

YOUTUBE SPAM DETECTION USING ENSEMBLE METHOD



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

YOUTUBE SPAM DETECTION USING ENSEMBLE METHOD

SYAZA LIYANA BINTI MUHAMAD SHAPEE



This report is submitted in partial fulfillment of the requirements for the Bachelor of Computer Science (Computer Network) with Honours.

اويؤر ستي بيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA
2021**

DECLARATION

I hereby declare that this project report entitled
YOUTUBE SPAM DETECTION USING ENSEMBLE METHOD
is written by me and is my own effort and that no part has been plagiarized
without citations.

STUDENT:


SYAZA LIYANA BINTI MUHAMAD SHAPEE

Date : 7 SEPTEMBER 2021



I hereby declare that I have read this project report and found
this project report is sufficient in term of the scope and quality for the award of
Bachelor of Computer Science (Computer Network) with Honours.

SUPERVISOR :


MR. NOR AZMAN BIN MAT ARIFF

Date :

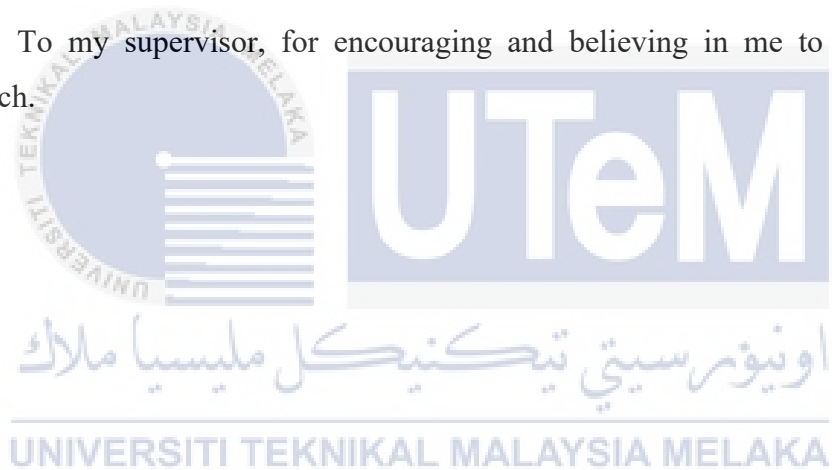
8/9/2021

DEDICATION

To my beloved parents, Zaleha binti Ali and Muhamad Shapee bin Ghazali who inspired me to be strong despite of many obstacles in life, for their prayers and their overwhelming support morally and financially. My sisters and brother, Syaza Hazirah, Syaza Najihah and Muhammad Adam have never left my side and are very special.

To my fellow friends, for being there for me throughout the entire bachelor program and their cooperation while conducting the research.

To my supervisor, for encouraging and believing in me to complete this research.



ACKNOWLEDGEMENTS

All praises be to Almighty Allah S.W.T who has blessed me with the belief, strength and capabilities to understand, learn and complete this research. Peace and prayers be upon our most beloved Prophet Muhammad S.A.W, the most beautiful soul, whose sayings, actions and stories have deeply inspired me enough to believe that there are no limitations to what I can achieve when we are fully committed to accomplish something, knowing that Allah is on my side.

I also admire the help and the guidance of my supervisor, Mr. Nor Azman bin Mat Ariff for his guidance, encouragement and patience from all aspects during the preparation of this research are highly appreciated.

I am blessed to have had such wonderful, loving and supporting parents, Zaleha binti Ali and Muhammad Shapee bin Ghazali for the education they gave me at home as I was growing up and the education, they paid for till I graduated. They have been my pillar of strength and till this very day, every small achievement I make, they always want to be the first to know and to congratulate.

To my fellow friends who have helped in the strenuous process in collecting information and preparing this research. May Allah bless you all for your patience and selfless commitment.

ABSTRACT

The number of YouTube users is constantly rising. However, such success is not without its drawbacks. Spam has become a common form of attack and threat, and most YouTube users are unaware of it. Receiving and being overwhelmed with unnecessary spam regularly has become one of the most internet-disruptive topics in today's world. The Support Vector Machine (SVM) is used in this study to develop a YouTube detection framework. The YouTube spam datasets were obtained from the UCI Machine Learning Repository. This project aims to show that an SVM model can accurately predict YouTube spam in a comment. Based on the SVM model, this research could produce a system that can detect spam and legitimate comments on YouTube.

ABSTRAK

Bilangan pengguna YouTube terus meningkat. Namun, kejayaan itu bukan tanpa kekurangannya. Spam telah menjadi bentuk serangan dan ancaman yang biasa, dan kebanyakan pengguna YouTube tidak menyedarinya. Menerima dan dibanjiri dengan spam yang tidak perlu secara berkala telah menjadi salah satu topik yang mengganggu internet di dunia sekarang. Mesin Vektor Sokongan (SVM) digunakan dalam kajian ini untuk mengembangkan kerangka pengesanan YouTube. Set data spam YouTube diperoleh dari UCI Machine Learning Repository. Projek ini bertujuan untuk menunjukkan bahawa model SVM dapat meramalkan spam YouTube dengan tepat dalam komen. Berdasarkan model SVM, penyelidikan ini dapat menghasilkan sistem yang dapat mengesan spam dan komen yang sah di YouTube.

اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

TABLE OF CONTENTS

	PAGE
DECLARATION	II
DEDICATION	III
ACKNOWLEDGEMENTS	IV
ABSTRACT	V
ABSTRAK	VI
TABLE OF CONTENTS	VII
LIST OF TABLES	XVI
LIST OF FIGURES	1
LIST OF ABBREVIATIONS	5
LIST OF ATTACHMENTS	6
CHAPTER 1: INTRODUCTION	7
1.1 Introduction	7
1.2 Problem Statement (PS)	8
1.3 Project Questions (PQ)	9
1.4 Project Objective (PO)	9
1.5 Project Scope	10
1.6 Project Contribution	10
1.7 Report Organization	11
1.7.1 Chapter I: Introduction	11

1.7.2	Chapter II: Literature Review	11
1.7.3	Chapter III: Project Methodology	11
1.7.4	Chapter IV: Analysis and Design	11
1.7.5	Chapter V: Implementation	11
1.7.6	Chapter VI: Discussion	11
1.7.7	Chapter VII: Project Conclusion	12
1.8	Conclusion	12
CHAPTER 2: LITERATURE REVIEW		13
2.1	Introduction	13
2.2	General Categories of Internet Security Attack (ISA)	15
2.2.1	ISA Definition	15
2.2.2	Denial of Service Attack (DoS)	15
2.2.3	Phishing Attack	15
2.2.3.1	Clone Phishing	16
2.2.3.2	Spear Phishing	16
2.2.3.3	DNS Base Phishing	16
2.2.4	Spam Attack	17
2.2.5	Malware	17
2.2.6	Virus	17
2.3	Classification of ISA	18
2.3.1	Active Attack	18
2.3.2	Passive Attack	18
2.4	Spam	19
2.4.1	Spam Definition	19
2.4.2	Spam Type	19

2.4.2.1	E-mail	19
2.4.2.2	Web Spam	20
2.4.2.3	Short Message Service (SMS) Spam	21
2.4.2.4	Image Spam	21
2.4.2.5	YouTube Spam	22
2.4.3	Spam Detection Technique	23
2.4.4	Spam Analysis Technique	24
2.5	Machine Learning	25
2.5.1	Machine Learning Definition	25
2.5.2	Dataset	26
2.5.3	Data Preprocessing	26
2.5.3.1	Definition	26
2.5.3.2	Preprocessing Type	26
2.5.4	Feature Extraction	27
2.5.4.1	N-Gram	28
2.5.4.2	Lexical Features	28
2.5.4.3	Ensemble Method	29
2.5.5	Data Splitting and Validation	32
2.5.5.1	Random Subsampling	32
2.5.5.2	Cross-validation	33
2.5.5.3	Bootstrapping	33
2.5.6	Feature Selection	33
2.5.6.1	Feature Selection Definitions	33

2.5.6.2	Feature Selection Type	34
2.5.7	Classification	36
2.5.8	Classification Type	36
2.5.8.1	Generative	36
2.5.8.2	Discriminative	38
2.6	Critical Review	39
2.6.1	Previous Research on Spam	39
2.6.1.1	Research Paper I	39
2.6.1.2	Research Paper II	39
2.6.1.3	Research Paper III	40
2.6.2	Previous Research on YouTube Spam	41
2.6.2.1	Research Paper I	41
2.6.2.2	Research Paper II	41
2.6.2.3	Research Paper III	42
2.6.2.4	Research Paper IV	42
2.7	Conclusion	44
CHAPTER 3: PROJECT METHODOLOGY		45
3.1	Introduction	45
3.2	Methodology	45
3.2.1	Previous Research	46
3.2.2	Information Gathering	47
3.2.3	Define Scope	47
3.2.4	Design and Implementation	47

3.2.5	Testing and Evaluation of Model	47
3.2.6	Documentation	47
3.3	Project Schedule and Milestones	48
3.3.1	Project Flowchart	48
3.3.2	Project Milestones	49
3.3.3	Project Gantt Chart	50
3.4	Requirement Analysis	50
3.4.1	Software Requirement	50
3.4.2	Hardware Requirement	51
3.5	Conclusion	52
CHAPTER 4: ANALYSIS AND DESIGN		53
4.1	Introduction	53
4.2	Problem Analysis	53
4.3	Project Design	53
4.3.1	Dataset	54
4.3.2	Data Preprocessing	55
4.3.3	Feature Extraction	58
4.3.4	Generate Bag of Word (BOW) Feature Vector	59
4.3.5	Data Splitting and Validation	59
4.3.6	Feature Selection	60
4.3.7	Normalization	62
4.3.8	Classification	62
4.3.9	Post-Classification	66
4.4	Conclusion	66
CHAPTER 5: IMPLEMENTATION		67

5.1	Introduction	67
5.2	Software Development Environment Setup	67
5.3	Process Module	68
5.3.1	Collection of Dataset	68
5.3.2	Data Preprocessing	68
5.3.2.1	Eminem Dataset Preprocessing	68
5.3.2.2	Psy Dataset Preprocessing	72
5.3.2.3	Shakira Dataset Preprocessing	76
5.3.2.4	LMFAO Dataset Preprocessing	79
5.3.2.5	Katy Perry Dataset Preprocessing	83
5.3.3	Feature Extraction	86
5.3.3.1	Eminem Dataset Feature Extraction	87
5.3.3.2	Psy Dataset Feature Extraction	89
5.3.3.3	Shakira Dataset Feature Extraction	90
5.3.3.4	LMFAO Dataset Feature Extraction	92
5.3.3.5	Katy Perry Dataset Feature Extraction	94
5.3.4	Feature Vector	95
5.3.4.1	Dataset Eminem Feature Vector	96
5.3.4.2	Dataset Psy Feature Vector	99
5.3.4.3	Dataset Shakira Feature Vector	102
5.3.4.4	Dataset LMFAO Feature Vector	106
5.3.4.5	Dataset Katy Perry Feature Vector	109

5.3.5	Data Training and Testing	113
5.3.5.1	Splitting Data	113
5.3.5.2	Model File	115
5.3.5.3	Predict File	116
5.4	Script Execution	116
5.4.1	Eminem Dataset Scripts	116
5.4.2	Psy Dataset Scripts	118
5.4.3	Shakira Dataset Scripts	119
5.4.4	LMFAO Dataset Scripts	120
5.4.5	Katy Perry Dataset Scripts	121
5.5	Result	122
5.5.1	Eminem Dataset Result	123
5.5.1.1	Attribute	123
5.5.1.2	10 Runs	123
5.5.1.3	Accuracy Table	125
5.5.2	Psy Dataset Result	126
5.5.2.1	Attribute	126
5.5.2.2	10 Runs	127
5.5.2.3	Accuracy Table	128
5.5.3	Shakira Dataset Result	130
5.5.3.1	Attribute	130
5.5.3.2	10 Runs	131
5.5.3.3	Accuracy Table	132

5.5.4	LMFAO Dataset Result	133
5.5.4.1	Attribute	133
5.5.4.2	10 Runs	134
5.5.4.3	Accuracy Table	135
5.5.5	Katy Perry Dataset Result	137
5.5.5.1	Attribute	137
5.5.5.2	10 Runs	138
5.5.5.3	Accuracy Table	139
5.5.6	Summary Dataset Result	140
5.6	Conclusion	141
CHAPTER 6: DISCUSSION		142
6.1	Introduction	142
6.2	Discussion of Project	142
6.3	Discussion on The Newly Proposed Method	143
6.4	Conclusion	144
CHAPTER 7: PROJECT CONCLUSION		145
7.1	Introduction	145
7.2	Project Summary	145
7.3	Project Constraint	146
7.4	Project Contribution	146
7.5	Project Limitation	146
7.6	Future Work	146

7.7	Conclusion	146
	REFERENCES	147
	APPENDIX	154



LIST OF TABLES

Table 1.1: Problem Statement	8
Table 1.2 Summary of Project Question	9
Table 1.3 Summary of Project Objective	9
Table 1.4: Project Contribution	10
Table 2.1: SPAM Literature	40
Table 2.2: YouTube SPAM literature	42
Table 3.1 Project Milestone	49
Table 3.2: Software Requirement for the Project	51
Table 3.3: Hardware Requirement of the Project	51
Table 4.1 Description of Dataset	55
Table 4.2: Advantage and Disadvantage of Rules	58
Table 4.3: Type of SVM Kernel	65
Table 5.1: Generate Ngram Eminem	87
Table 5.2: Feature Descriptor Eminem	88
Table 5.3: Parameter Details of Generate Ngram Psy	89
Table 5.4: Feature Descriptor Psy	90
Table 5.5: Generate Ngram Shakira	91
Table 5.6: Feature Descriptor Shakira	92
Table 5.7: Parameter Details of Generate Ngram LMFAO	92
Table 5.8: Feature Descriptor LMFAO	93
Table 5.9: Generate Ngram Katy Perry	94
Table 5.10: Feature Descriptor Katy Perry	95
Table 5.11: Details Parameter of Feature Vector Eminem	96
Table 5.12: Generate Feature Vector Eminem for Both Experiments	97
Table 5.13: Details Parameter of Convert CSV to ARFF Eminem	98
Table 5.14: Details Parameter of Feature Vector Psy	99

Table 5.15: Generate Feature Vector Psy for Both Experiments	100
Table 5.16: Details Parameter of Convert CSV to ARFF Psy	102
Table 5.17: Details Parameter of Feature Vector Shakira	103
Table 5.18: Generate Feature Vector Shakira for Both Experiments	104
Table 5.19: Details Parameter of Convert CSV to ARFF Shakira	105
Table 5.20: Details Parameter of Feature Vector LMFAO	106
Table 5.21: Generate Feature Vector LMFAO for Both Experiments	107
Table 5.22: Details Parameter of Convert CSV to ARFF LMFAO	109
Table 5.23: Details Parameter of Feature Vector Katy Perry	110
Table 5.24: Generate Feature Vector Katy Perry for Both Experiments	111
Table 5.25: Details Parameter of Convert CSV to ARFF Katy Perry	112
Table 5.26: Sorted Number of Instances of Each Dataset	113
Table 5.27: Training and Testing Size	114
Table 5.28: Parameter description Eminem	117
Table 5.29: Parameter description Psy	118
Table 5.30: Parameter description Shakira for experiment 1	119
Table 5.31: Parameter description LMFAO for experiment 1	121
Table 5.32: Parameter description Katy Perry for experiment 1	122
Table 5.33: 80% of Total Instances Attributes Eminem	123
Table 5.34: Single Classifier of Eminem Accuracy	125
Table 5.35: 80% of Total Instances Attributes Psy	127
Table 5.36: Single Classifier of Psy Accuracy	128
Table 5.37: 80% of Total Instances Attributes Shakira	130
Table 5.38: Single Classifier of Shakira Accuracy	132
Table 5.39: 80% of Total Instances Attributes LMFAO	134
Table 5.40: Single Classifier of LMFAO Accuracy	135
Table 5.41: 80% of Total Instances Attributes Katy Perry	137
Table 5.42: Single Classifier of Katy Perry Accuracy	139
Table 5.43: Dataset Summarization	140
Table 6.1: The Benchmark	143
Table 6.2: Final Result	144

LIST OF FIGURES

Figure 2.1: Literature Review's Structure	14
Figure 2.2: Type of Spam	19
Figure 2.3: Example E-mail Spam	20
Figure 2.4: Example of SMS Spam	21
Figure 2.5: Example spam comment on YouTube	22
Figure 2.6: Illustration of Bagging Technique in Ensemble Method	30
Figure 2.7: Illustration of Boosting Technique in Ensemble Method	31
Figure 2.8: Wrapper Methods	35
Figure 2.9: Filter Methods	35
Figure 3.1: Framework of the System	46
Figure 3.2: Project Flowchart	48
Figure 4.1: Project Design	54
Figure 4.2: Example of Psy dataset	55
Figure 4.3: Example of features will be use	55
Figure 4.4: Raw Data	56
Figure 4.5: Tokenization	56
Figure 4.6: Token Length	56
Figure 4.7: Case Normalization	57
Figure 4.8: Special Character	57
Figure 4.9: Stop Word Removal	57
Figure 4.10: Stemming	57
Figure 4.11: Subsampling	59
Figure 4.12: SVM Graph	62
Figure 4.13: Hyperplane Placement	63
Figure 4.14: Real Distribution Data Example	64
Figure 4.15: Non-Linear Graph	64
Figure 4.16: SVM with Multidimensional Space	65
Figure 5.1: Process Module	68

Figure 5.2: Data Preprocessing Module	68
Figure 5.3: Sort Dataset Eminem	69
Figure 5.4: Content Dataset Eminem	69
Figure 5.5: Remove Non-ASCII Character	70
Figure 5.6: No Author	70
Figure 5.7: No Date	71
Figure 5.8: Convert To Lowercase	71
Figure 5.9: Sample scriptEminem.bat	72
Figure 5.10: Sample File Created for Eminem	72
Figure 5.11: Sort Dataset Psy	73
Figure 5.12: Content Psy Dataset	73
Figure 5.13: Remove Non-ASCII Character	74
Figure 5.14: No Author	74
Figure 5.15: Convert To Lowercase	75
Figure 5.16: Sample scriptPsy.bat	75
Figure 5.17: Sample File Created for Psy	76
Figure 5.18: Sort Dataset Shakira	76
Figure 5.19: Content Shakira Dataset	77
Figure 5.20: Remove Non-ASCII Character	77
Figure 5.21: No Author	78
Figure 5.22: Convert To Lowercase	78
Figure 5.23: Sample scriptShakira.bat	79
Figure 5.24: Sample File Created for Shakira	79
Figure 5.25: Sort LMFAO Dataset	80
Figure 5.26: Content LMFAO Dataset	80
Figure 5.27: Remove Non-ASCII Character	81
Figure 5.28: No Author	81
Figure 5.29: Convert To Lowercase	82
Figure 5.30: Sample scriptLMFAO.bat	83
Figure 5.31: Sample File Created for LMFAO	83
Figure 5.32: Sort Katy Perry Dataset	84
Figure 5.33: Content Katy Perry Dataset	84
Figure 5.34: Remove Non-ASCII Character	84
Figure 5.35: No Author	85

Figure 5.36: Convert To Lowercase	85
Figure 5.37: Sample scriptKatyPerry.bat	86
Figure 5.38: Sample File Created for Katy Perry	86
Figure 5.39: Generate Ngram Eminem	87
Figure 5.40: Batch File Eminem	88
Figure 5.41: Generate Ngram Psy	89
Figure 5.42: Batch File Psy	90
Figure 5.43: Generate Ngram Shakira	91
Figure 5.44: Batch File Shakira	91
Figure 5.45: Generate Ngram LMFAO	92
Figure 5.46: Batch File LMFAO	93
Figure 5.47: Generate Ngram Katy Perry	94
Figure 5.48: Batch File Katy Perry	95
Figure 5.49: Feature Vector Module	95
Figure 5.50: Batch File for Ngram Feature Vector Eminem	96
Figure 5.51: Batch File Convert CSV to ARFF Eminem	98
Figure 5.52: Batch File for Ngram Feature Vector Psy	99
Figure 5.53: Batch File Convert CSV to ARFF Psy	102
Figure 5.54: Batch File for Ngram Feature Vector Shakira	103
Figure 5.55: Batch File Convert CSV to ARFF Shakira	105
Figure 5.56: Batch File for Ngram Feature Vector LMFAO	106
Figure 5.57: Batch File Convert CSV to ARFF LMFAO	109
Figure 5.58: Batch File for Ngram Feature Vector Katy Perry	110
Figure 5.59: Batch File Convert CSV to ARFF Katy Perry	112
Figure 5.60: Train and Test Module	113
Figure 5.61: tern and Structure Training and Testing Data	115
Figure 5.62: Train Process of Model and Scale	116
Figure 5.63: Process Generating Predict File	116
Figure 5.64: Run code program Eminem experiment 1	117
Figure 5.65: Run code program Eminem experiment 2	117
Figure 5.66: Run code program Psy experiment 1	118
Figure 5.67: Run code program Psy experiment 2	118
Figure 5.68: Run code program Shakira experiment 1	119
Figure 5.69: Run code program Shakira experiment 2	119

Figure 5.70: Run code program LMFAO experiment 1	120
Figure 5.71: Run code program LMFAO experiment 2	120
Figure 5.72: Run code program Katy Perry experiment 1	121
Figure 5.73: Run code program Katy Perry experiment 2	122
Figure 5.74: Example of 10 Runs Experiment for 1 gram by Eminem	124
Figure 5.75: Graph of Experiment 1 Eminem	124
Figure 5.76: Multiple Classifier of Eminem Accuracy	126
Figure 5.77: Example of 10 Runs Experiment for 1 gram by Psy	128
Figure 5.78: Graph of Experiment 1 Psy	128
Figure 5.79: Multiple Classifier of Psy Accuracy	129
Figure 5.80: Example of 10 Runs Experiment for 1 gram by Shakira	131
Figure 5.81: Graph of Experiment 1 Shakira	131
Figure 5.82: Multiple Classifier of Shakira Accuracy	133
Figure 5.83: Example of 10 Runs Experiment for 1 gram by LMFAO	135
Figure 5.84: Graph of Experiment 1 LMFAO	135
Figure 5.85: Multiple Classifier of LMFAO Accuracy	136
Figure 5.86: Example of 10 Runs Experiment for 1 gram by Katy Perry	138
Figure 5.87: Graph of Experiment 1 Katy Perry	139
Figure 5.88: Multiple Classifier of Katy Perry Accuracy	140

اوتنور سیتی بیکنیکل ملیسیا ملاک

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF ABBREVIATIONS

ASCII	-	American Standard Code For Information Interchange
BOW	-	Bag-Of-Word
CS	-	Chi-Square
DNS	-	Domain Name System
DOS	-	Denial Of Service Attack
Email	-	Electronic Mail
FN	-	False Negatives
FP	-	False Positives
FS	-	Feature Selection
FYP	-	Final Year Project
GIF	-	Graphic Interchange Format
HTML	-	Hypertext Markup Language
IG	-	Information Gain
IDE	-	Integrated Drive Electronics
Weka	-	Waikato Environment for Knowledge Analysis
SVM	-	Support Vector Machine

LIST OF ATTACHMENTS

Appendix A- Gantt Chart PSM 1	154
Appendix B- Gantt Chart PSM 2	156



CHAPTER 1: INTRODUCTION

1.1 Introduction

After Google, YouTube is the second most popular search engine users use to communicate through the medium. YouTube does not consider time or distance of interaction to be limitations, and users can upload more than 100 hours of video footage. Because YouTube allows people to easily upload and store videos online, it is often used for finding information and discovering videos on related topics. YouTube allows people to easily upload and store videos online, it is also often used for finding information and discovering videos on related topics. Since YouTube is one of the most popular ways to reach a wide audience, it is subject to a variety of attacks and threats, one of the most serious of which is spamming.

