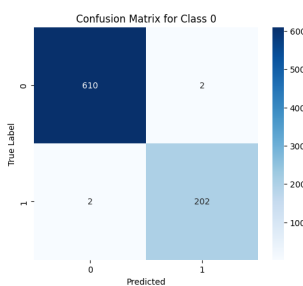


Figure 117: Confusion Matrix for Class 0 for ViT Model

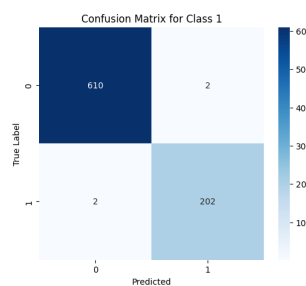


Figure 118: confusion matrix of class 1 for ViT model

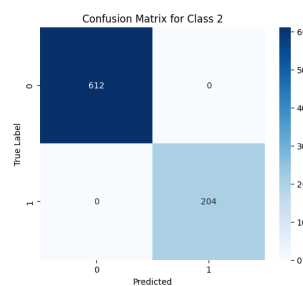


Figure 119: confusion matrix of class 2 for ViT model

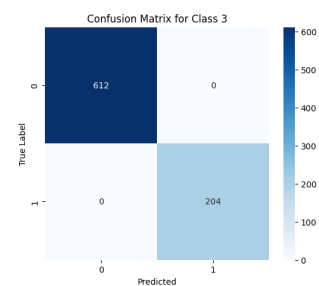


Figure 120: confusion matrix of class 3 for ViT model

### Confusion Matrix:

For ViT model, class 0 correctly identified 202 instances with a low number of misclassifications, demonstrating high precision and accuracy in figure:117. Similarly, Class 1 performs well, with 202 correct classifications and few errors, indicating that the model is reliable (see figure:118). Class 2 and Class 3 both show perfect identification, with 204 correct predictions each and no instances misclassified as other classes, demonstrating the model's exceptional ability to distinguish between these categories accurately in figure:119, and figure:120. These matrices demonstrate the model's efficacy across all test classes.

## A.2 Performance Evaluation of Private Data for Each Classes

In this section, we describe the performance matrix for each classes of textual and image data using private dataset.

### A.2.1 Performance evaluation of text data for each classes using RoBERTa model

Table:23 shows the performance by RoBERTa model for four classes individually. For Class 0, the model achieved an accuracy of 0.984, which was very high, demonstrating its ability to correctly identify this class. The recall was slightly lower (0.959), indicating that it captures the majority but not all relevant instances. The precision was quite high, at 0.977, indicating that when it predicts Class 0, it was usually correct. The F1-score, which balances precision and recall, was 0.968, indicating excellent overall performance in this class.

Class Label	Accuracy	Recall	F1-Score	Precision
0	0.984	0.959	0.968	0.977
1	0.986	0.981	0.970	0.961
2	0.995	0.990	0.991	0.991
3	0.998	0.997	0.997	0.997

Table 23: Performance of Each Classes by RoBERTa Model

### Confusion Matrix:

The matrix of RoBERTa model for class 0 (see figure:121) contains 772 true positives and 33 false negatives. There are 18 false positives. Class 1 contains 756 true positives and 15 false negatives. The false positive count is 31 (see figure:122). Class 2 shows 794 true positives with only eight false negatives and seven false positives in figure:123. Class 3 Like Class 2, it has an excellent prediction rate, with 795 true positives, 2 false negatives, and 2 false positives which is depicted in figure:124.