Nowadays, we see that it is very common for users to receive and become flooded with unnecessary spamming daily, causing a slew of technological and personal issues for users. Spam comment may be defined as a comment that is irrelevant to the web page's particular content. Comment spams have been used to publish unusual unwanted material, advertise sales, advertise pornographic content, degrade the website's reputation, and raise the number of views on the website (Ali & Amin, 2016). This research aims to counter YouTube comments spam and prevent these problems by using machine learning method with better accuracy.

1.2 Problem Statement (PS)

The number of YouTube users is constantly rising. However, such success is not without its drawbacks. Spam has become a common form of attack and threat, and most YouTube users are unaware of it. Spam is classified as harmful because it causes a cyber security risk to end users. Spam comment may be defined as a comment that is irrelevant to the web page's particular content. Comment spams have been used to publish unusual unwanted material, advertise sales, advertise pornographic content, degrade the website's reputation, and raise the number of views on the website (Ali & Amin, 2016). Hence, the spammer took advantage of the opportunity to spread malware via comment fields, exploiting vulnerabilities in the users' computer (Aziz et al., 2018). Spam detection on YouTube is considered as one of the most effective solutions to the issue. The problem was defined as in table 1.1.

Table 1.A: Problem Statement

PS	Problem Statement
PS1	Spam has become a common form of attack and threat, and most YouTube users are unaware of it. Predicting and recognizing the comments are genuine, legitimate, and which are spam becomes more difficult. Another issue with Spam YouTube is that it is very easy for malicious and irrelevant content to spread via their propagation.

1.3 Project Questions (PQ)

Before this study, three project questions (PQ) needed to be answered. The summarization of the project question based on the problem statement as seen in Table 1.2.

Table 1.B Summary of Project Question

PS	PQ	Project Question
PS1	PQ1	What exactly is a spam comment on YouTube in the context of social media?
	PQ2	What criteria will be used to classify spam comments on YouTube?
	PQ3	Is the machine learning approach better at detecting between real and spam comments on YouTube?

1.4 Project Objective (PO)

The project objective defines an expected outcome for this research. The problem statement (PS) and project question (PQ) must be addressed to achieve the expected outcomes. There are three objectives to accomplish for this project and the relationship of PS, PQ, and PO in this study, as shown in table 1.3.

Table 1.C Summary of Project Objective

PS	PQ	PO	Project Objective
PS1	PQ1	PO1	To study the behavior of YouTube spams.
	PQ2	PO2	To develop an ensemble method that can be used to classify YouTube spam.
	PQ3	PO3	To test and verify the performance of the proposed ensemble method.

1.5 Project Scope

The scope of this project (PS) is limited to the following criteria as follow:

- I. Datasets were collected from the UCI Machine Learning Repository
- II. Performance of the proposed model is measured using accuracy.
- III. Focus on YouTube spam only and only the process of content in a comment.

1.6 Project Contribution

The aim of this project contribution (PC) is to test spam comments in YouTube detection by implementing the project and selecting the best features to improve prediction accuracy and validate using various YouTube datasets. Table 1.4 gives an outline of this contribution to the project.

Table 1.D: Project Contribution

PS	PQ	PO	PC	Project Contribution
PS1	PQ1	PO1	PC1	The most accurate output pattern on YouTube Spam has been detected.
	PQ2	PO2	PC2	The behavior of the Machine Learning was identified when various sizes of data were used.
	PQ3	PO3	PC3	The various datasets of YouTube are validated.

1.7 Report Organization

1.7.1 Chapter I: Introduction

This chapter focus as the reference for outlining the most important details before starting the project. To ensure that this project is clearly understood, it explained the problem statement, project objective, project scope, expected outcome, and conclusion.

1.7.2 Chapter II: Literature Review

This chapter covers the previous researcher's related findings, who are later evaluated to identify the differences between them, resulting in the contributions of this project.

1.7.3 Chapter III: Project Methodology

This chapter discusses on which methodology should be used to evaluate the project's system and to provide a rigid set of works that needed to be completed in order to meet the project objectives.

1.7.4 Chapter IV: Analysis and Design

This chapter focus on analysing and designing procedures that relate to this project evaluation.

1.7.5 Chapter V: Implementation

This chapter will define the procedures to obtain a precision result. The results of this project will be collected to demonstrate that it is complete and the results will be recorded to create an estimate, which will then be compared to other methods.

1.7.6 Chapter VI: Discussion

This chapter examines the findings and analyses them to determine if the objectives were achieved.

1.7.7 Chapter VII: Project Conclusion

This chapter will outline the project, describe the contribution, and highlight the project's constraints. This chapter also discuss what should be done to develop the project in the future.

1.8 Conclusion

This study aims to find out and develop knowledge about spam comment attacks in order to enhance the method of predicting and recognize which comments are genuine, legitimate and which are spam. A Machine Learning algorithm is used to construct a model. This research should be able to detect and deter spam comment attacks on YouTube, as well as monitor events using machine learning techniques. The following chapter digs deeper into the related works or literature focused on internet security attacks and machine learning.



CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This chapter describes and discusses the related work and previous research on internet security attacks using classification techniques, as shown in Figure 2.1. this literature review would describe in detail by explaining categories of internet security attacks, classification of internet security attacks, spam, machine learning, and collection of features and classification used by the previous researcher. By the conclusion of this chapter, it will give a clear understanding of spam and be more tolerant of it.

اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

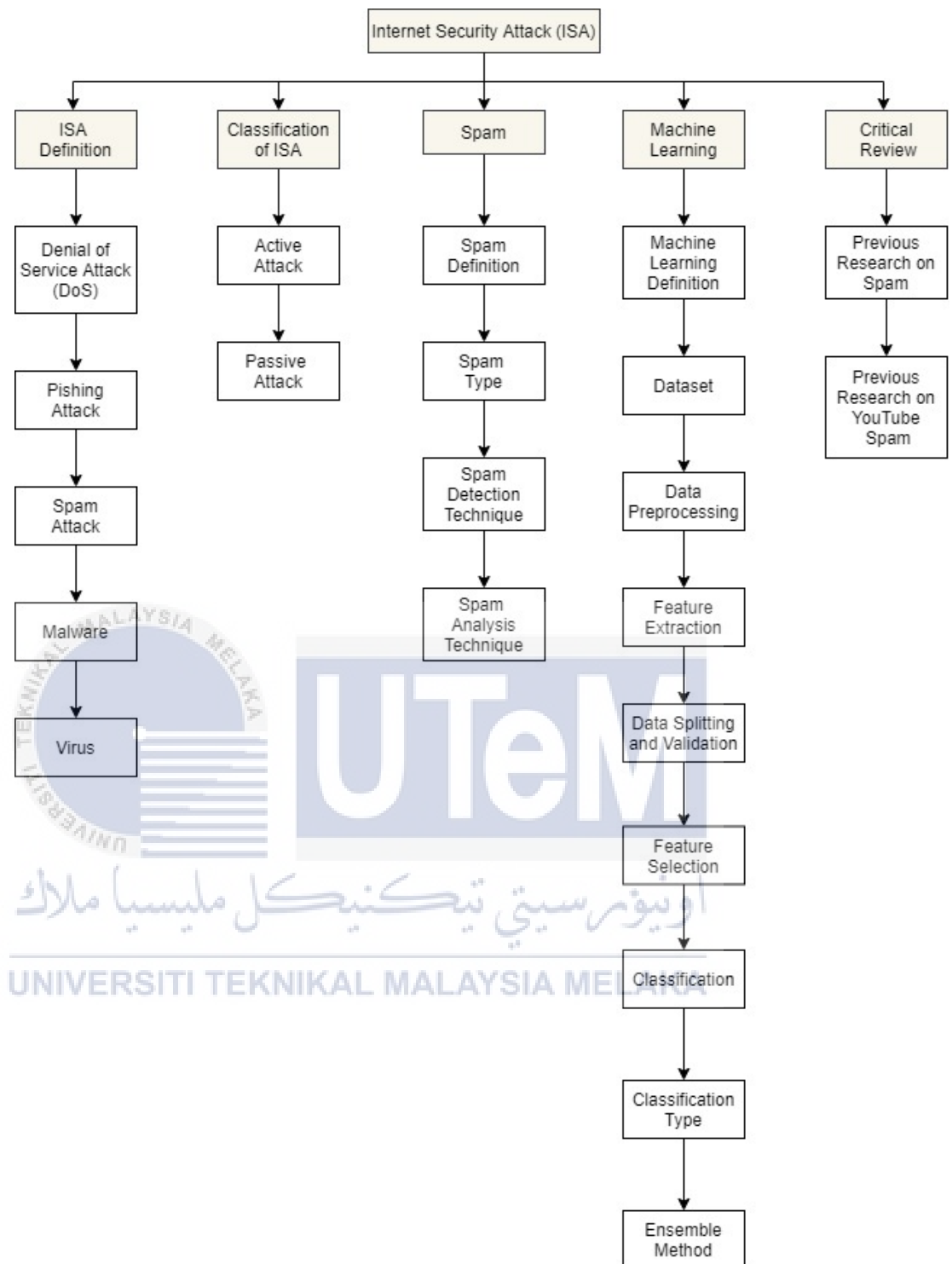


Figure 2.1: Literature Review's Structure

2.2 General Categories of Internet Security Attack (ISA)

2.2.1 ISA Definition

Internet is a worldwide network that connects computers that do not limit the information of every user. People can exchange information and communicate from any location through an Internet connection. A malicious attack happened from the internet's rapid growth. Attacks on the network are aimed at stealing, modifying, harming, or acquiring unauthorized access to a site's content, for example (S.Mangrulkar et al., 2014).

2.2.2 Denial of Service Attack (DoS)

When legitimate users seem unable to access information systems, computers, or other network services, this is known as a denial of service (DoS) attack. As a result, it prohibits authorized users from accessing the network by making the network too busy or overload. A denial-of-service attack occurs by overwhelming the targeted host or network with traffic until it becomes unable to respond or fails, denying legitimate user access. If an organization's resources and services are unavailable, DoS attacks will cost them both time and money (Gupta & Sharma, 2020). Ping of Death, SSPing, Land, Win Nuke and SYN Flood are examples of DoS attacks (N. Ahmad & Habib, 2010).

2.2.3 Phishing Attack

Phishing is the act of trying to obtain personal information from a victim by impersonating a trustworthy third party in electronic communication, who may be an individual or a reputable company. A phishing attack aims to persuade recipients to reveal personal information such as bank account numbers, passwords, and credit card numbers (Damodaram, 2016). The attacker creates a copy of an actual Web page to trick users into sending personal, financial, or password data to what they believe is their service provider's Website, for example, by using specially configured e-mails or text messengers (Yahaya et al., 2017). When a user attempts to sign in with their account information, the attacker collects the user's login credentials and then uses the website information.

2.2.3.1 Clone Phishing

Clone phishing is a phishing attack in which an attacker attempts to clone a website that the user often uses. To imitate the actual websites, the clone website normally requests login credentials (Banu & Banu, 2013). Clone phishing would be that an attacker creates a virtually identical copy of a valid message to manipulate the victim into believing it is genuine. The email is sent from an address that seems to be that of the actual sender, and the message's body is identical to that of a previous message. Instead of support@youtube.com, the email address could be support@youtube.com. The attacker may convince the victim to send repeated messages by claiming that it is an updated version or that the previous message contained an error (Floderus & Rosenholm, 2019).

2.2.3.2 Spear Phishing

Sending emails to specific targets by pretending to a trustworthy sender is known as spear phishing. The aim is to infect computers with malware or persuade victims to share personal information or money (Floderus & Rosenholm, 2019). As a result, these attacks are extremely difficult to detect by users, raising security issues among internet users.

2.2.3.3 DNS Base Phishing

DNS-based phishing or known as Pharming is a form of attack in which a website's traffic is shifted to a fake domain. Pharming prevents domain names from being resolved to IP addresses, resulting in the domain name of a legitimate website being mapped to the IP address of a fake website. For example, if we type www.cimbclicks.com.my into the address bar, we will be redirected to www.google.com.my (Banu & Banu, 2013). The majority of phishing and pharming attacks target financial companies intending to steal card numbers and passwords from online bank customers.

2.2.4 Spam Attack

Spam attacks are where an app is used to send thousands of messages to its users in a structured and unauthorized manner. Fake or hacked accounts send these updates which, also contain fake ads and connections that actual users are prompted to click on. Spam is a form of spam mail that consists of an unsolicited commercial email address or junk mail that has been sent to several recipients and does not have a legitimate opt-out mechanism and the most nuisance that email users face today (Krishnamurthy, 2015).

2.2.5 Malware

Malware attacks are the most common network threat on the Internet today. Malware is any malicious software designed to carry out malicious actions on a computer device (Mokoena & Zuva, 2018). Also, malware describes programs that are designed to harm, damage, or infect machines, networks, and other services. Malware comes in a variety of forms, and they are divided into the following categories. Viruses, worms, trojan horses, rootkits, spyware, adware, cookies, sniffers, botnets, keyloggers, spam, and ransomware are only a few examples (Tahir, 2018). In most cases, malware is often developed by groups of hackers who are mostly interested in making money, either by distributing malware or selling it to the highest bidder on the Dark Web. However, malware is still bad news when it shows up on a computer or network, regardless of whether or how it came to be.

2.2.6 Virus

A computer virus is the most dangerous aspect of computers. This virus spreads through computers and networks by copying itself, typically without the user's knowledge. Virus attacks on computers are more dangerous than they seem, causing more damage to the device. It is important to understand the actions that a virus takes in one's environment, as well as the events that might occur (Vaidya, 2017). Viruses are spread as corrupted applications or documents are moved from one device to another through a network, a drive, file sharing methods, or infected e-mail attachments. To remain undetected by anti-virus devices, some viruses use various stealth techniques (Fund et al., 2021).

2.3 Classification of ISA

Network attacks can make network services slow, inaccessible for a short period, or unavailable for an extended period. Security attacks divide into two categories, passive attacks, and active attacks. When an active attack tries to change system resources or disrupt their service, it negatively impacts the network's integrity or availability. A passive attack tries to learn or use information from the system without affecting system capabilities, putting confidentiality at risk (S.Mangrulkar et al., 2014).

2.3.1 Active Attack

According to K. Ahmad et al. (2015), active attacks are those that the original message is modified or a false message is created. The attacker attempts to bypass or hack into networks that are designed to be secure. Viruses, Trojan horses, worms, and stealth could all be used to do this. These attacks are very complicated and difficult to detect. It can be categorized into three parts, interruption, fabrication, and modification. Active attacks result in data file disclosure or distribution, Denial of Service (DoS), or data modification (S.Mangrulkar et al., 2014).

2.3.2 Passive Attack

The attacker's goal in passive attacks is to collect data. They do not want to change the original message's content (K. Ahmad et al., 2015). Since it does not change the data, it is quite difficult to identify. Passive attacks use methods such as message releases, traffic monitoring, sniffing, and key loggers. Likewise, a passive attack examines unencrypted traffic for clear-text keys and personal data that may be used in other types of attacks. Without the user's permission or consent, a passive attack results in the disclosure of information or data files to an attacker (S.Mangrulkar et al., 2014).

2.4 Spam

2.4.1 Spam Definition

Spamming, or the act of sending unsolicited and irrelevant information, has been found in a variety of contexts, including email, instant messages, websites, and Internet telephony. Spam is the indiscriminate sending of unsolicited messages in bulk via electronic messaging systems (Hayati et al., 2010). More significantly, spam is defined by the consent of the recipient, not by the content. Spam is an issue that revolves around the recipient's consent to receive messages rather than the quality of the messages sent. As a result, spam has been the internet's nuisance (Krishnamurthy, 2015).

2.4.2 Spam Type

Figure 2.2 shows the several types of spam that are described in this section.

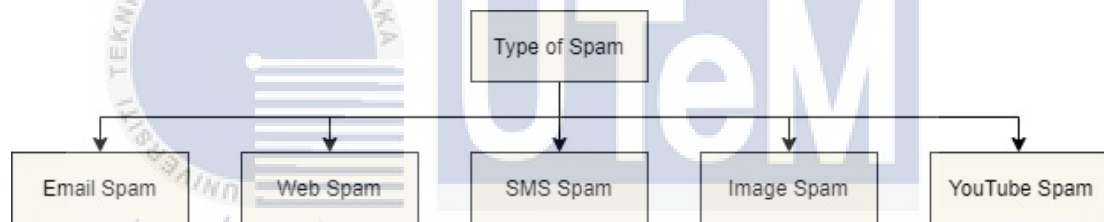


Figure 2.2: Type of Spam

2.4.2.1 E-mail

Apart from the telephone and other communication methods, email is the most widely used form of communication. Employees in a company are more alert to incoming emails than to telephone calls. As a practice, with tighter timelines and more actively trafficked schedules, email spam wastes time (Juneja & Pateriya, 2014). Spamming is one of the most serious risks to email which is some unsolicited beneficial communication.

Thus, it also slows down the email process by taking up server storage space as it is sent to several users from the same company (Sharma, 2018). Spam emails are spread by spammers for basic marketing reasons, but they may also be used to carry out more malicious acts such as financial instability and reputational harm on both a

personal and company. Spamming is also being used in a variety of other multimedia networking networks (Karim et al., 2019). An e-mail spam example shown in Figure 2.2.

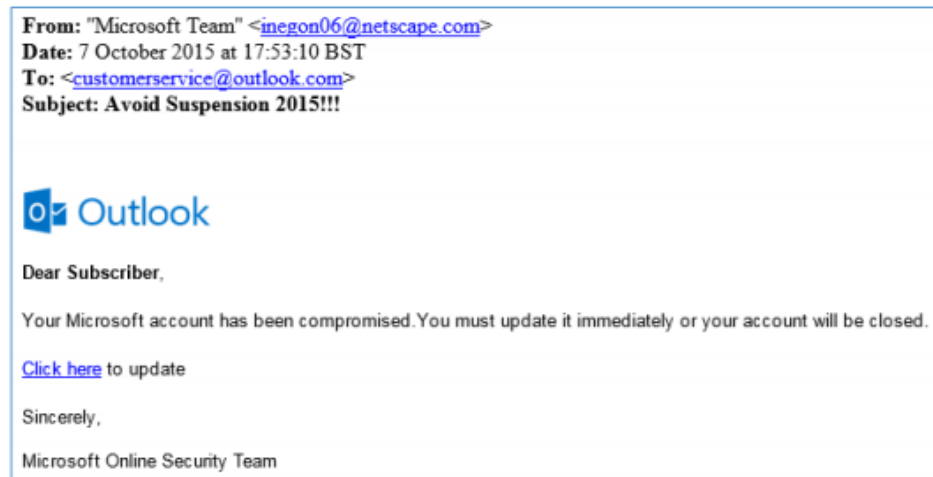


Figure 2.3: Example E-mail Spam

2.4.2.2 Web Spam

In today's modern world of ever-increasing knowledge on the internet, search engines are known as a platform for gaining access to it. Increasing the reliability of a site's pages is a legitimate way to improve its ranking in search engine rankings, but it is time-consuming and expensive. Another way is to use illegal and immoral tactics to boost the search engine ranking (Oskuie & Razavi, 2014).

Based on Najadat & Hmeidi (2008), web spam defines it as an effort to deceive and manipulate a search engine's relevancy algorithm or any action taken with the intent of manipulating a page's rating and ranking. In other words, spammers attempt to deceive search engines by increasing the rank of their web pages to attract user interest to their pages. Web spam reduces the accuracy of search results, wasting users' time in the process. When the number of these pages grows, so does the number of pages bots and indexers examine and sort. In this case, search engines' levels will be reduced, and the time spent searching in response to a user query rises (Oskuie & Razavi, 2014).

2.4.2.3 Short Message Service (SMS) Spam

Short Message Service (SMS) traffic continues to rise regularly. As a result, mobile attacks such as spammers that flood the service with spam messages delivered to groups of recipients increase. Spam defines it as unsolicited bulk messages in various formats, such as unnecessary ads, credit offers, or fake lottery winner notifications (Amir Sjarif et al., 2019). Because of its widespread, SMS spam has become a constant problem for mobile users. Mobile SMS spam bothers mobile phone owners, and, like e-mail spam, it creates extra societal frictions with mobile devices (Abdulhamid et al., 2017).



Figure 2.4: Example of SMS Spam

2.4.2.4 Image Spam

Spam can mostly be in text form, but it can also be in image form, such as advertising text in images attached to emails. Image spam is a form of spam or, more precisely, a spamming technique in which a spam message delivers as an image. It is a spam email that uses photos to bypass text to get through spam filters. It considers bypassing and avoiding spam filters that look for specific keywords. Based on Das and Prasad (2014), the purpose is to bypass spam filters that analyze emails solely on their textual content. The spam detection can misclassify such spam emails, but when the recipients open them, the hidden message becomes visible.

2.4.2.5 YouTube Spam

YouTube has grown to become one of the most popular social media platforms. The application is mainly built on video sharing. YouTube is one of the most popular information-gathering sites on the internet. As a result, many spammers will want to lure YouTube users by spamming the comments section. Since they include information that unrelated to the context of the posted video, such comments generally refer to as spam. Spam is often referred to by automated bots pretending to users. Irrelevant comments may be unintentional, such as malicious links, off-topic hate comments, and judgmental comments on controversial topics, or they can be intentional (Kavitha et al., 2020).

Spam creates a slew of issues, including wasting the user's time, memory, and network bandwidth. Thanks to the attack of spam, companies, and users may suffer financial losses. Many attackers advertise in the YouTube comment section, while others spread computer viruses, and some spam messages are designed to steal the user's financial information (Samsudin et al., 2019).



Figure 2.5: Example spam comment on YouTube

2.4.3 Spam Detection Technique

There are several techniques in use today that attempt to discourage or prevent the growth of massive amounts of spam or junk email. Spam filters are being used to pass the available techniques around. Spam detection techniques, also known as spam filters, examine various parts of an email message to specify if it is spam or not (Sharma, A., sha, M., Manisha, & Jain, D, 2014). The features of text-based sites are used by some spam techniques. Furthermore, users of such systems can simply learn to recognize, ignore, or stop text spams like URLs (Chowdury et al., 2013).

People can discover spam comments on YouTube using a variety of detection techniques. The technique uses to distinguish between spam and genuine comments. Spam detection employs a bag of words model, in which each message assigns to a specific number of terms. A bag-of-words model is a method of extracting text features modeling, such as machine learning algorithms. The model is efficient and adaptable, and it can be used to remove features from records in a variety of ways. Any information about the structure of words in the text is discarded, which that considered a bag of words. The model only cares whether recognized words appear in the document, not wherein the document they appear (Ellis-monaghan, 2006).

Spam detection techniques are divided into two categories which are origin-based technique and content-based spam detection technique. Origin-based spam detection is a technique for determining whether an email message is a spam or not based on network knowledge (Sharma, A., sha, M., Manisha, & Jain, D, 2014). The most significant pieces of network knowledge are the email address and the IP address. Blacklist, Whitelist, and Realtime Blackhole List (RBL) are the three parts of origin-based spam detection techniques. Blacklists are sets of email addresses or IP addresses that have been used to deliver spam in the past. When developing a filter, if the recipient of an email is on a blacklist, the email is considered inappropriate and would be classified as spam. The definition of a whitelist is the precise opposite of a blacklist. It consists of a list of authorized and penetrable entries. The messages classify as ham messages, and they may be acknowledged by the recipient. Then there is the real-time blackhole list (RBL) which also known as the internet blacklist list. Through previous violence occurring with just part of the package, users often assemble blacklists in

real-time on the internet and apply a whole range of IP addresses to the blacklist (Reddy & Reddy, 2019).

A spam detection technique that analyses content features such as word count, language models and content duplication is known as a content-based spam detection technique (Kamoru et al., 2017). Rule-based filters, Bayesian filters, Support Vector Machines (SVM), and Artificial Neural Networks (ANN) are some of the most common content-based spam detections. The Rule-Based Approach Filters use a set of guidelines based on the words in the whole message to determine if it is spam or not. To determine whether an email address is a spam or ham. A connection is created between each message and a set of guidelines. The most sophisticated method of content-based filtering is Bayesian filters, which use probability laws to determine the messages are valid and which are spam. Support Vector Machines (SVM) have proven to be effective in classifying text documents. SVMs are kernel methods which is the main concept is to integrate text documents into a vector space where geometry and linear algebra can be performed. In the vector space, SVMs attempt to construct a linear distinction between the two groups (Sharma, A., sha, M., Manisha, & Jain, D, 2014). At last, an artificial neural network (ANN) consists of a series of linked weight-related input or output units. The network learns to change weights during the learning process so that the input times will predict the right class label.

2.4.4 Spam Analysis Technique

The spam analysis technique used by the detection technique in image spam attempts to retrieve embedded text combination with visual attributes such as color, shape, and structure, and hence used to measure object resemblance. The color space distribution of the spam image tends to be rough RGB or LAB (Annadatha & Stamp, 2018). Gradient orientation histograms are used in image processing for image recognition in computer vision. The technique examines gradient orientation in specific areas of an image and may reveal text features (Attar et al., 2013).

Spam analysis technique, which uses different machine learning algorithms to detect email spam. Machine learning is a subset of supervised general learning and text classification in particular. Knowledge extraction and classification are two examples of open source text analysis tools (Chaudhari & Govilkar, 2015). The spam

is processed in many stages to decide whether the text is genuine or not. The raw data selection conducts data cleaning, such as cryptographic protocols, stop word removal, and stemming, during the pre-processing phase. According to Aziz et al.(2018), the data clean procedure is then being used to identify and remove features to produce vector and dictionary functions as input to the algorithm of choice. Then, in the classification process, text classification assigns predefined categories to text documents. We describe the test set after training and measure efficiency by comparing the predicted labels to the real labels. As claimed by Allahyari et al (2017), testing is carried out using accuracy, precision, recall, and F-measurement.

2.5 Machine Learning

2.5.1 Machine Learning Definition

By name, machine learning is a subset of computer science that is derived from artificial intelligence research through pattern recognition and computational learning theory (Paluszek & Thomas, 2019). Also, from a training set of completed or previous projects, a machine learning algorithm successfully learns how to estimate (Singh et al., 1999). Pattern analysis, anomaly detection, inference, and neural networks are examples of how a computer can be programmed to perform complex tasks using machine learning applications (R. K. Nath et al., 2020). Unsupervised and supervised machine learning are two kinds of machine learning techniques.

When given new data, supervised machine learning uses a pre-defined set of the training set to help it draw an appropriate conclusion and includes a feature derived from a specific collection of data (Paluszek & Thomas, 2019). The training data is made up of a series of examples that train the computer (A. Nath & Dasgupta, 2016). The model is trained using labelled and classified data. Unsupervised machine learning, on the other hand, is software that is given a set of data and is required to identify correlations and connections within it (Paluszek & Thomas, 2019). As stated in Nath & Dasgupta (2016), labeled instances are not available in unsupervised learning. The training was conducted on data that was either unlabeled or unclassified.

2.5.2 Dataset

Datasets are used in virtually every research area where evidence is used as the methodological basis for research (Renear et al., 2010). To ensure that a dataset can have a variety of pieces of data and use it to train an algorithm to identify predictable trends within the dataset as a whole. Also, to compare existing algorithms, a new algorithm is being used as a benchmark. Experiments aimed at overcoming the challenge of developing engineering training would look for current datasets. In this research, a dataset focusing on YouTube spam will be used, which comes from UCI's repository and has been cited by many scholars.

2.5.3 Data Preprocessing

2.5.3.1 Definition

Data preprocessing is an important step in Machine Learning because the accuracy of data and the valuable information that can be obtained from it has a significant impact on our model's capacity to learn. Thus, preprocessing our data before feeding it into our model is important. According to Agarwal (2015), the method of converting raw data into a usable format is known as data preprocessing. Data in the real world is often unreliable, inaccurate, redundant, and noisy. Data preprocessing includes several steps that help in the conversion of unprocessed data into a usable format.

2.5.3.2 Preprocessing Type

Preprocessing is an image processing method that has been used to improve and enhance the image quality of a specific application over the original image. Preprocessing is one of the first stages in ensuring that the processes that follow are very accurate. Due to many sounds identified in these objects, the raw images from the scanning center and the databases cannot be seen directly. It was pre-processed before testing. Conversion transfer, picture resizing reduction, noise reduction, and quality enhancement all contribute to a model that properly locates information (Perumal & Velmurugan, 2018). Gray Scale Conversion refers to noise-reduction filtering technique such as power spectrums and a fuzzy filter. Image noise is a random change in brightness or color information in created images, and image scaling is an important

aspect of image processing technology to improve and minimize image size in pixel format.

The number of words used to describe documents can be decreased through text preparation, filtering, lemmatization, and stemming. To reduce dictionary sizes and hence the dimension of the document categorization inside the collection, use filtering, lemmatizing, or stemming processes. Based on Dr.S.Kannan & Gurusamy (2015), To increase the performance of the IR system, data preprocessing techniques lower the quantity of the dataset. Stopping word processing entails removing phrases that give little or no information about writing, such as posts, conjunctures, and prepositions. Stop words rarely add to the context or substance of a textual document. A common term like and, are, this, and others are removed from the list of stopped words. They are ineffective for document categorization. However, according to Gharatkar et al. (2018), this stop word deletion enhanced classification accuracy in the majority of situations, therefore it is suggested that stop word deletion be used for text categorization.

Stemming attempts to establish fundamental forms of words by removing plurals, verbs, and other additional words from nouns. A stem is made up of the same or extremely similar words as the rest of the sentence. Stemming also a method is used to determine the root or stem of a word. It converts words into their origins, allowing it to include a large amount of linguistic information based on language. Words like user, users, used, and using may all be stemmed back to the root word "use" (Gaigole, Patil, & Chaudhari, 2013). To minimize the size of the feature space and increase the text classification system's effectiveness, it is preferable to delete and stem the term.

2.5.4 Feature Extraction

Feature Extraction is a more comprehensive method that involves attempting to build a transformation of the input space onto such a low-dimensional subspace that maintains the majority of the important data. Dimensionality reduction, also known as feature extraction which allows the size of the feature space to be reduced without losing information from the original feature space. Another disadvantage of feature extraction is that the linear combination of the original characteristics is typically

unintelligible, and information about how much each original feature contributes is frequently lost (Khalid et al., 2014).

2.5.4.1 N-Gram

The n-gram character is made up of a series of n-word strings. A series of n fragments from a common text sequence is called an N-gram. For each letter or word, n-gram similarity algorithms compare the n-grams in two strings (Aiyar & Shetty, 2018a). A stream of n characters moving around the text produces the number of n-grams created for a certain document (typically $n = 1$ to 5). One tone at a time, the screen changes. Each n-gram, on the other hand, counts the number of occurrences. To extract the dominating character n-grams in a corpus, the method Local Maxs was devised. It's an algorithm that determines local maxima for each n-gram by comparing it to related n-grams. To avoid the sparse data problem that emerges in word-level n-grams, traditional word bag representation employs the n-grams character bag representation. The n-grams character representation bundle is language-independent and does not require text pre-processing.

2.5.4.2 Lexical Features

Lexical features provide the kind of keywords, characters, and attributes that wish to use, such as the number of upper case or the average length of the sentences. Syntactic features attempt to describe revisers writing style and include components. Several punctuations or words like "a," "the," and "of." When performing a cross-lingual task, lexical features may be used to classify the learners. Those semantic characteristics, such as those that can be extracted from word embedding and support the identification of paraphrastic reuse, will be used to classify the participants as a whole (Moritz & Steding, 2019). Lexical features are character or word-based features that indicate the types of words and characters that the writer prefers to use. Examples of lexical features include the number of characters in the upper case and the average length of a word. Lexical features are also used to identify the writer's style (Crawford et al., 2015). The lexical analyzer separates these words into several tokens by eliminating any white space or comment in the code. Meanwhile, lexical simplification (LS) is substituting complicated words in a given phrase with more basic alternatives with similar meanings.

2.5.4.3 Ensemble Method

Multiple learners are taught to tackle the same issue using ensemble methods. Unlike standard learning techniques that attempt to build one learner from training data, ensemble techniques attempt to build and combine a group of learners. According to Matteo & Giorgio (2001), the term "ensemble" refers to a group of machine learning that combine their choices or learning algorithms, various perspectives of data, or other unique features to make more consistent and accurate predictions in supervised and unsupervised learning situations. For example, the majority vote ensemble that combines the choices of various machine's learnings and the class that gets the majority of votes, the class predicted by most learning machines, is the class expected by the overall ensemble. This ensemble method benefits from alleviating the limited sample size by averaging and combining several classification models to minimize the possibility of overfitting the training data (Yang et al., 2010). Therefore, the training data set may be used more efficiently, which is crucial for many small sample size biological applications. The goal of developing and implementing the ensemble method is to obtain a better classification of training data.

2.5.6.2.1 Contribution of Ensemble Method

To summarize the early contributions that lead to ensemble methods, three threads may be defined, combining classifiers, ensembles of weak learners, and a mixture of experts. Thus, when it comes to pattern recognition, combining classifiers has been researched the most. Combining classifiers refers to how researchers work on strong classifiers and then connect the rules to create an even stronger combined classifier. The combination of several classifiers to produce a stronger classifier will improve comprehension. Consequently, this will use a version of combining rules in conjunction with this method to accomplish the desired outcome. However, while working with an ensemble of weak learners, the researcher will first work with weak learners before designing a robust algorithm to increase the ensemble's efficiency from weak to stronger in the Neural Network (NN), it was common to study a mixture of experts. The researcher in this technique will use the divide-and-conquer approach. It will learn how to combine parametric models and apply the combining rules resulting from the divide-and-conquer method.

2.5.6.2.2 Ensemble Learning Method

According to Yang et al. (2010), enhanced classification tasks frequently achieve by aggregating several classifiers as an ensemble committee and consent in predicting invisible data. As stated before, using ensemble methods aims to obtain a more accurate classification of training data and greater generalization. It is frequently accomplished, however, at the risk of higher model complexity. The traditional bias-variance decomposition analysis typically explains a superior generalization feature of the ensemble method. Expressly, earlier research indicated that methods such as bagging enhance generalization by reducing variance, whereas techniques comparable to boosting accomplish this by minimizing bias. Thus, the primary goal of these ensemble learning is to minimize bias and variation (Pandey, 2020).

2.5.6.2.2.1 Bagging

Bagging techniques constitute a family of algorithms that construct multiple instances of a black-box estimator on the original training set's random subsets, then combine their forecasts to produce a final prediction (Pandey, 2020). By invoking a base learning method, bagging trains a collection of base learners, each from a random bootstrap sample. A bootstrap sample is created by subsampling the training data set with replacement, with the sample size being equal to the size of the training data set (Zhou, 2009).

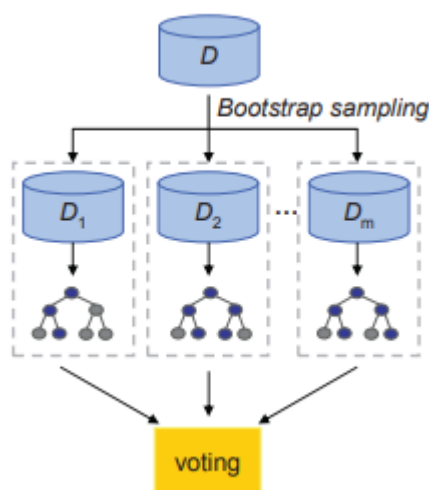


Figure 2.6: Illustration of Bagging Technique in Ensemble Method

Source: A review of ensemble methods in bioinformatics, 2016

2.5.6.2.2.2 Boosting

In boosting techniques, basic estimators are constructed progressively, and the combined estimator's bias is minimized. The goal is to combine many weak models to create a stronger ensemble. When used on weak learners, such as decision stumps, boosting may substantially decrease both the bias and the variance. As a result, boosting is often more successful when used on weak learners (Zhou, 2009).

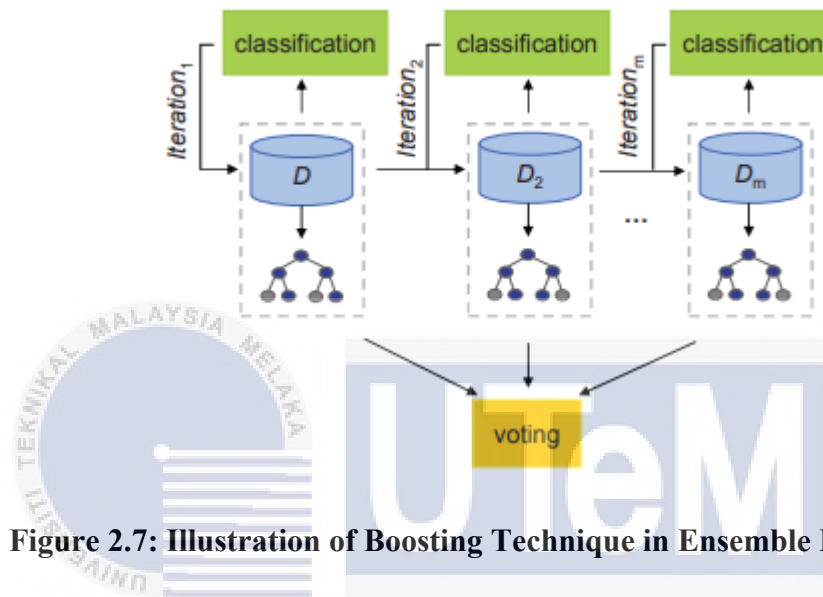


Figure 2.7: Illustration of Boosting Technique in Ensemble Method

Source: A review of ensemble methods in bioinformatics, 2016

2.5.6.2.3 Ensemble Combination Rules

Different combinations of rules, such as Majority Voting and Weighted Average Voting, may combine machine learning classifiers. In machine learning, a Voting Classifier is a model that trains on an ensemble of several models and produces a result of class based on the greatest highest likelihood of the selected class as the result.

2.5.6.2.3.1 Majority Voting

The majority voting ensemble is the most widely used ensemble fusion technique. Each base classifier votes for a particular class in this method and the class that receives the most votes predicted by an ensemble. In each test instance,

each model makes a prediction, and the final feature prediction is the one that gets the most votes (Pandey, 2020).

2.5.6.2.3.2 Weighted Average Voting

The researchers may enhance one or more models' significance by giving weights to them in weighted average voting. Each model's prediction is multiplied by the weight, and the average is determined (Pandey, 2020). Whenever there is a weighted vote, the voting weights should be different for the various output classes for each classifier. There should be a significant amount of weight placed on a specific output category whereby the classifier performs well. As a result, selecting the proper weights of votes for all of the classes of each classifier is a significant problem. It is possible to think of the weighting issue being an optimization problem (Zhang et al., 2014).

2.5.5 Data Splitting and Validation

Several different data sets are used in a conventional multivariate calibration approach. Estimating an equation's parameters or training a neural network to set up the model both require the validation or training data set. All of these data sets have specific data and without it, the models and forecasts would be affected. Many researchers use the existing data set in their research. To generate the product, the process is carried out through training and testing. There are numerous subsampling techniques, which include random subsampling, cross-validation, and bootstrapping.

2.5.5.1 Random Subsampling

The accuracies from each division are combined in random subsampling, which is partitioned into disjoint training and numerous test sets. Random subsampling, also known as multiple holdouts or redundant analysis tests, is concerned with randomly dividing data into subsets, with the size of the subgroups being determined by the client. The random subsampling method has the advantage of being able to be repeated continually. According to a prior study, random subsampling is a better predictor than cross-validation and can be used to divide data into calibration, testing, and validation subsets (Hajare Akash, 2021).

2.5.5.2 Cross-validation

One of the most commonly used data resampling approaches for estimating the genuine prediction error of models and tuning model parameters is cross-validation. Testing can be done based on accuracy with test data parameters and training data given by cross-validation to provide data classifications that have data accuracy or similarity in the measurement result to the actual numbers or data (Hulu & Sihombing, 2020). For k-fold cross-validation, the data is divided into k equal parts. A given set of data is divided into a K number of parts or folds, each of which is utilized as a test set at some level in K-fold cross-validation.

2.5.5.3 Bootstrapping

A bootstrap method is an approach in which the training dataset is selected at random and replaced. Bootstrap resampling was created to help researchers figure out what would have happened if they used a different random sample instead, and how different the expected findings would be when applying a model to new data. Testing is done on the remaining examples that were not chosen for training. The value is likely to change from the fold to the fold, unlike K-fold cross-validation. The average error rate of each iteration is used to calculate the model's error rate.

2.5.6 Feature Selection

2.5.6.1 Feature Selection Definitions

The process of obtaining a subset from an initial feature set according to a feature selection criterion that picks the relevant features of the dataset is referred to as feature selection. It aids in the compression of data processing scales by removing unnecessary and irrelevant characteristics (Cai et al., 2018). As a result, feature selection improves the comprehension of data, the reduction of processing requirements, the reduction of the effect of the curse of dimensionality, and the improvement of predictor performance. It's also to pick a subset of variables from the input that can accurately characterize the data while decreasing the impacts of noise or irrelevant variables while still producing good prediction results (Chandrashekar & Sahin, 2014).

2.5.6.2 Feature Selection Type

As a dimension reduction strategy, feature selection seeks to choose a small subset of useful features from the original features by deleting the unnecessary, redundant, or noisy features (Miao & Niu, 2016). The process of selecting the best characteristics from among all the features that can be used to distinguish classes is known as feature subset selection. Feature selection can usually result in better learning performance, such as increased learning accuracy, lower computing cost, and improved model interpretability. To describe the best subset, selecting the features should come with a variety of variable selection methods and can be done without transforming data to a reduced value dataset. Wrapper method, filter method, embedded approach, information gain and chi-squared approach are some of the techniques available for variable selection.

2.5.6.2.4 Wrapper

The wrapper method evaluates the features using the intended learning algorithm. Because the feature selection process is designed for the classifier to be used, the wrapper method defeats filter approaches. To find the most relevant gene to cancers, the researchers used Support Vector Machine (SVM) algorithms based on Recursive Feature Elimination (RFE). Unfortunately, because of the high computational cost of using the wrapper method for a considerable feature space, each feature set should be assessed and evaluated with the trained classifier, resulting in a moderate feature selection procedure (Khalid et al., 2014).