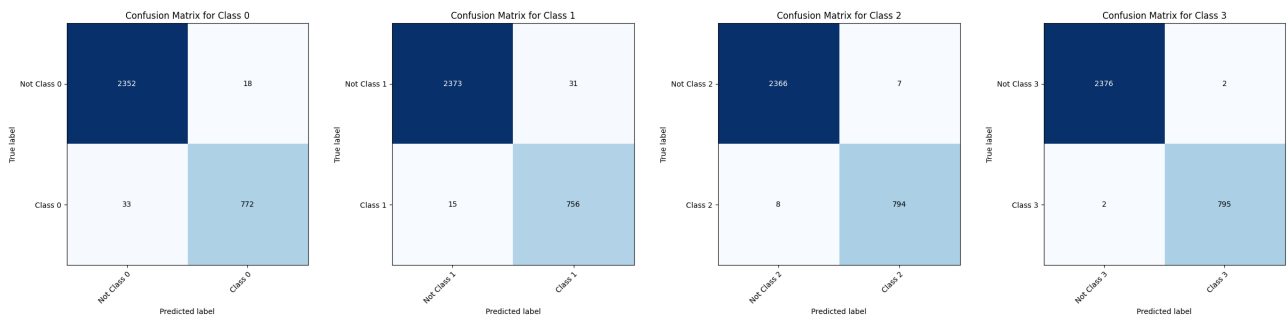


Figure 121: Confusion Matrix for Class 0 for RoBERTa Model of Private Data  
 Figure 122: confusion matrix of class 1 for RoBERTa model of Private Data  
 Figure 123: confusion matrix of class 2 for RoBERTa model of Private Data  
 Figure 124: confusion matrix of class 3 for RoBERTa model of Private Data

### A.2.2 Performance evaluation of text data for each classes using RoBERTa model

A performance for each classes by the ViT model using metrics like recall, F1-score, and precision with accuracy on private data is shown in table:24 for the four class labels (0, 1, 2, and 3) separately. For class 0, the model had an accuracy of 96.1%, an f1-score of 92.6% . Class 1 contained 94.2% accuracy, and class 2 also classified better, with an accuracy of 97.1%. Between four classes, class 3 stood out for nearly perfect performance metrics, with an accuracy of 99.0%.

Class Label	Accuracy	Recall	F1-Score	Precision
0	0.961	0.962	0.926	0.893
1	0.942	0.880	0.880	0.880
2	0.971	0.923	0.941	0.960
3	0.990	0.962	0.980	1.00

Table 24: Performance of Each Classes by ViT Model

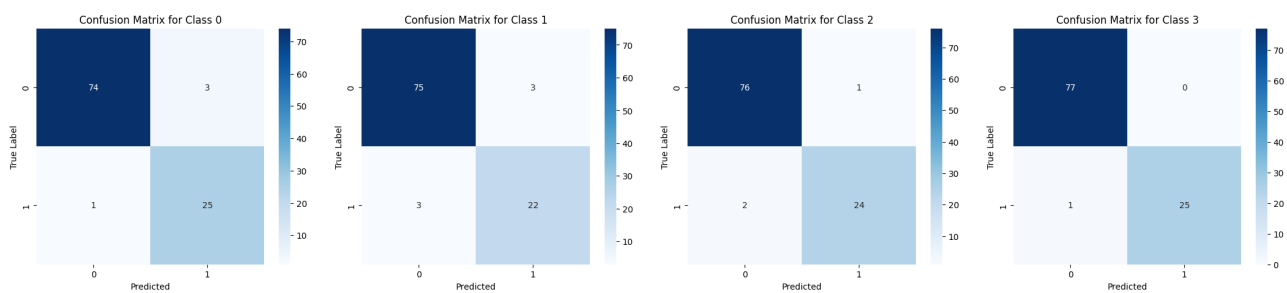


Figure 125: Confusion Matrix for Class 0 for ViT Model of Private Data  
 Figure 126: confusion matrix of class 1 for ViT model of Private Data  
 Figure 127: confusion matrix of class 2 for ViT model of Private Data  
 Figure 128: confusion matrix of class 3 for ViT model of Private Data

#### Confusion Matrix:

In ViT model, 26 instances in Class 0 (see figure:125) matrix were correctly identified, with one instance being misclassified another Class. Out of the 77 non-Class 0 instances, only 3 were mistakenly

classified as Class 0. In Class 1 Matrix (see figure:127), Of the 25 cases, 22 were correctly classified, and 3 cases were incorrectly classified with another class. Class 2 matrix Showed excellent accuracy, with 24 of 26 correct predictions; however, two cases were mislabeled with another classes (see figure:127). Perfect performance for Class 3 instances as shown in figure:128, with all 25 correctly identified, and none of the non-Class 3 instances (77 total) were incorrectly labeled as Class 3.

### **A.3 Source Code to Replicate The Experiment**

The Source code of our experiment can be accessed by following link:

- GitHub:
  - Public Dataset: <https://github.com/israt-tabassum/cyberbullying-classification-public-data>
  - Private Dataset: <https://github.com/israt-tabassum/cyberbullying-classification-private-data>
  - GUI: <https://github.com/israt-tabassum/cyberbullying-classification-website>