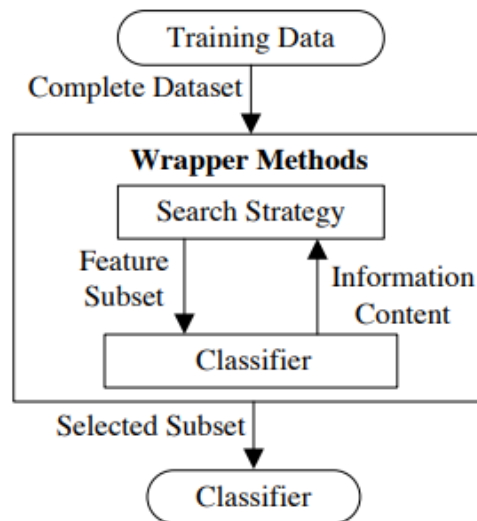


Figure 2.8: Wrapper Methods

2.5.6.2.5 Filter

The filter method uses the characteristics of data to identify the most discriminative features. Filter methods, in general, provide feature selection before classification and clustering tasks and follow a two-step process. For the first, all features are ranked using a set of criteria. The features with the highest ratings are then chosen (Miao & Niu, 2016). When compared to the wrapper method, the filter method has a lower computational cost and is lighter and faster. But they have poor classification reliability and are better suited to high-dimensional data sets (Khalid et al., 2014).

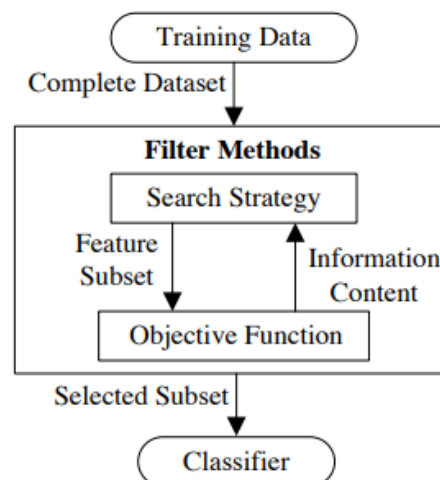


Figure 2.9: Filter Methods

2.5.6.2.6 Chi-Square (CS)

According to Chandrashekar & Sahin (2014), embedded method allows for variable selection without the need to divide data into training and testing sets. Recently, an embedded method is created that combines the benefits of both the filter and wrapper approaches. During the model creation process, embedded models do the feature selection. Chi-Square (CS) is a prominent feature-selection method that examines individual features by computing CS statistics for classes. This method is a statistically recoverable, individually variable assessment and computation. Model accuracy is improved by using CS feature selection to create the best feature set.

2.5.7 Classification

The classification phase is given as a supervised procedure, in which each sample is assigned to a category, indicating the relevance of a certain purpose attribute or merely the class attribute. The physical quality can take on absolute values, each of which corresponds to a class. The classification process separates the sample set into two mains, the training and test sets. The classification process is separated into two stages, training and evaluation. Training occurs when a classification model is developed from the training set, and evaluation occurs when the model is evaluated in the test set.

2.5.8 Classification Type

2.5.8.1 Generative

In general, generative models that can simulate and analyze sequences of future occurrences would learn to capture complicated real-world phenomena like physical interactions. Hence, generative systems are primarily intended to address the underlying distribution of information by developing a system that matches training data. The use of huge unlabeled datasets in combination with predictive generative models provides a good alternative to supervised learning. A sophisticated generative model must construct an internal representation of the world to properly anticipate future occurrences (Kumar et al., 2015). Gaussian, Naive Bay, Multicultural Mixtures,

Hidden Markov Model (HMM), Bayesian Gaussian, and Markov random fields are examples of artificial data points produced via pre generative probability as samples.

The supervised probabilistic classification Naive Bayes (NB) classifier creates a set of probabilities by calculating the frequency and value combinations of a given data set. The Naive Bayes classification has lately achieved a lot of popularity and has demonstrated to be quite effective. The data is gathered, and a probabilistic technique based on assumptions is proposed. By using Bayes' law, a new sample is generated, along with the category in which it is most likely to have been put. The Naive Bayes classification is perhaps the easier and often used classifier. Although their simplicity and limited assumptions, Naive Bayes classifiers perform well in difficult circumstances. These classifiers have the advantage of just requiring a good bit of training data to assess the classification parameters. According to Tran (2019), this is the text classification algorithm of choice.

According to Pietrzykowski & Saifun (2014), the Hidden Markov Model (HMM), named after the Russian mathematician Andrey Markov, is statistical and the predicted labels of surrounding keywords are frequently ignored when using a probabilistic method. It is indeed a broad and valuable category of stochastic processes. Hence, it is defined by the Markov Property, which states that the process's future state is solely determined by the current state and not by the sequence of events that preceded it. The HMM function enables it to be combined into bigger ones, with each HMM being trained individually for each data class. The HMM can handle bag-of-word sequences, and each state has its page category. Furthermore, because HMM is trained using partially labeled page sequences, state variables in the training set are only partially observed and detected (Kang et al., 2017).

Bayesian networks (BNs) portray systems as a web of interactions between variables, from the fundamental cause to the ultimate consequence, with all cause-effect assumptions made clear. Because of its capacity to combine many challenges, interactions, and consequences, as well as analyze tradeoffs, BNs are frequently considered suited for modeling environmental systems (Chen & Pollino, 2012). The edges between nodes illustrate probabilistic dependency between random variables, although every node in the network is a random variable (Tran, 2019). Sparse BNs,

such as the Naive Bayes model and Hidden Markov Models, can capture them, but BNs with great complexity can capture them. As a result, BNs provide a reliable simulation system.

2.5.8.2 Discriminative

The discriminative models analyze cultural differences by comprehending any vocabulary to describe the probability processes behind computing assets aim at improving task performance. Discriminative methods for unsupervised and supervised learning, as well as other particular algorithms, are intended to be instructive rather than comprehensive. For several application areas, discriminative classifiers have been claimed to develop higher test set accuracy. Popular models include the Support Vector Machine (SVM), Logistic Regression, Traditional Neural Networks, Nearest Neighbor (NN), and Conditional Random Fields (CRF) (Yogatama et al., 2017).

The Support Vector Machine is a comparatively recent and promising approach for learning different functions in pattern recognition (classification) tasks or conducting function estimates in regression issues (Vapnik, 1995). Meanwhile, according to Noble (2006), Support Vector Machine (SVM) is a computer algorithm that assigns labels to things by learning by example. For example, by reviewing hundreds or thousands of fraudulent and non-fraudulent credit card activity records, an SVM may learn to distinguish fraudulent credit card activity. SVM has unique qualities that allow it to minimize an empirical classification error while also increasing the geometric margin.

The Nearest-Neighbor (NN) approach is fairly basic and may be customized in a number of ways. It's simple to use, accurate, and flexible to a variety of challenges (Yu et al., 2002). Besides, Nearest-Neighbor (NN) algorithm to data categorization calculates the possibility of a data point belongs to one of two groups based on which data points are nearest to it. The rule determines the unknown category of data points based on the class of its nearest neighbor. Based on Bhatia & Vandana (2010), pattern recognition, text classification, model ranking, object identification, and event recognition applications all apply this rule.

2.6 Critical Review

2.6.1 Previous Research on Spam

2.6.1.1 Research Paper I

In recent years, social networking sites have grown in popularity. Users use them to make new connections and keep in touch with existing mutual by sharing their current opinions and activities. Twitter is the most significant rising of these platforms. Based on research by McCord & Chuah (2011), they look at the differences between spammers' tweets and genuine users' tweets in this research. Their objective is to find relevant traits that may be used in typical machine learning systems to detect spam and authentic accounts automatically. They used a combination of user-based and content-based criteria to detect spam in tweets. The retrieved features may be divided into two categories, user-based features, and content-based features. User-based features are based on a user's relationships, such as who a user follows, referred to as friends, and who follows a user, referred to the followers, as well as personal activities, such as the periods and frequency with which a user's tweets. They use various characteristics for content-based features, including replies or mentions, keywords, retweets, and hashtags. They analyze the performance of four classic classifiers, including Random Forest, Support Vector Machine, Naive Bayesian, and K-Nearest Neighbor classifiers.

2.6.1.2 Research Paper II

Research by Olatunji (2019), at the current time, email has become highly popular. It has recently been recognized as the cheapest, most popular, and quickest mode of communication. Despite the numerous advantages of email, its use has been affected by the widespread presence of unwanted emails. This research paper examines based on a support vector machines (SVM) model to improve spam detection accuracy while paying to special attention to exhaustive parameter search approaches. SVM is a statistically based machine learning algorithm that can model complicated relationships between variables. The formulation in SVM generates a global quadratic optimization problem with box constraints, which can be solved quickly using interior point techniques. SVM specials can readily translate non-separable issues to higher dimensions, where they can be readily separated, thanks to their kernel functions.

2.6.1.3 Research Paper III

According to the research by Amir Sjarif et al. (2019), Short Message Service (SMS) traffic continues to rise daily. As a result, mobile attacks, such as spammers flooding the service with spam messages sent to users, have increased dramatically. Mobile spam is becoming more of a concern as the number of spams continues to rise day by day, despite filtering issues. Pre-processing is the initial step in the conversion of unstructured data to more structured data. Because symbols are often replaced for terms in SMS text messages. Stop words in SMS were eliminated using the stop word list remover for the English language in this research.

Table 2.A: SPAM Literature

Author	Title	Technique	Result	Dataset
M. McCord, M. Chuah	Spam Detection on Twitter Using Traditional Classifiers	Random Forest, Support Vector Machine, Nave Bayesian, and K-Nearest Neighbor are the performance of four conventional classifiers.	Using Random Forest Classifier, the findings demonstrate that spam detection system has an accuracy of 95.7% and a F measure of 95.7%.	Twitter spam datasets
Sunday Olusanya Olatunji	Improved email spam detection model based on support vector machines	Using SVM technique of Spam Classification Support.	The training and testing sets had accuracy of 95.87 and 94.06%, respectively.	Spam base obtained from UCI machine learning repository
Nilam Nur Amir Sjarif, Nurulhuda	SMS Spam Message Detection using	The Term Frequent-Inverse Document	With an accuracy of 97.50%, the Random	The data was collected

Firdaus Mohd Azmi, Suriayati Chuprat, Haslina Md Sarkan, Yazriwati Yahya, Suriani Mohd Sam	Term Frequency-Inverse Document Frequency and Random Forest Algorithm	Frequency (TF-IDF) and Random Forest Algorithm will be used.	Forest method beats other methods.	from the UCI Machine Learning Repository.
--	--	--	------------------------------------	---

2.6.2 Previous Research on YouTube Spam

2.6.2.1 Research Paper I

According to the research by Aiyar & Shetty(2018), they proposed a new method for detecting spam or inappropriate comments on the YouTube video-sharing network. These comments have a significant impact on a channel's reputation and credibility as well as the experience of current users. YouTube's solution to these problems would be to restrict comments with links, which is a very narrow solution. The researchers use traditional machine learning methods like Random Forest, Support Vector Machine (SVM), and Naive Bayes, as well as heuristic methods like N-Grams, to try to recognize these comments throughout this research. The purpose of having to classify algorithms and employing heuristics like N-Grams that can reliably detect spam with a high F1 rate. Hence, increase classification accuracy.

2.6.2.2 Research Paper II

Broadband's adoption rate has increased the number of people who use the Internet. Video streaming and sharing services grew in popularity as customers' internet connections improved. According to research by Alberto et al. (2016), YouTube is a well-known video content distribution platform with social network features like support for text comments, which allows channel owners and users to communicate. YouTube has implemented a monetization mechanism to encourage creators to create high-quality original content while also expanding the visualization. Following the implementation of this mechanism, the platform was flooded and overwhelmed with offensive content, much of which was low-quality information

known as spam. The main purpose of this research is to discover promising methods and parameters for an online application designed to detect inappropriate text comments on YouTube. This research also includes a thorough review of many well-known machine learning approaches for automatically filtering such unwanted comments. Furthermore, any preprocessing was carried out, such as stop word removal or stemming, because some study findings suggest that such approaches reduce spam classifier performance.

2.6.2.3 Research Paper III

Next study by Alper Kursat Uysa (2018), one of the most common types of text classification is spam filtering. The results of five state-of-the-art text filtering algorithms for YouTube spam filters employing the two well-known classifiers, Naive Bayes and Decision Tree. The tests used five data sets, including spam comments from various subjects. Because various features available, filter-based techniques are often applied for text grading, and these techniques do not interact with classifiers during the selection process. Five well-known filter-based text selection techniques are used in this study. The Information Gain (IG), Gini Index (GI), Distinguishing Feature Selector (DFS), Discriminative Feature Selection (DFSS), and Relative Discriminative Criteria (RDC) were all included.

2.6.2.4 Research Paper IV

Next previous research by (Mehmood et al., 2018), spam is a diversionary tactic in today's era of digital communication. Massive information transmission is supported by the ongoing increase in users on social media platforms such as Facebook, Twitter, YouTube, and others. Usually, this data spreads via comments and reviews that opened up new channels for spammers. Thus, in this research, the researchers solve the constraints mentioned earlier and introduce a new spam comment detection model based on stacking with ensemble learning method on Term Frequency (TF) and Inverse Document Frequency (IDF) text features. Compared with previous approaches that are single learning classifiers, the researchers suggested a model that focuses on combining classifier functions to obtain optimum predictive accuracy.

Table 2.B: YouTube SPAM literature

Author	Title	Technique	Result	Dataset
Shreyas Aiyar, Nisha P Shetty	N-Gram Assisted YouTube Spam Comment Detection	Classifiers that have been used Random Forest, SVM, and Multinomial NB n values ranging from 1 to 6 in a 5-fold cross-validation.	The highest score was 0.984 for SVM with 6 grams for character gram.	Open Youtube API from different Youtube channels.
Tulio C. Alberto, Johannes V. Lochter, Tiago A. Almeida	TubeSpam: Comment Spam Filtering on YouTube	Support Vector Machine (SVM), Decision Tree, Naïve Bayes, Random Forest, K-Nearest Neighbors.	The majority of them were able to attain accuracy rates of over 90% with very low or no blocked ham rates.	The datasets were collected from Psy, KatyPerry, LMFAO, Eminem and Shakira, which publicly available at YouTube Spam Collection.
Alper Kursat Uysal	Feature Selection for Comment Spam Filtering on YouTube	NB and Decision Tree were used as classifiers. Cross-validation three times. Micro-F1 is a metric that is	With 50 features, the Eminem dataset employing DT has the greatest score of 96.61%. The DT classification has a better track record	The data was collected from the UCI Machine Learning Repository.

		used to assess performance.	than the NB classification.	
Arif Mehmood, Byung-Won On, Ingyu Lee3, Imran Ashraf, Gyu Sang Choi	Spam comments prediction using stacking with ensemble learning	Evaluate several state-of-the-art classification methods and conclude that Decision Trees, Logistic Regression, Naive Bayes Bernoulli, Random Forests, and Support Vector Machines.	TF/text IDF's characteristics for innovative stacking model strategy performance selection provide 92.19% accuracy that is considered superior compared to single model base or even stacking with NB.	The data was collected from the UCI Machine Learning Repository.

2.7 Conclusion

This chapter includes a much more detailed overview of an internet security attack, as well as classification, spam machine learning, and a critical review. In the critical review category, a hypothetical study of past studies into spam and YouTube spam is mentioned. For this research, data sets are collected from the UCI Machine Learning Repository. The proposed methodology will be explained in more detail in the following chapter.

CHAPTER 3: PROJECT METHODOLOGY

3.1 Introduction

This chapter will describe the methodology and technique used in this study and provide a detailed overview. In addition, this study will develop a framework based on numerous key areas described in the earlier chapter. The framework is required to guarantee that study should be carried out and completed following the proper steps and procedures, as well as that the plan is implemented effectively within a certain timeframe. This project's flowcharts, milestones, and Gantt Chart show the progress that can be achieved to finish the task within the timeframe.

3.2 Methodology

It is critical to guarantee that this project is implemented according to the suggested flow. The methodology is a set of ways or techniques to handle certain difficulties through methodological approaches. Thus, methodology is used to look at theoretical approaches that show how the system can accomplish and satisfy the project's goals. This research will follow the six phases, ensuring that the project runs well. Previous research, information gathering, defining scope, design, and implementation, system assessment methods, and documentation are all included. The structure of the methodology model used in this project is shown in Figure 3.1.

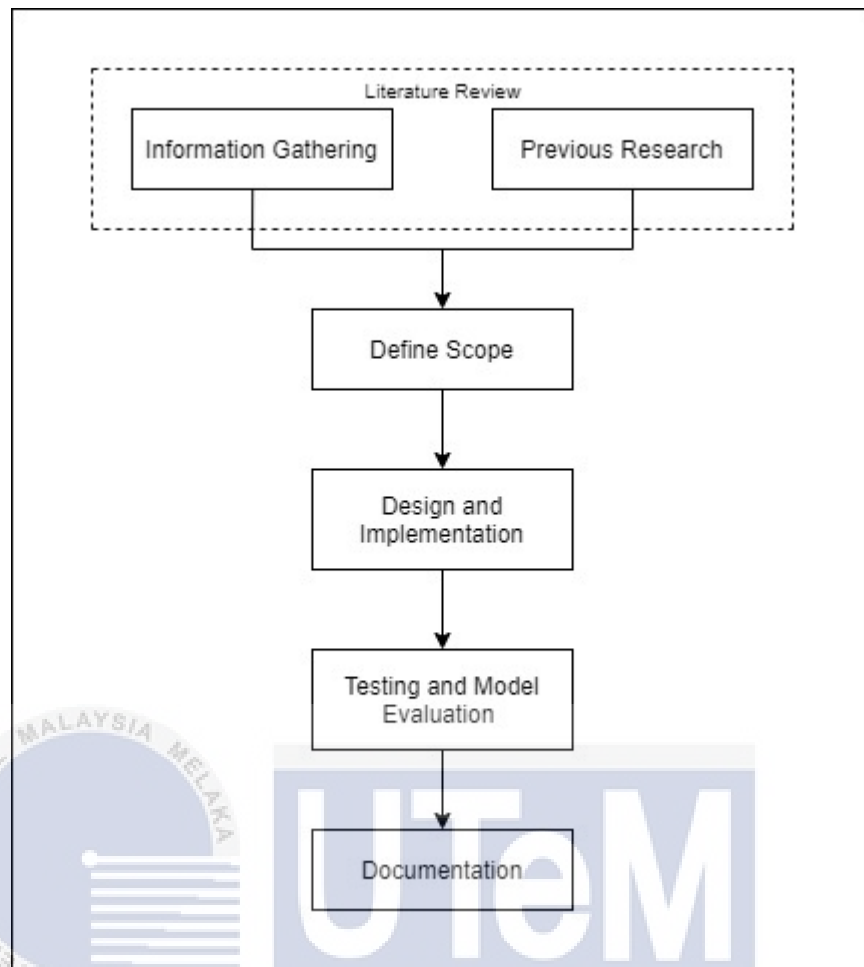


Figure 3.1: Framework of the System

3.2.1 Previous Research

Based on previous research, this phase provides a more comprehensive assessment of the project's execution. It's essential to double-check that all needs have been specified in an earlier study. Because the domain is necessary before the project can be completed to address the projects that have been identified. Spam types, learning machines, feature extraction, feature selection, and classification method are all examples of domains. As a result, previous research will give an overview of how the proposed theoretical frameworks work in their respective fields. In the phase, it is addressed in depth.

3.2.2 Information Gathering

The gathering of information for the purpose of gaining a comprehensive understanding of research topics. The proof of how serious the problems are has been established. The SVM classification system, on the other hand, is used in this study. The gathering of information from previous studies will help in the selection of algorithms and methodologies for this study.

3.2.3 Define Scope

The project's scope is constrained because the research is focused on analyzing YouTube spam data. As a consequence, the YouTube spam collecting data set from the UCI Machine Learning Repository is being used. Meanwhile, new approaches are being proposed for identifying spam comments with more accuracy or fewer false positives.

3.2.4 Design and Implementation

This section explains how the new models are implemented based on past research. The dataset was obtained from the UCI Machine Learning Repository first. The data set is then divided into a few key characteristics for this study. The data set is then retrieved and trained using ensemble method features. The new feature set is then employed in classification with SVM, which resulted in the development of a newer detection model.

3.2.5 Testing and Evaluation of Model

The evaluation and testing process is used to see if the produced model satisfies the specified requirements. In this context, the accuracy is compared using the characteristics created at the end of the procedure. Comprehensive design analysis will guide future work on the research's development.

3.2.6 Documentation

The documentation process aids in the organization of the outcomes in a more structured and appropriate way. Each experiment is properly written, with all methods

restricted and enough documentation that serves as a reference and proof of each activity.

3.3 Project Schedule and Milestones

3.3.1 Project Flowchart

A flowchart is used to create a systematic overview of the tasks and their relationships. To avoid delays or other limitations, this procedure defines the required resources that must be mapped to their appropriate tasks. The flow diagram of the project's general phases is shown in Figure 3.2.

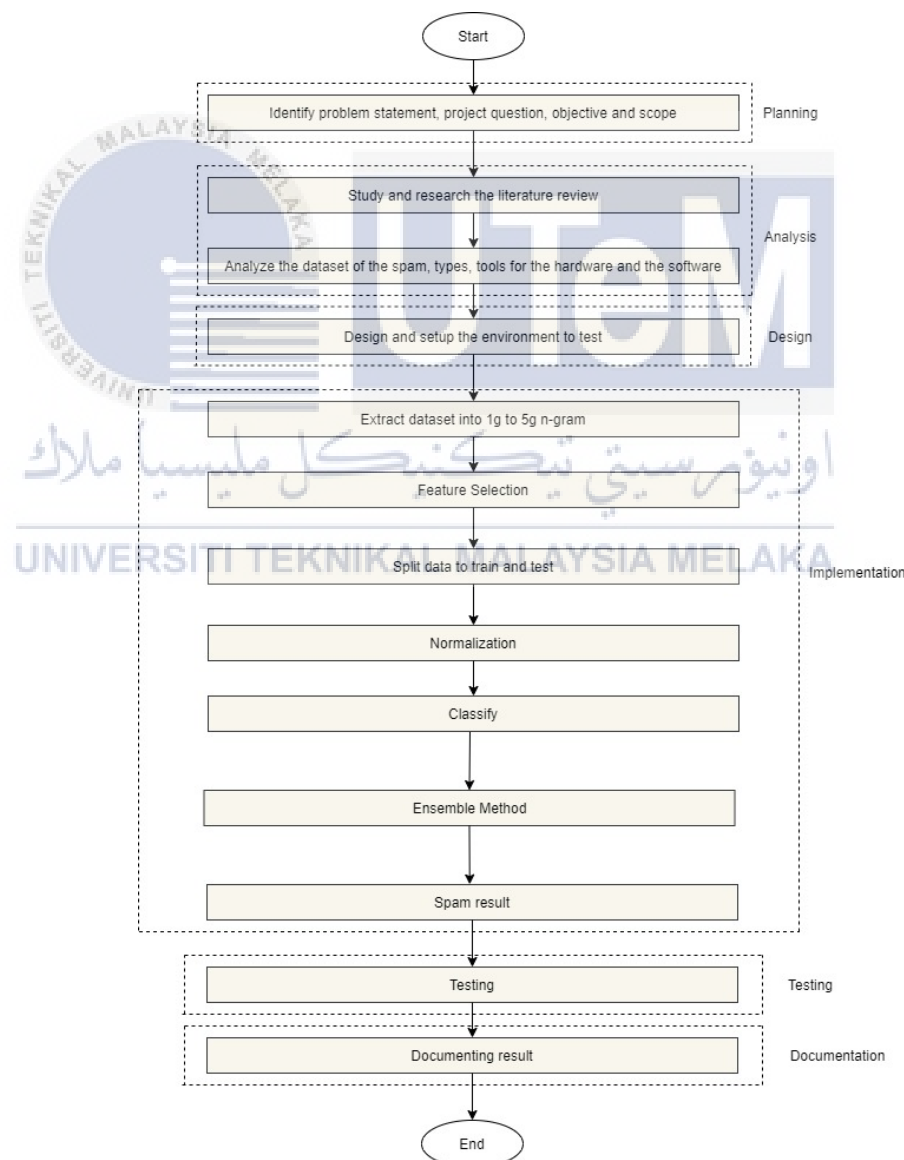


Figure 3.2: Project Flowchart

3.3.2 Project Milestones

The project milestone is crucial for keeping track of forthcoming events or goals throughout the timeline. Using milestones has significant importance since it helps us to judge whether or not the plan is developing according to the timeline. The research project's milestones are listed in Table 3.1.

Table 3.A Project Milestone

Week	Phase	Action	Deliverables
1-4	Planning	(15/3/2021) – (21/3/2021) Proposal discussion with Supervisor. Identify title, problem statement and scope.	Complete Proposal
		(22/3/2021) – (28/3/2021) Study and research the literature review. Write and submit project proposal to Supervisor and Committee.	
		(29/3/2021) – (4/4/2021) Proposal accepted. Identify title, problem statement, objective and scope of project	Chapter 1: Introduction
		(5/4/2021) – (11/4/2021) Chapter 1 is done and submit to supervisor for evaluation.	Progress report Chapter 1
5-9	Analysis	(12/4/2021) – (25/4/2021) Studies on related work and previous research and finding of spam classification.	Chapter 2: Literature Review
		(26/4/2021) – (9/5/2021) Study methodology on previous research.	Chapter3: Methodology
		(10/5/2021) – (16/5/2021)	MID SEMESTER BREAK

10-16	Design	(17/5/2021) – (23/5/2021) Information collection and analysis. Design the project and choose the tools for implement.	Chapter 4: Analysis and Design
		(24/5/2021) – (6/6/2021) Design the environment for implementation.	Progress report on Chapter 4
		(7/6/2021) – (20/6/2021) Project demonstration and report.	Progress report on Chapter 4 and demonstration
		(21/6/2021) – (27/6/2021) Final project demonstration.	Progress report on Chapter 4 and demonstration
Sem Break			

3.3.3 Project Gantt Chart

The Gantt chart is a timetable for each project activity. When one of the project's operations is late, it has an impact on the project's overall durability and costs, which are projected to rise in the future.

Refer to appendix section.

3.4 Requirement Analysis

Many criteria are setting to carry out this research successfully. Software and hardware are examples of such methods, which are detailed in the following sections.

3.4.1 Software Requirement

Some software is provided in this project to finish the development of the system, as well as an explanation of how to use it. Table 3.2 lists the software that used in the project.

Table 3.B: Software Requirement for the Project

Software	Description
Windows 10	An environment of operating system used for project execution.
Microsoft Word 2016	Software used to complete the project reporting and documentation.
Microsoft Excel 2016	Software to sort the data according to attributes and instances.
Microsoft Project 2016	Software used to develop a Gantt Chart.
draw.io online	Software used to create flowchart and draw diagrams.
Eclipse IDE Application	Software used to code execution for machine learning algorithm implementation in Java.

3.4.2 Hardware Requirement

As a workstation, the laptop is used for all activities, from reporting to study.

Table 3.3 shows the laptop specifications.

Table 3.C: Hardware Requirement of the Project

Specification	Description
Processor Type	Intel(R) Celeron(R) N4000 CPU @ 1.10GHz 1.10 GHz
Operating System	Windows 10 Home Single Language
Operating System Architecture	64 bits
RAM	4 GB
Storage	500 GB HDD and 118 GB ADATA SSD
Display Resolution	1366x768
WLAN	802.11n

3.5 Conclusion

To summarize, the methodology is the most crucial and critical step towards the development and evaluation of a project's progress. This chapter discusses each of their phases and approaches. The main goal of this research is to determine the accuracy of spam comments on YouTube using all available techniques and methodologies. The approach and methods of designing learning machines with SVM will be discussed in-depth in the next chapter.



CHAPTER 4: ANALYSIS AND DESIGN

4.1 Introduction

The chapter discusses the design and analysis of the methodologies used in the project. The method of the studies and the form of the project's design create the connection between the tasks presented and the structure of the project's design. Additionally, the previous chapter discussed one implementation of the methodology's concept. However, the analysis and design phases result in a thorough grasp of the plan's nature and a specific model that eliminates potential errors.

4.2 Problem Analysis

The primary objective of this research is to create a model of machine learning to distinguish between spam and legitimate comments on YouTube. The method of detection is done automatically by the algorithm of machine learning. Besides, to verify five distinct features of the datasets: comment id, author, date, content, and class. Unfortunately, this study focuses on the two final features, which are content and class features, as they are difficult to identify between spam and legitimate comments to encompass all five features. This research will address YouTube Spam Classification using the Support Vector Machine (SVM), a machine learning technique widely used in machine learning. Hence, this is the most effective way of problems resolution.

4.3 Project Design

This project's design will show how to proceed with research studies. The main idea for spam comments detection in a machine learning system is presented in Figure 4.1. Obtaining datasets, pre-processing data, extracting features, and generating

BOW are all steps in the process. Then there was the procedure of splitting the train and test sets into two parts, normalization and feature selection. Moreover, analysts must be selected by the use of a machine learning study's research dataset. The data set is the raw data obtained for a particular study field and used to train machine learning algorithms to identify spam comments.

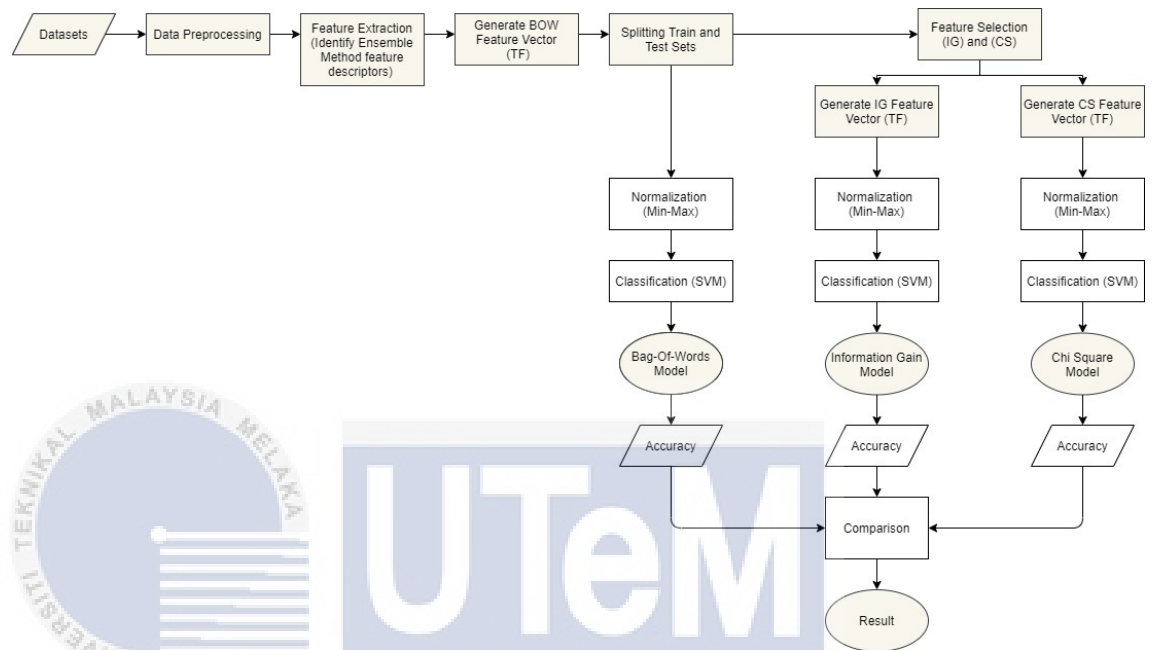


Figure 4.1: Project Design

4.3.1 Dataset

A data set is essential to analyze and apply the specified machine learning algorithm's concept and theory. The dataset is the raw data collected with this research issue. The datasets used in this paper are the YouTube comments from the Machine Learning Repository's YouTube Spam Collection Data Set. The datasets include five YouTube video comments with a total of 1956 comments. There are a total of 1005 spam comments, with the remaining as legitimate. The datasets addressed in Table 4.1 are provided.

Table 4.A Description of Dataset

Dataset	Number of Spam	Number of Legitimate	Total comment
Psy	175	175	350
Katy Perry	175	175	350
Eminem	245	203	448
LMFAO	236	202	438
Shakira	174	196	370
TOTAL	1005	951	1956

COMMENT_ID	AUTHOR	DATE	CONTENT	CLASS
LZQPQhLyRh80U	Julius NM	2013-11-07T06:20:48	Huh, anyway check out this you[tube] channel: kobyoshi02	1
LZQPQhLyRh_C2c	adam riyati	2013-11-07T12:37:15	Hey guys check out my new channel and our first vid THIS IS US THE MONKEYS!!! I'm	1
LZQPQhLyRh9MS	Evgeny Murashkin	2013-11-08T17:34:21	just for test I have to say murdev.com	1
z13jhp0bxqncu51	ElNino Melendez	2013-11-09T08:28:43	me shaking my sexy ass on my channel enjoy ^_^i»¿	1
z13fwbwp1oujth	GsMega	2013-11-10T16:05:38	watch?v=vtaRGgvGtWQ Check this out .i»¿	1
LZQPQhLyRh9-wf	Jason Haddad	2013-11-26T02:55:11	Hey, check out my new website!! This site is about kids stuff. kidsmediausa . com	1
z13lfzdo5vmidi1o	ferleck ferles	2013-11-27T21:39:24	Subscribe to my channel i»¿	1
z122wfnzgt30fhu	Bob Kanowski	2013-11-28T12:33:27	i turned it on mute as soon is i came on i just wanted to check the views...i»¿	0
z13ttt1jcraxek2c	Cony	2013-11-28T16:01:47	You should check my channel for Funny VIDEOS!!i»¿	1

Figure 4.2: Example of Psy dataset

CONTENT	CLASS
Huh, anyway check out this you[tube] channel: kobyoshi02	1
Hey guys check out my new channel and our first vid THIS IS US THE MONKEYS!!! I'm	1
just for test I have to say murdev.com	1
me shaking my sexy ass on my channel enjoy ^_^i»¿	1
watch?v=vtaRGgvGtWQ Check this out .i»¿	1
Hey, check out my new website!! This site is about kids stuff. kidsmediausa . com	1
Subscribe to my channel i»¿	1
i turned it on mute as soon is i came on i just wanted to check the views...i»¿	0
You should check my channel for Funny VIDEOS!!i»¿	1

Figure 4.3: Example of features will be use

4.3.2 Data Preprocessing

Pre-processing is a term that refers to activities that convert raw data to data that a machine learning algorithm can easily understand. Pre-processing is the initial step in which highly doubt sources and documents are subjected to specific enhancements such as tokenization, token length, case normalization, special character removal, stop word removal, and stemming.

```

Suscribe My Channel Please XD lol!>
Wow. Comments section on this still active. Not bad. Also 5277478 comments. (Now 79) i>
http://binbox.io/lfiro#l23i>
Ching Ching ling long ding ring yaaaaaa Ganga sty FUCK YOU.i>
https://www.indiegogo.com/projects/cleaning-the-pan--2 please help me with my projecti>
There is one video on my channel about my brother...i>
Check my channel, please!i>

```

Figure 4.4: Raw Data

Tokenization achieves by gathering and analysing the contents of the comments. White spaces, symbols, or special characters signify the breakdown of sentences into words and tokens. The word reduced included characters such as the basic delimiter, brackets, mathematical operators, and special characters.

```

Suscribe My Channel Please XD lol!>
Wow . Comments section on this still active . Not bad . Also 5277478 comments . (Now 79) i>
http : //binbox.io/lFIRO#l23i>
Ching Ching ling long ding ring yaaaaaa Ganga sty FUCK YOU . i>
https : //www.indiegogo.com/projects/cleaning-the-pan--2 please help me with my projecti>
There is one video on my channel about my brother ... i>
Check my channel , please ! i>

```

Figure 4.5: Tokenization

Therefore, must refine each tokenized term. The token length is determined by the number of letters in each word-processed, between two to ten. Words that begin with a number one and greater than 11 will be noted only for their frequency.

```

Suscribe My Channel Please XD lol!>
Wow . Comments section on this still active . Not bad . Also 5277478 comments . (Now 79) i>
http Ching Ching ling long ding ring yaaaaaa Ganga sty FUCK YOU . i>
https please help me with my projecti>
There is one video on my channel about my brother ... i>
Check my channel , please ! i>

```

Figure 4.6: Token Length

The collection of data sets may include the unique character in both lower and upper case. Special characters and lower-case letters have to process and grouped appropriately. This part includes eliminating all full stops, semicolons, symbols, quote marks and converting all nouns to lower case.

```
suscribe my channel please xd lol!>
wow . comments section on this still active . not bad . also 5277478 comments . (now 79) i>
http ching ching ling long ding ring yaaaaaa ganga sty fuck you . i>
https please halp me with my projecti>
there is one video on my channel about my brother ... i>
check my channel , please ! i>
```

Figure 4.7: Case Normalization

```
suscribe my channel please xd lol
wow comments section on this still active not bad also 5277478 comments now 79
http ching ching ling long ding ring yaaaaaa ganga sty fuck you
https please halp me with my project there is one video on my channel about my brother
check my channel please
```

Figure 4.8: Special Character

Stop words may help minimize the size of the database while still processing precisely chosen terms. The most often used stop words in English are a, about, an, are, at, be, by, for, and how. Generally, stop words do not provide context or meaning to textual documents.

```
suscribe channel please xd lol wow comments section active bad 5277478 comments 79
ching ching ling long ding ring yaaaaaa ganga sty fuck
https please halp project channel brother channel please
```

Figure 4.9: Stop Word Removal

Stemming is a term that refers to the process of converting multiple word forms into a single recognizable form, as well as the reduction of words to their roots. This phase assists in circumstances when the data set is not very big and considerably enhances the dependability of the predicted output and the frequency and identification of training for a variety of classifiers. Apart from that, it helps limit the number of features, which helps keep the models' sizes manageable.

```
suscrib channel pleas xd lol wow comment section activ bad 5277478 comment 79
ching ching ling long ding ring yaaaaaa ganga sti fuck
http pleas halp project channel brother channel pleas
```

Figure 4.10: Stemming

The advantages and disadvantages of implementing the rules for this research shown in Table 4.2.

Table 4.B: Advantage and Disadvantage of Rules

Rules	Advantage	Disadvantage
Tokenization	Breaking the unprocessed text into sentences and then tokens.	If the token size is too large, it will not be easy to distinguish between usable outputs. It will not be easy to generalize a positive relationship between both the texts and labels.
Stop Word Removal	Stop word removal improves classification accuracy in most situations and minimizes database size by processing the chosen words.	Words enhance the vector space, complicating the classification procedure.
Stemming	Reduce the number of features in the space vector by converting words to their roots. This rule improves the learning speed and efficiency of the text classification system.	Stemming may occur in the same word-stemming into two different words.
Case Normalization	There is no difference between upper-case and lower-case variants of words. With fewer features, more significant discrimination is possible. Thus, replace the whole comments with lower-case or upper-case letters.	The constant comments are those that have the same meaning.

4.3.3 Feature Extraction

The feature extraction process begins with a baseline set of measured data and generates relevant and non-redundant values. This study uses an ensemble approach to build features and train numerous learners to solve the same issue, forming and combining a learning model.

4.3.4 Generate Bag of Word (BOW) Feature Vector

Bag of Word (BOW) is developed as a baseline for comparing its accuracy to the new model established at the end of this research's tests. BOW is a vector space model used to construct BOW, which is necessary for determining the dataset's frequency characteristics. Datasets must be obtained from UCI's Machine Learning Repository and divided into two sections: one for training and one for testing until BOW is generated. To complete this step, the probability output from the previous phase will be used to construct a new feature vector from the data obtained. According to this study, the direct approach is the term frequency (TF), which relates to the number of regularly occurring terms in a text.

4.3.5 Data Splitting and Validation

The technique of random subsampling was used to split and validate the data in this research. The data set is divided into a user-defined subset by random subsampling. The training set has a specific value, and the model is trained on this data to generalize it to additional data. The test or validation dataset will be used to evaluate the model's prediction for this subset. The YouTube Spam comment collection data is divided 80/20 for this purpose, as seen in Figure 4.11.

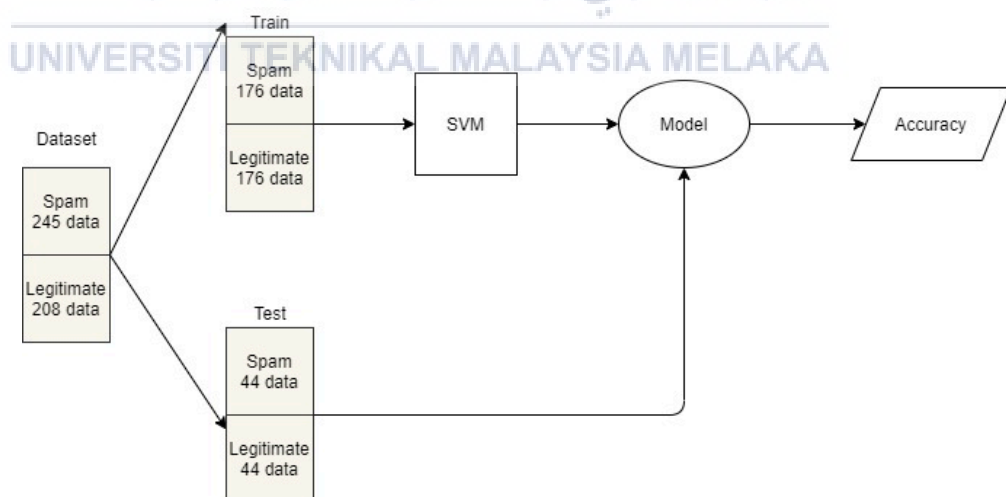


Figure 4.11: Data Train and Test 80/20

Due to the dataset's limited size, it was processed using two different methodologies. Firstly, data must be used in the whole training process to establish the

best classifier and the error rate. Two underlying issues cause this issue: the training data is likely to overfit, and the error rate is too optimistic or too low. As a result, a second solution was designed and implemented for this project. The method involves randomly subsampling the training data to create different subgroups.

As seen in Figure 4.11, there are 453 number of instances and there are 245 instances of spam and 208 of legitimate data. A data set will be splitting into 80/20 method. Following that, 440 random examples are generated for the train and test data using the k-fold 80/20 methodology. Then it reflects the 80% of selected instances, for training which are 176 for spam and another 176 for legitimate comments. Then, it reflects another 20% of selected instances, for testing which are 44 for spam and another 44 for legitimate comments. The train and test data are used to appropriately train the classification algorithm with appropriate instances to distinguish spam and legitimate data, leading to a fair result. Following the training will identify candidates based on variables and validate the most significant accomplishments, referred to as the best kernel. The model will be produced using the SVM method from the data model that has to be trained. As a consequence, the model generated by the SVM approach may give more accurate results.

4.3.6 Feature Selection

Under this section, feature selection eliminates any unwanted, inappropriate, or repeated features from the dataset that do not enhance the model's accuracy but may decrease. The output of train data set information features is then tested using Information Gain (IG) and Chi-Square (CS) throughout the feature selection process. The classification accuracy of the train set is measured as a result.

Information Gain feature selection includes techniques for data preparation such as classification regression and clustering. It also includes techniques for supervised classification mining and visualization. The Attribute Evaluator determines the information gain (entropy) for each attribute of the output variable using the Ranker Search technique. The input values range between 0 and 1. Thus, the more perceptive features would receive a higher value and be chosen, whereas the lesser would get a lower score and be eliminated. To calculate the value of entropy and information gain, the following formula will be used.

Formula of calculating entropy:

$$E = - \sum_i^C P_i \log_2 P_i$$

Where:

E = no. of inputs, C = class, P_i = probability of dataset

Formula of calculating information gain:

$$G = E - \frac{m_L}{m} E_L - \frac{m_R}{m} E_R$$

Where:

m = total no. of instances

The Chi-Square (CS) works by evaluating the categorical coded data obtained with the frequencies expected for the classification variable. It is considered non-significant and eliminated if the feature variable is indeed not reliant on the other variables. The following formula is used to calculate the chi-square value.

$$\chi^2(t, c) = \left[\frac{N \times (AD - BC)^2}{(A + C)(B + C)(A + B)(C + D)} \right]$$

Where:

t = time, c = class, N = total no. of documents

A = no. of t occurrences and c occurrences, D = no. of non-occurrences of t and c

B = no. of t occurrences, C = no. of c occurrences

4.3.7 Normalization

The process of reducing data to a narrower range is known as normalization. In this study, data is scaled from 0 to 1 by normalization. Normalization is primarily essential for classification methods. It modifies the values of the dataset's numeric columns to use a more precise and standard scale while preserving the range of values. It converts the dataset into a more readable format.

4.3.8 Classification

In supervised learning, the Support Vector Machine (SVM) classifies linear and nonlinear data by maximizing the margin between support points and using a nonlinear mapping to convert the training set data into a higher dimension (Berk, 2020). In the project classifier's classification phase, this SVM is used to evaluate spam and legitimate comments. SVM represents each data item as an n-dimensional space point ($n = \text{number of features}$), the data contains two features, x and y , with two tags, red and blue. This research aimed at a classifier that, given a set of (x,y) coordinates, produces whether it is red or blue. Then, on a plane, plot the training data set as shown in Figure 4.12.

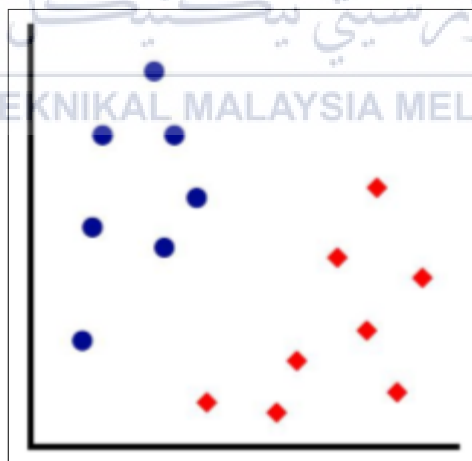


Figure 4.12: SVM Graph

Source: SVM classifier,2013

The SVM classifier will illustrate how to plot in the graph after a plot has been identified. With plotting using a two-part classifier, the first section being train data,

the second section being test data. By separating the given feature pattern sequence, the classifier places the hyperplane. Figure 4.13 shows the classification of hyperplanes which is the black line that separate tags blue and class to differentiate between two groups of SVM.

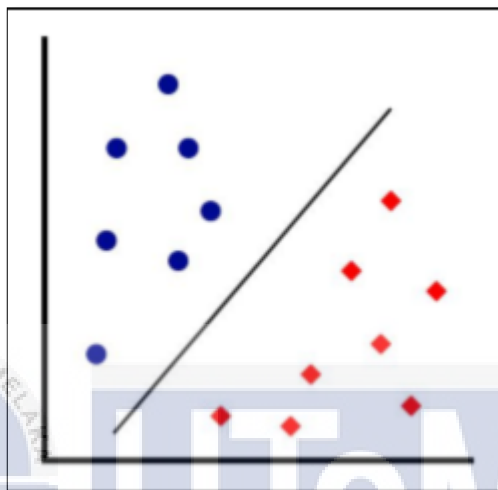


Figure 4.13: Hyperplane Placement

Source: SVM classifier, 2013

This segmentation is the only technique to overcome the drawbacks of classification accuracy. The accurate and inaccurate predictions are then combined into a matrix expected to fall from each row. A genuine distribution is seen in Figure 4.16. The confusion matrix combines four diverse features and values of prediction, true positive (TP), true negatives (TN), false positives (FP) and false negatives (FN). True positive is expected to be positive and true, whereas false positive is expected to be positive, though it was false. Because it has been expected that genuine negative interpretation is negative, but the false negative interpretation is negative and true.

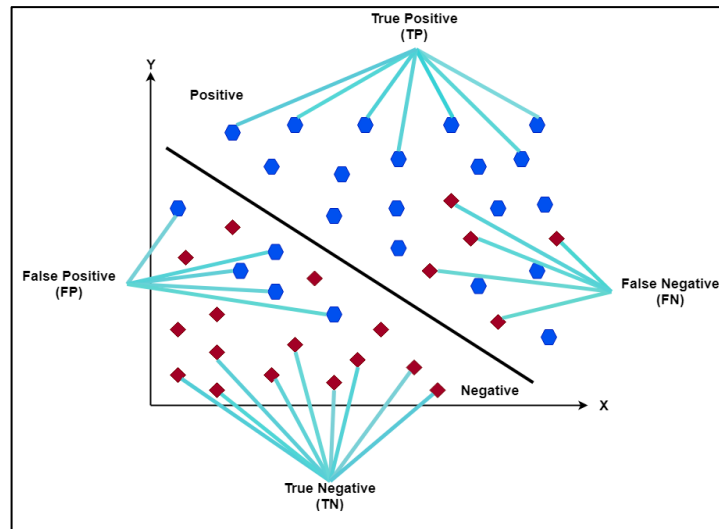


Figure 4.14: Real Distribution Data Example

Source: Support Vector Machine (SVM) algorithm,2017

The above scenario is known as the Linear SVM, with a line dividing vectors, and most real-life situations are generally non-linear, as shown in Figure 4.15.

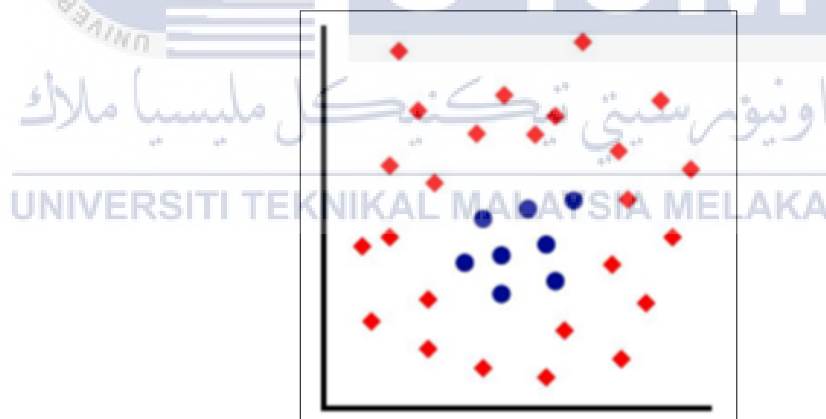


Figure 4.15: Non-Linear Graph

Source: The SVM classifier,2013

To address this problem, we must use the kernel function that is provided by SVM. When applied, the kernel will convert a non-linear space into a linear one. The kernel function applies data to each instance that may be separated using the data separator. Two-variable data, represented by the variables x and y , are transformed into multiple new function spaces, represented by the letter z . As seen in Figure 4.16,

the two-dimensional region was converted into a multidimensional space via the transformation process.

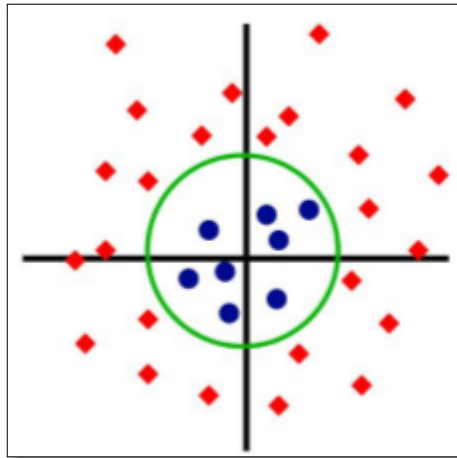


Figure 4.16: SVM with Multidimensional Space

Source: The SVM classifier, 2013

Table 4.3 presents the kernel types, each with its unique set of functions. The accuracy of its hyperplanes must be compared to all of its tests to determine which kernel is the best. High accuracy is chosen and applied throughout this project. When the system is detected, the new model relies on accuracy identification. Hence, compared to several other accuracies generated by using BOW, IG, and CS to guarantee that this model recognises YouTube spam and legitimate comments more accurately.

Table 4.C: Type of SVM Kernel

Kernel	Description
Linear	A straight line is used to separate the vectors linearly.
Polynomial	It performs well with data that is not linearly separable.

Radial Basis Function (RBF)	Non-linear data may be classified using this method. The Artificial Neural Network (ANN) may use a radial basis function to activate a function.
Sigmoid	It originally comes from neural networks, a less time-consuming way of categorization.

4.3.9 Post-Classification

Post classification refers to filtering noise and enhancing the quality of the classified output. The ensemble method is used to generate the prediction file from the classification (SVM) section. The ensemble method is implemented through the combining classifier to create a more robust classifier. The prediction file is trained using an algorithm, and predictions are made using a combination classifier throughout this procedure. This Learning algorithms method aims to decrease overall error by combining predictions from a range of different classifiers to get final results that are better than using the individual classifier and wants to find out how the use of ensemble method's ability to detect spam by testing the performance of an ensemble method that combines n-gram features with various classifiers and experiment with both weighted and non-weighted ensembles.

4.4 Conclusion

In summary, this chapter details the findings of this study and all of its methodologies. The analysis procedure helps in identifying and decomposing the research methodology's components. It is important to ensure that the planned work is successful in attaining the research's objectives. All of the detailed design included in this chapter is necessary to guarantee that existing projects function smoothly. All of this is accomplished by reviewing the research literature, which serves as a guide for achieving the objectives. SVM is used to construct a model, which is then compared to detection accuracy against IG and Chi-Square feature selection approaches. The next chapter will discuss the implementation phase, which must be carried out following the analysis and design achieved in this chapter.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

This chapter covers the implementation process, which will be completed by the analysis and design of the project mentioned in the previous chapter. The experiment consists of many steps, which are discussed in this chapter. This analysis concludes with a prediction about whether the comments are spam or legitimate. Also, this study was employed to ensure that researcher achieved the project's goals of resolving the problem statements.

5.2 Software Development Environment Setup

Throughout this implementation process, the windows operating system will be used as the platform for this project. Therefore, the Windows operating system has been installed, providing an optimal computational capability for running the project's other programs and scripts. In addition, Eclipse IDE 2021-06 was selected as a platform for system development and support, together with the Java Development Kit (JDK) as a compiler package, which performs building, executing, and managing all functions of Java-writing applications.

Then there's Gnuplot, which is a portable command-line controlled graphing application. It was initially developed to enable researchers to interact with mathematical functions and data to understand them better. Last but not least, Python is a programming language that places a strong emphasis on code readability via the extensive usage of whitespace. As a result, programmers may create clear, logical code for both small and large-scale projects with the language features and object-oriented approach provided by the language.

5.3 Process Module

The primary module is divided into many tasks that the researcher must complete for the experiment to be successful. For example, the processing module is shown in Figure 5.1.



Figure 5.1: Process Module

5.3.1 Collection of Dataset

The datasets collection module retrieves data from the UCI Machine Learning Repositories. It consists of five datasets, as described in the preceding chapter. The datasets are classified as spam and legitimate. The purpose of this experiment is to assess the accuracy of spam using a machine learning technique. This project uses the Classifiers to generate False Positive and False Negative values.

5.3.2 Data Preprocessing

Five stages were involved in the data preprocessing, as described in the previous chapter. First, follow the steps shown in figure 5.2.



Figure 5.2: Data Preprocessing Module

5.3.2.1 Eminem Dataset Preprocessing

From step 1 to step 5, as shown in Table 5.1, this part will explain how the researcher preprocessed the results of dataset Eminem and the parameters used with the scripts. Then, following the data preprocessing module steps, this part represents how Eminem's dataset's pre-processing is carried out.

a) Sorted

The class dataset gives a value of 0 for legitimate and 1 for spam. As shown in Figure 5.3, we sorted the dataset into legitimate and spam categories. Lines 2 to 246 are considered spam, whereas lines 247 to 454 are considered legitimate.

1	author	date	content	class
2	0	1	hey guys im a 17yr old rapper trying to get exposure... i live in belgium	1
246	192	0	3 yrs ago i had a health scare but thankfully im okay. i realized i wasnt	1
247	193	1	i always end up coming back to this song 	0
454	253	0	this great warning will happen soon.	0

Figure 5.3: Sort Dataset Eminem

b) Content

After sorting, delete all the headers and the date column. The spam begins at line 1 until 245 and the legitimate starts at line 246 until 453. Figure 5.4 illustrates how the author, content and class in this file will proceed.

1	0	hey guys im a 17yr old rapper trying to get exposure... i live in belgium	1
245	192	3 yrs ago i had a health scare but thankfully im okay. i realized i wasnt	1
246	193	i always end up coming back to this song 	0
453	253	this great warning will happen soon.	0

Figure 5.4: Content Dataset Eminem

c) Remove Special Character

i. Non-ASCII Character

The non-ascii character is manually removed using Notepad++. They were replaced with nothing as it will be removed by using the formula “[^x00-x7F]+” as shown in figure 5.5.

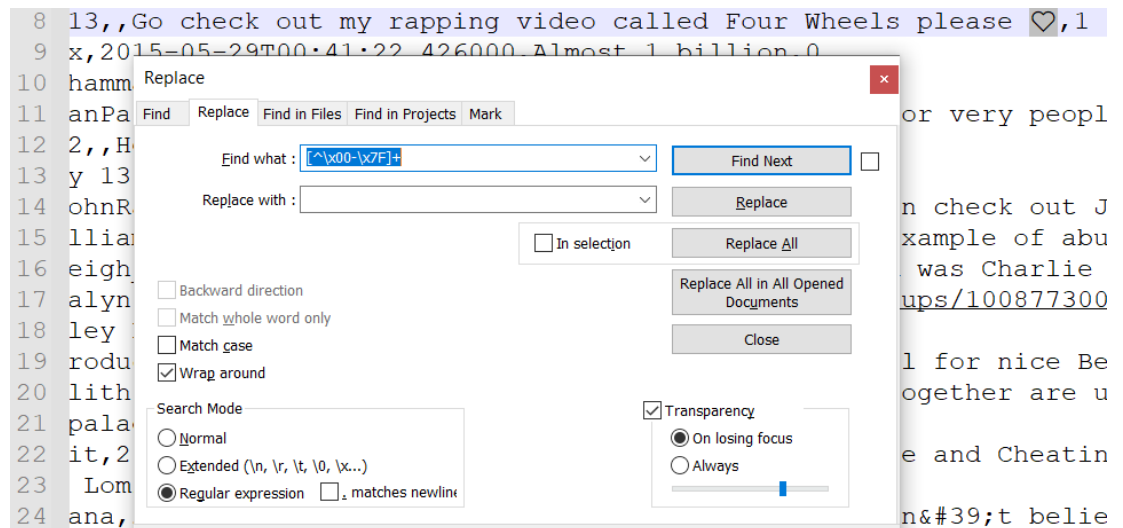


Figure 5.5: Remove Non-ASCII Character

ii. No Author

Those comments that do not have an author may be changed manually by using Notepad++, and they were replaced with the phrase “no author” since it will be deleted by using the characters “,” or “,” as shown in figure 5.6. Meanwhile, if a comment does not include a date, it is substituted with 0, and the comment will be deleted using the symbol “,” as illustrated in figure 5.7. In addition, it must be in regular expression search mode to function correctly.

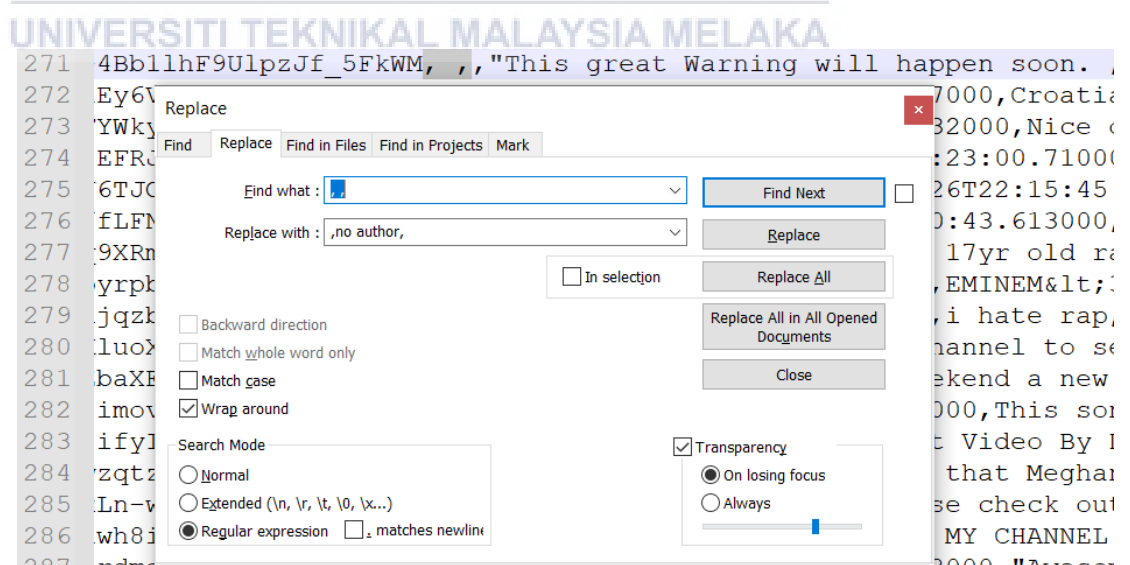


Figure 5.6: No Author

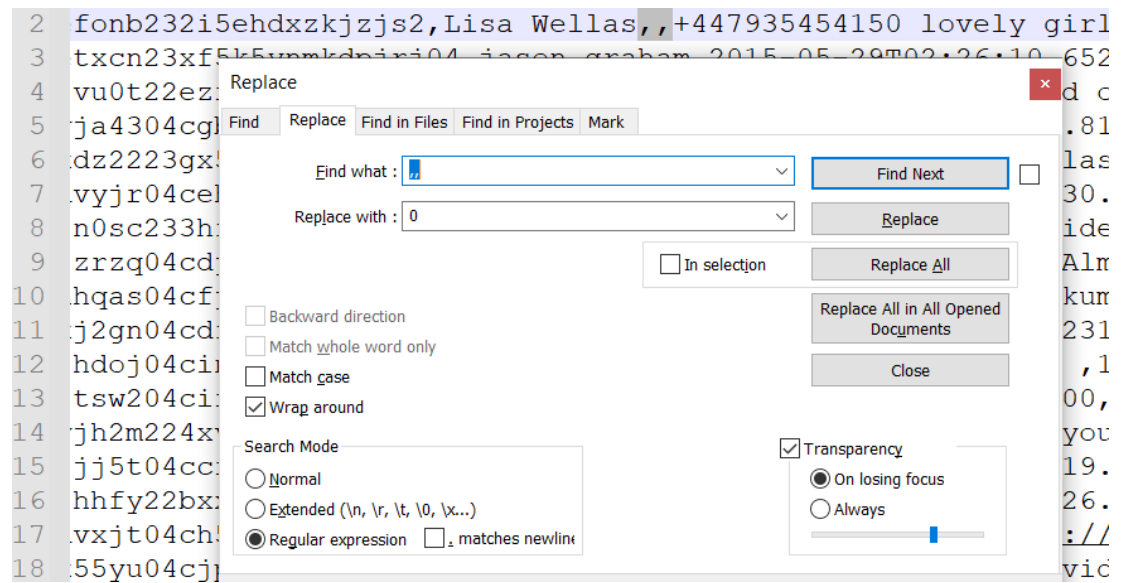


Figure 5.7: No Date

iii. Convert To Lowercase

The uppercase character is manually converted to lowercase using Notepad++. They are replaced with “\L\$0” as it will be replaced by using the formula “(?-s).+” as shown in figure 5.8.

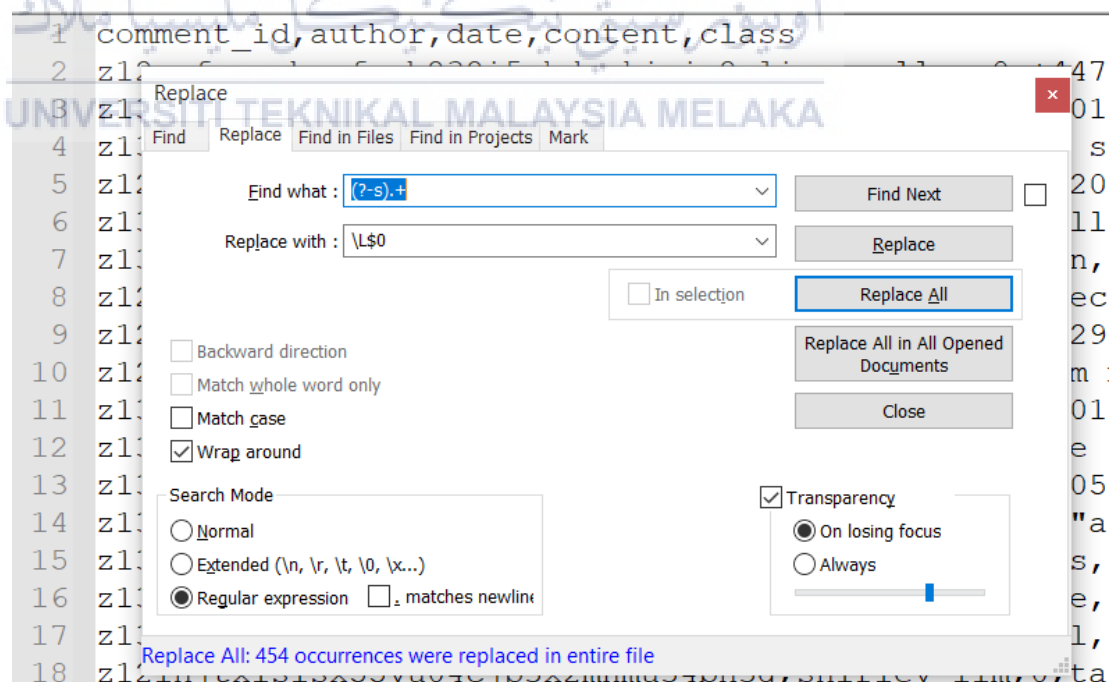


Figure 5.8: Convert To Lowercase

d) Preprocessing Execution Script

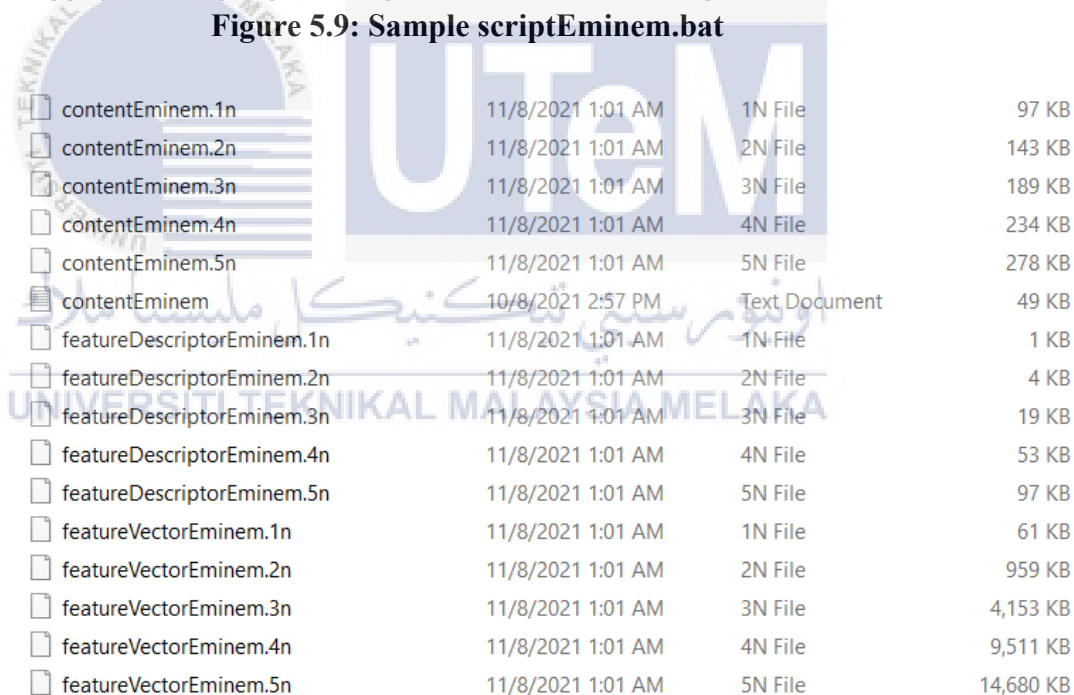
The next step is to execute the script using a batch file to prepare the Eminem dataset, the script running in the Command Prompt. As shown in Figure 5.9, the example of the batch file. Execution may require some time process running for 1n to 5n. Once executed, the program creates multiple files by the script included in the batch file. The file created as a result of running the batch file is shown in Figure 5.10.

```

java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 1 D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 2 D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 3 D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 4 D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 5 D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.1n I
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.2n I
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.3n I
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.4n I
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.5n I
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureDescriptorEminem
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureDescriptorEminem
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureDescriptorEminem
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVectorEminem.1n D:\PSM2\
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVectorEminem.2n D:\PSM2\
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVectorEminem.3n D:\PSM2\
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVectorEminem.4n D:\PSM2\
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVectorEminem.5n D:\PSM2\

```

Figure 5.9: Sample scriptEminem.bat



File Name	Date Modified	Type	Size
contentEminem.1n	11/8/2021 1:01 AM	1N File	97 KB
contentEminem.2n	11/8/2021 1:01 AM	2N File	143 KB
contentEminem.3n	11/8/2021 1:01 AM	3N File	189 KB
contentEminem.4n	11/8/2021 1:01 AM	4N File	234 KB
contentEminem.5n	11/8/2021 1:01 AM	5N File	278 KB
contentEminem	10/8/2021 2:57 PM	Text Document	49 KB
featureDescriptorEminem.1n	11/8/2021 1:01 AM	1N File	1 KB
featureDescriptorEminem.2n	11/8/2021 1:01 AM	2N File	4 KB
featureDescriptorEminem.3n	11/8/2021 1:01 AM	3N File	19 KB
featureDescriptorEminem.4n	11/8/2021 1:01 AM	4N File	53 KB
featureDescriptorEminem.5n	11/8/2021 1:01 AM	5N File	97 KB
featureVectorEminem.1n	11/8/2021 1:01 AM	1N File	61 KB
featureVectorEminem.2n	11/8/2021 1:01 AM	2N File	959 KB
featureVectorEminem.3n	11/8/2021 1:01 AM	3N File	4,153 KB
featureVectorEminem.4n	11/8/2021 1:01 AM	4N File	9,511 KB
featureVectorEminem.5n	11/8/2021 1:01 AM	5N File	14,680 KB

Figure 5.10: Sample File Created for Eminem

5.3.2.2 Psy Dataset Preprocessing

This section will explain how the Psy dataset is preprocessing using the data preprocessing module's steps.

a) Sorted

The class dataset gives a value of 0 for legitimate and 1 for spam. As shown in Figure 5.11, we sorted the dataset into legitimate and spam categories. Lines 2 to 176 are considered spam, whereas lines 177 to 351 are considered legitimate.

1	author	date	content	class
2	julius nm	0	huh, anyway check out this you[tube] channel:	1
176	photo edit	0	hi guys please my android photo editor downlo	1
177	bob kanow	0	i turned it on mute as soon is i came on i just w	0
351	ray benich	0	the first billion viewed this because they thoug!	0

Figure 5.11: Sort Dataset Psy

b) Content

After sorting, delete all the headers and the date column. The spam begins at line 1 until 175 and the legitimate starts at line 176 until 350. Figure 5.12 illustrates how the author, content and class in this file will proceed.

1	julius nm	huh, anyway check out this you[tube] channel:	1
175	photo edit	hi guys please my android photo editor downlo	1
176	bob kanow	i turned it on mute as soon is i came on i just w	0
350	ray benich	the first billion viewed this because they thoug!	0

Figure 5.12: Content Psy Dataset

c) Remove Special Character

i. Non-ASCII Character

They were replaced with nothing as it will be removed by using the formula “[^x00-x7F]+” as shown in figure 5.13.

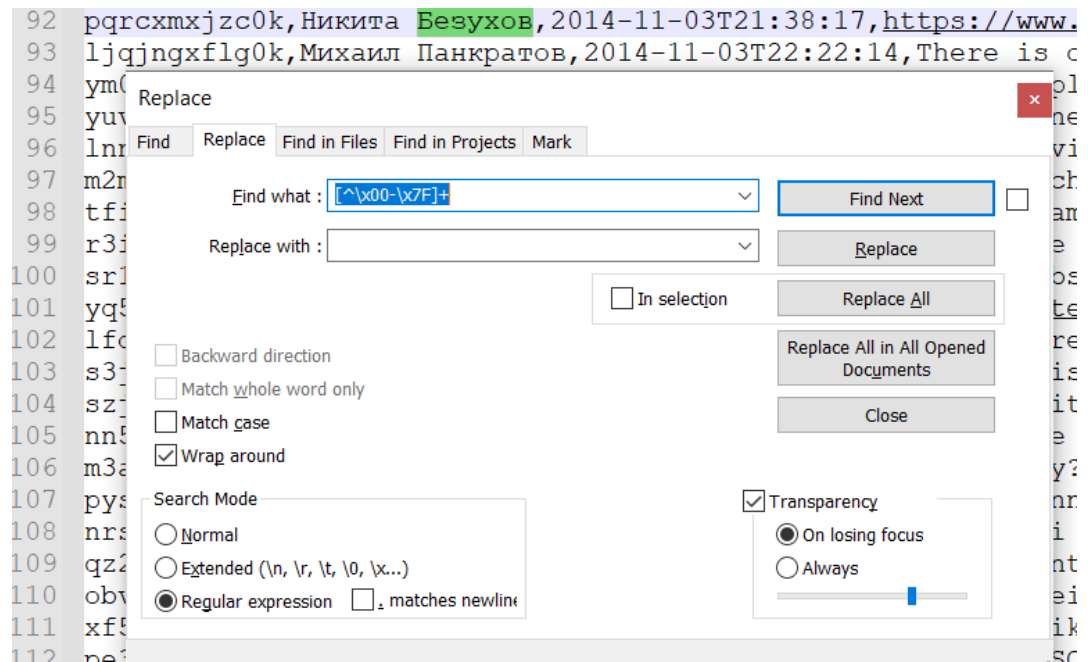


Figure 5.13: Remove Non-ASCII Character

ii. No Author and No date

Those comments that do not have an was replaced with the phrase “no author” since it will be deleted by using the characters “,” or “,” as shown in figure 5.14.

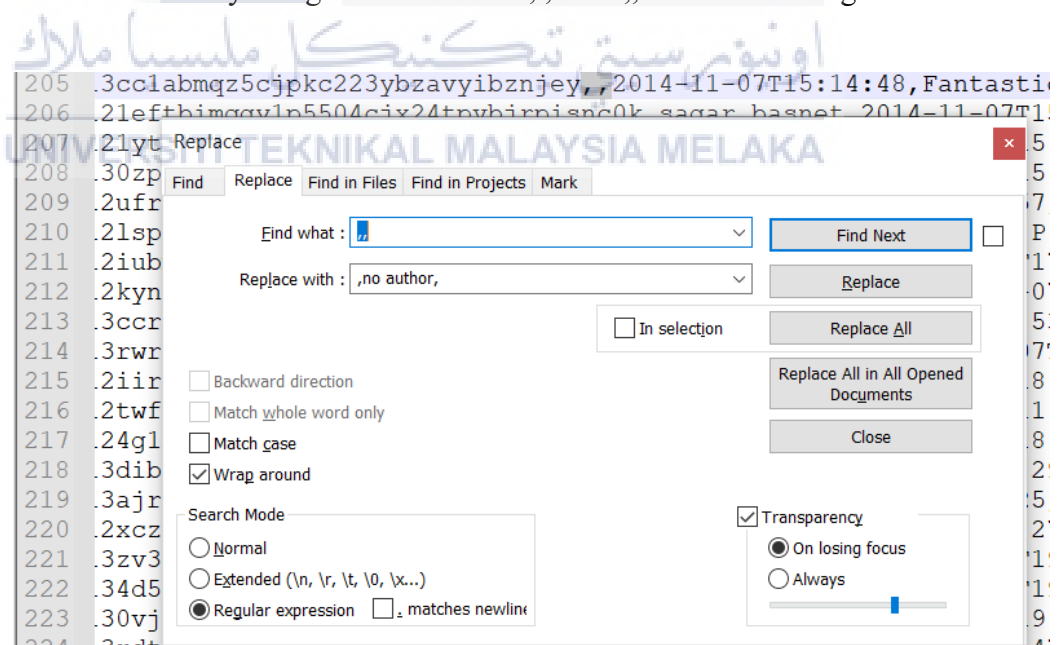


Figure 5.14: No Author

iii. Convert To Lowercase

They are replaced with “\L\$0” as it will be replaced by using the formula “(?-s).+” as shown in figure 5.15.

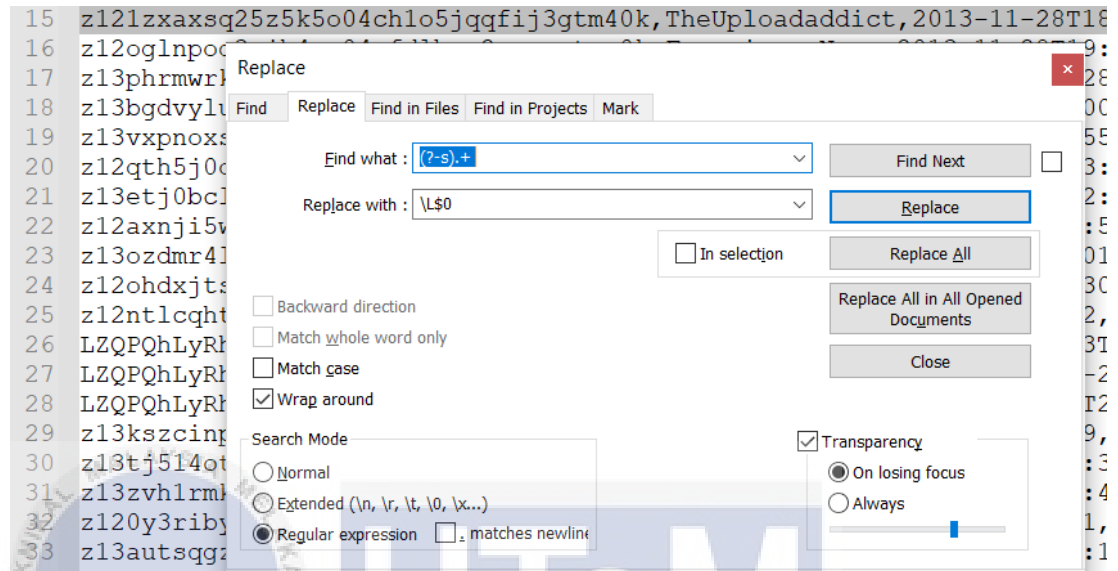


Figure 5.15: Convert To Lowercase

d) Preprocessing Execution Script

The next step is to execute the script using a batch file to prepare the Psy dataset, the script running in the Command Prompt. As shown in Figure 5.16, the example of the batch file. Execution may require some time process running for 1n to 5n. Once executed, the program creates multiple files by the script included in the batch file. The file created as a result of running the batch file is shown in Figure 5.17.

```

java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 1 D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 2 D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 3 D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 4 D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 5 D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.1n D:
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.2n D:
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.3n D:
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.4n D:
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.5n D:
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureDescriptorPsy
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureDescriptorPsy
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureDescriptorPsy
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureDescriptorPsy
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureDescriptorPsy
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy.1n D:\PSM2\Yc
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy.2n D:\PSM2\Yc
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy.3n D:\PSM2\Yc
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy.4n D:\PSM2\Yc
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy.5n D:\PSM2\Yc

```

Figure 5.16: Sample scriptPsy.bat

contentPsy.1n	11/8/2021 1:01 AM	1N File	59 KB
contentPsy.2n	11/8/2021 1:01 AM	2N File	87 KB
contentPsy.3n	11/8/2021 1:01 AM	3N File	115 KB
contentPsy.4n	11/8/2021 1:01 AM	4N File	142 KB
contentPsy.5n	11/8/2021 1:01 AM	5N File	168 KB
contentPsy	10/8/2021 3:15 PM	Text Document	30 KB
featureDescriptorPsy.1n	11/8/2021 1:01 AM	1N File	1 KB
featureDescriptorPsy.2n	11/8/2021 1:01 AM	2N File	5 KB
featureDescriptorPsy.3n	11/8/2021 1:01 AM	3N File	22 KB
featureDescriptorPsy.4n	11/8/2021 1:01 AM	4N File	52 KB
featureDescriptorPsy.5n	11/8/2021 1:01 AM	5N File	87 KB
featureVectorPsy.1n	11/8/2021 1:01 AM	1N File	45 KB
featureVectorPsy.2n	11/8/2021 1:01 AM	2N File	949 KB
featureVectorPsy.3n	11/8/2021 1:01 AM	3N File	3,800 KB
featureVectorPsy.4n	11/8/2021 1:01 AM	4N File	7,305 KB

Figure 5.17: Sample File Created for Psy

5.3.2.3 Shakira Dataset Preprocessing

This section will explain how the Psy dataset is preprocessing using the data preprocessing module's steps.

a) Sorted

The class dataset gives a value of 0 for legitimate and 1 for spam. As shown in Figure 5.18, we sorted the dataset into legitimate and spam categories. Lines 2 to 175 are considered spam, whereas lines 176 to 371 are considered legitimate.

1	author	content	class
2	0	see some more song open google and type shakira guruofmovie	1
175	133	**check out my new mixtape**** **check out my new mixtape***	1
176	134	nice song	0
371	311	shakira is the best dancer	0

Figure 5.18: Sort Dataset Shakira

b) Content

After sorting, delete all the headers and the date column. The spam begins at line 1 until 174 and the legitimate starts at line 175 until 370. Figure 5.19 illustrates how the author, content and class in this file will proceed.

1	0	see some more song open google and type shakira guruofmovie	1
174	133	**check out my new mixtape**** **check out my new mixtape***	1
175	134	nice song	0
370	311	shakira is the best dancer	0

Figure 5.19: Content Shakira Dataset

c) Remove Special Character

i. Non-ASCII Character

They were replaced with nothing as it will be removed by using the formula “[^x00-x7F]+” as shown in figure 5.20.

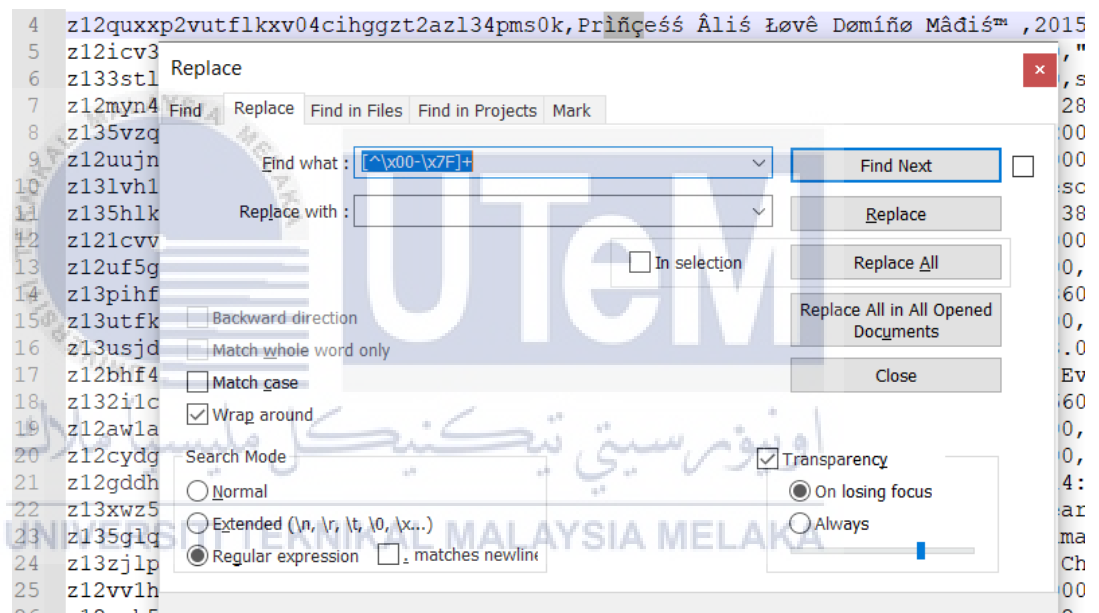


Figure 5.20: Remove Non-ASCII Character

ii. No Author

Those comments that do not have an author were replaced with the phrase “no author” since it will be deleted by using the characters “,” or “,” as shown in figure 5.21.

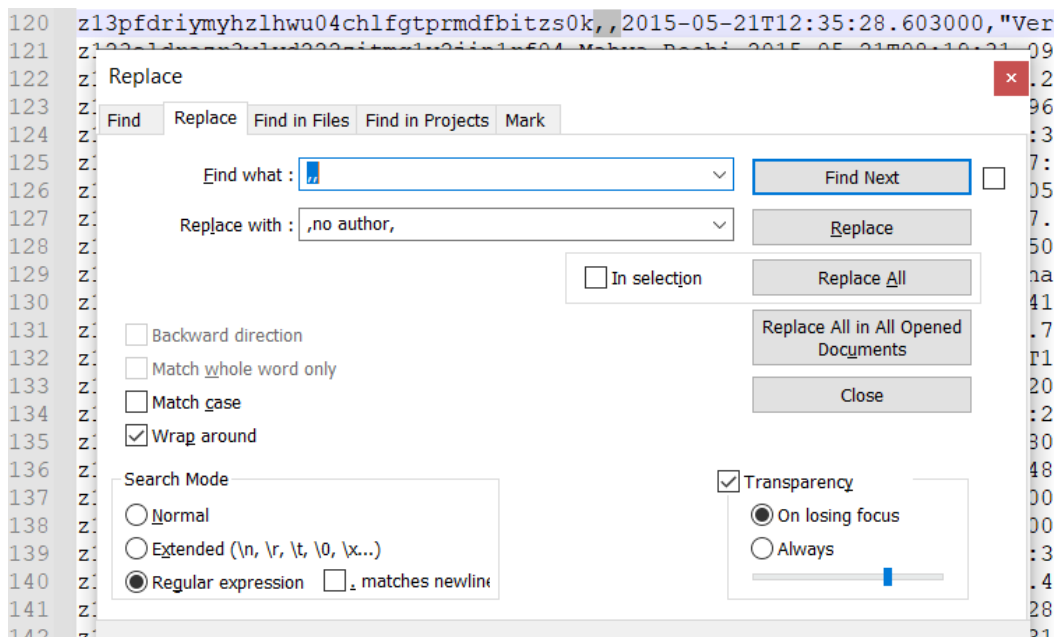


Figure 5.21: No Author

iii. **Convert To Lowercase**

They are replaced with “\L\$0” as it will be replaced by using the formula “(?-s).+” as shown in figure 5.22.

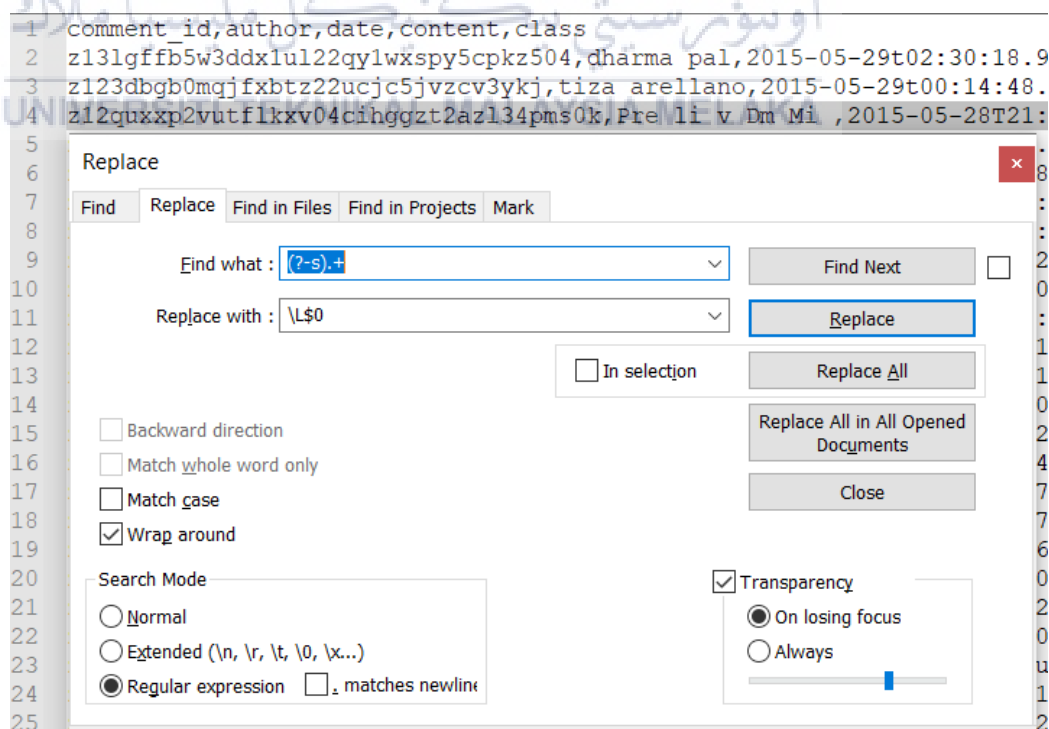


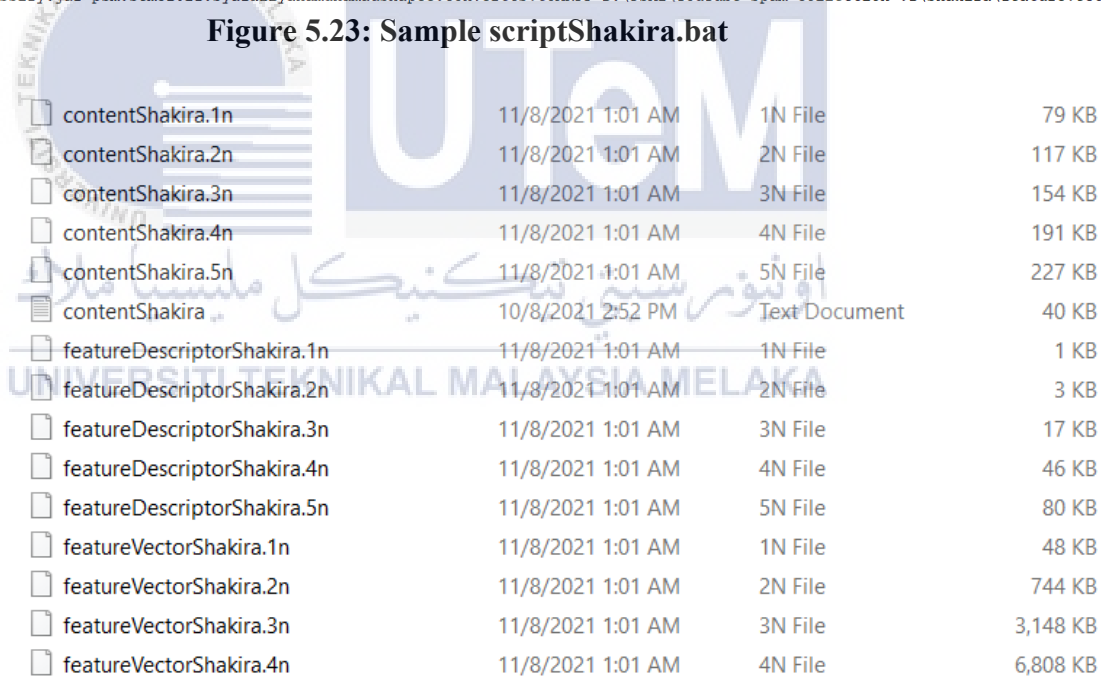
Figure 5.22: Convert To Lowercase

d) Preprocessing Execution Script

The next step is to execute the script using a batch file to prepare the Psy dataset, the script running in the Command Prompt. As shown in Figure 5.23, the example of the batch file. Execution may require some time process running for 1n to 5n. Once executed, the program creates multiple files by the script included in the batch file. The file created as a result of running the batch file is shown in Figure 5.24.

```
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 1 D:\PSM2\YouTube-Spam-Collection-v1\Shakira\contentShakira.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 2 D:\PSM2\YouTube-Spam-Collection-v1\Shakira\contentShakira.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 3 D:\PSM2\YouTube-Spam-Collection-v1\Shakira\contentShakira.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 4 D:\PSM2\YouTube-Spam-Collection-v1\Shakira\contentShakira.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 5 D:\PSM2\YouTube-Spam-Collection-v1\Shakira\contentShakira.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\Shakira\cont
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\Shakira\cont
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\Shakira\cont
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\Shakira\cont
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureD
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureD
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureD
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureD
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShaki
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShaki
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShaki
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShaki
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShaki
```

Figure 5.23: Sample scriptShakira.bat



File Name	Date/Time	Type	Size
contentShakira.1n	11/8/2021 1:01 AM	1N File	79 KB
contentShakira.2n	11/8/2021 1:01 AM	2N File	117 KB
contentShakira.3n	11/8/2021 1:01 AM	3N File	154 KB
contentShakira.4n	11/8/2021 1:01 AM	4N File	191 KB
contentShakira.5n	11/8/2021 1:01 AM	5N File	227 KB
contentShakira	10/8/2021 2:52 PM	Text Document	40 KB
featureDescriptorShakira.1n	11/8/2021 1:01 AM	1N File	1 KB
featureDescriptorShakira.2n	11/8/2021 1:01 AM	2N File	3 KB
featureDescriptorShakira.3n	11/8/2021 1:01 AM	3N File	17 KB
featureDescriptorShakira.4n	11/8/2021 1:01 AM	4N File	46 KB
featureDescriptorShakira.5n	11/8/2021 1:01 AM	5N File	80 KB
featureVectorShakira.1n	11/8/2021 1:01 AM	1N File	48 KB
featureVectorShakira.2n	11/8/2021 1:01 AM	2N File	744 KB
featureVectorShakira.3n	11/8/2021 1:01 AM	3N File	3,148 KB
featureVectorShakira.4n	11/8/2021 1:01 AM	4N File	6,808 KB

Figure 5.24: Sample File Created for Shakira

5.3.2.4 LMFAO Dataset Preprocessing

This section will explain how the LMFAO dataset is preprocessing using the data preprocessing module's steps.

a) Sorted

The class dataset gives a value of 0 for legitimate and 1 for spam. As shown in Figure 5.25, we sorted the dataset into legitimate and spam categories. Lines 2 to 237 are considered spam, whereas lines 238 to 439 are considered legitimate.

1	author	content	class
2	0	2:19	1
237	223	check out this video on youtube:	1
238	19	check out this playlist on youtube: 	0
439	407	nice :3	0

Figure 5.25: Sort LMFAO Dataset

b) Content

After sorting, delete all the headers and the date column. The spam begins at line 1 until 236 and the legitimate starts at line 237 until 438. Figure 5.26 illustrates how the author, content and class in this file will proceed.

1	0	2:19	1
236	223	check out this video on youtube:	1
237	19	check out this playlist on youtube: 	0
438	407	nice :3	0

Figure 5.26: Content LMFAO Dataset

c) Remove Special Character

i. Non-ASCII Character

They were replaced with nothing as it will be removed by using the formula “[^x00-x7F]+” as shown in figure 5.27.

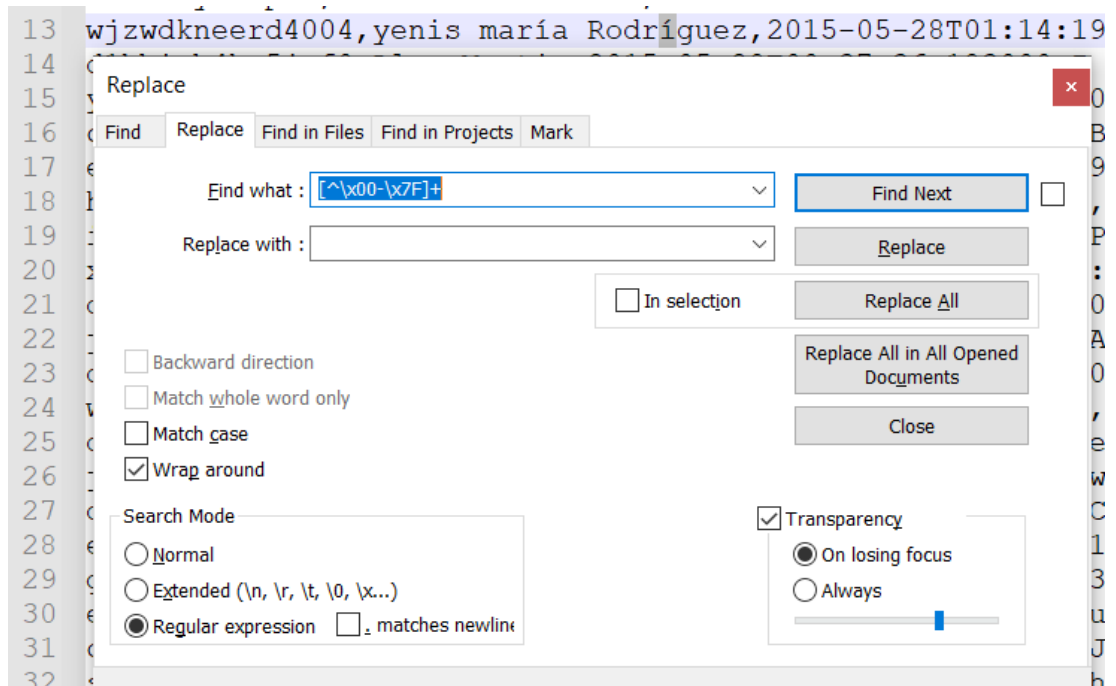


Figure 5.27: Remove Non-ACII Character

ii. No Author

Those comments that do not have an was replaced with the phrase “no author” since it will be deleted by using the characters “,” or “,” as shown in figure 5.28.

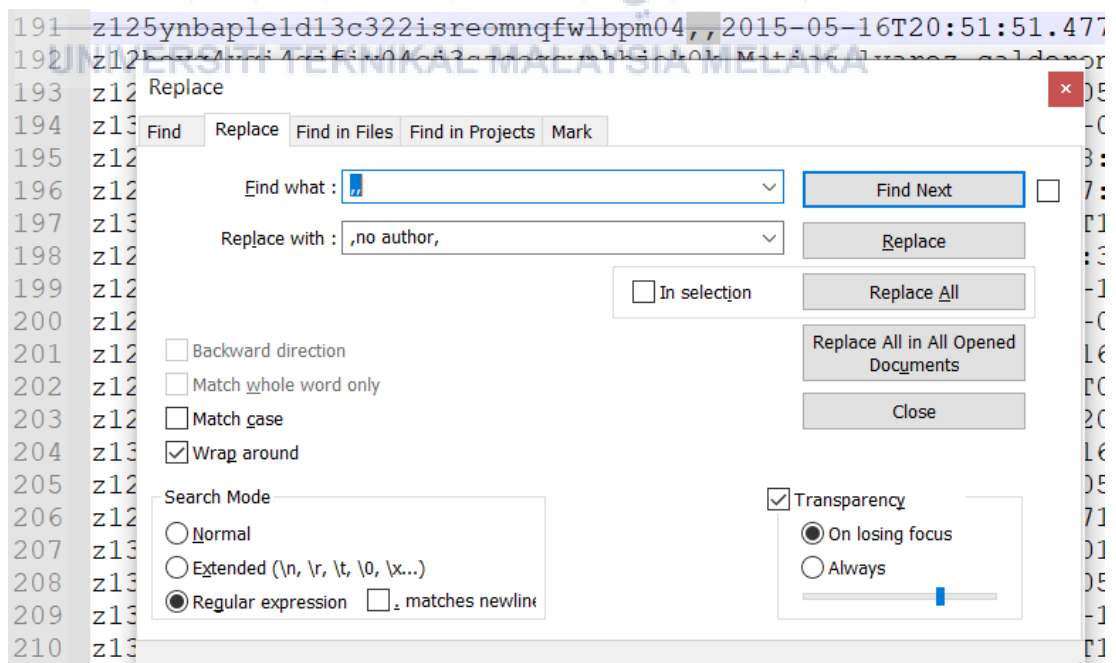


Figure 5.28: No Author

iii. Convert To Lowercase

They are replaced with “\L\$0” as it will be replaced by using the formula “(?-s).+” as shown in figure 5.29.

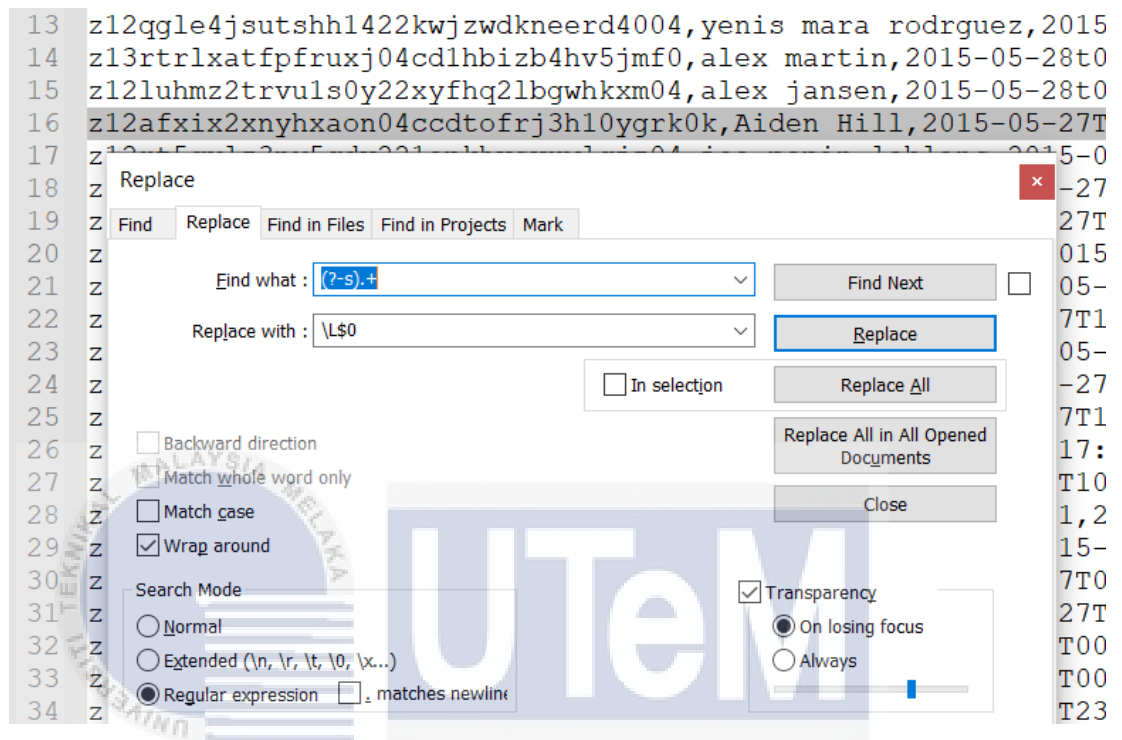


Figure 5.29: Convert To Lowercase

d) Preprocessing Execution Script

The next step is to execute the script using a batch file to prepare the Psy dataset, the script running in the Command Prompt. As shown in Figure 5.30, the example of the batch file. Execution may require some time process running for 1n to 5n. Once executed, the program creates multiple files by the script included in the batch file. The file created as a result of running the batch file is shown in Figure 5.31.

```

java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 1 D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\contentLmfao.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 2 D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\contentLmfao.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 3 D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\contentLmfao.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 4 D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\contentLmfao.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 5 D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\contentLmfao.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\co
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\co
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\co
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\co
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\co
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featur
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featur
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featur
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featur
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featur
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmf
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmf
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmf
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmf
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmf

```

Figure 5.30: Sample scriptLMFAO.bat

contentLmfao.1n	10/8/2021 4:17 PM	1N File	51 KB
contentLmfao.2n	10/8/2021 4:17 PM	2N File	75 KB
contentLmfao.3n	10/8/2021 4:17 PM	3N File	97 KB
contentLmfao.4n	10/8/2021 4:17 PM	4N File	119 KB
contentLmfao.5n	10/8/2021 4:17 PM	5N File	140 KB
contentLmfao	10/8/2021 3:04 PM	Text Document	26 KB
featureDescriptorLmfao.1n	11/8/2021 1:01 AM	1N File	1 KB
featureDescriptorLmfao.2n	11/8/2021 1:01 AM	2N File	3 KB
featureDescriptorLmfao.3n	11/8/2021 1:01 AM	3N File	14 KB
featureDescriptorLmfao.4n	11/8/2021 1:01 AM	4N File	33 KB
featureDescriptorLmfao.5n	11/8/2021 1:01 AM	5N File	54 KB
featureVectorLmfao.1n	11/8/2021 1:01 AM	1N File	55 KB
featureVectorLmfao.2n	11/8/2021 1:01 AM	2N File	813 KB
featureVectorLmfao.3n	11/8/2021 1:01 AM	3N File	2,967 KB
featureVectorLmfao.4n	11/8/2021 1:01 AM	4N File	5,701 KB
featureVectorLmfao.5n	11/8/2021 1:01 AM	5N File	7,826 KB

Figure 5.31: Sample File Created for LMFAO

5.3.2.5 Katy Perry Dataset Preprocessing

This section will explain how the Katy Perry dataset is preprocessing using the data preprocessing module's steps.

a) Sorted

The class dataset gives a value of 0 for legitimate and 1 for spam. As shown in Figure 5.32, we sorted the dataset into legitimate and spam categories. Lines 2 to 176 are considered spam, whereas lines 177 to 351 are considered legitimate.

1	author	content	class
2	0	i love this so much. and also i generate free leads on auto pilot & you car	1
176	168	katy perry, i am the "dcio cabelo", "decio hair". i am 60 years of age. i don't h	1
177	169	katy perry does remind me of a tiger,like as if its her spirit animal :3 &3	0
351	339	who is going to reach the billion first : katy or taylor ?	0

Figure 5.32: Sort Katy Perry Dataset

b) Content

After sorting, delete all the headers and the date column. The spam begins at line 1 until 175 and the legitimate starts at line 176 until 350. Figure 5.33 illustrates how the author, content and class in this file will proceed.

1	0	i love this so much. and also i generate free leads on auto pilot & you car	1
175	168	katy perry, i am the "dcio cabelo", "decio hair". i am 60 years of age. i don't h	1
176	169	katy perry does remind me of a tiger,like as if its her spirit animal :3 &3	0
350	339	who is going to reach the billion first : katy or taylor ?	0

Figure 5.33: Content Katy Perry Dataset

c) Remove Special Character

i. Non-ASCII Character

They were replaced with nothing as it will be removed by using the formula “[^x00-x7F]+” as shown in figure 5.34.

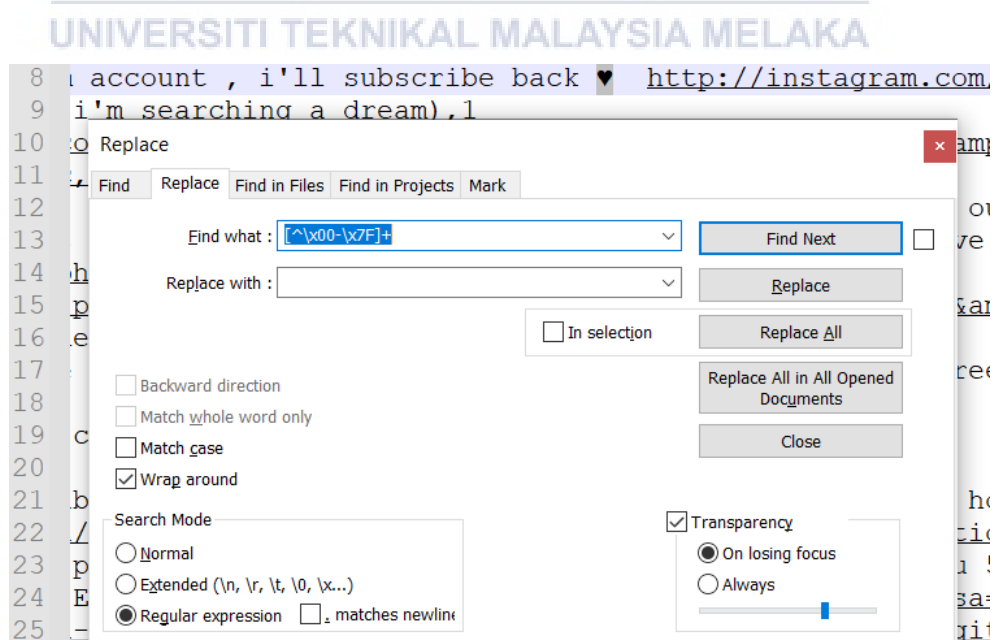


Figure 5.34: Remove Non-ASCII Character

ii. No Author

Those comments that do not have an author were replaced with the phrase “no author” since it will be deleted by using the characters “,” or “,” as shown in figure 5.35.

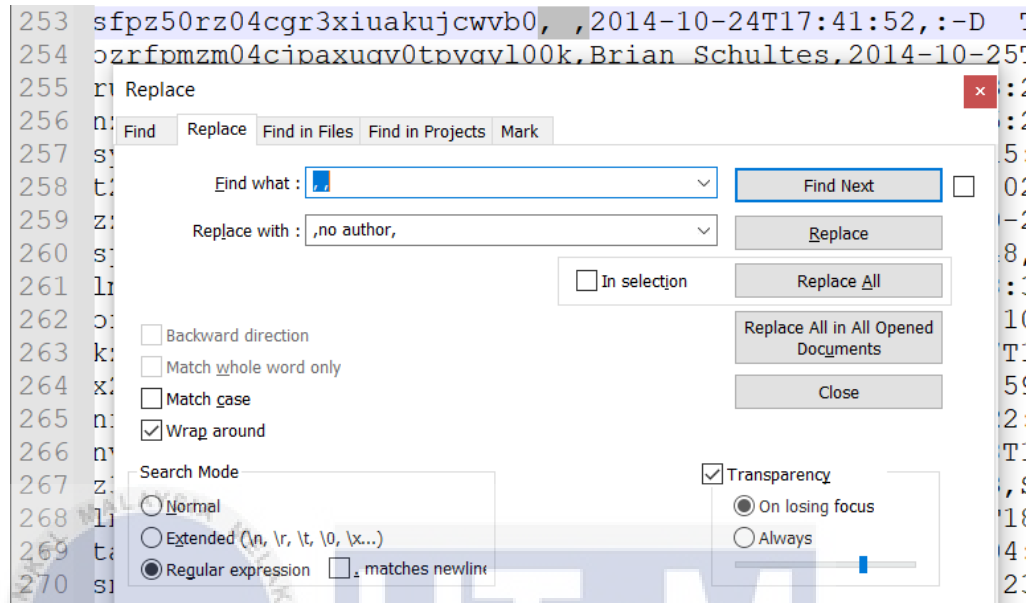


Figure 5.35: No Author

iii. Convert To Lowercase

They are replaced with “\L\$0” as it will be replaced by using the formula “(?-s).+” as shown in figure 5.36.

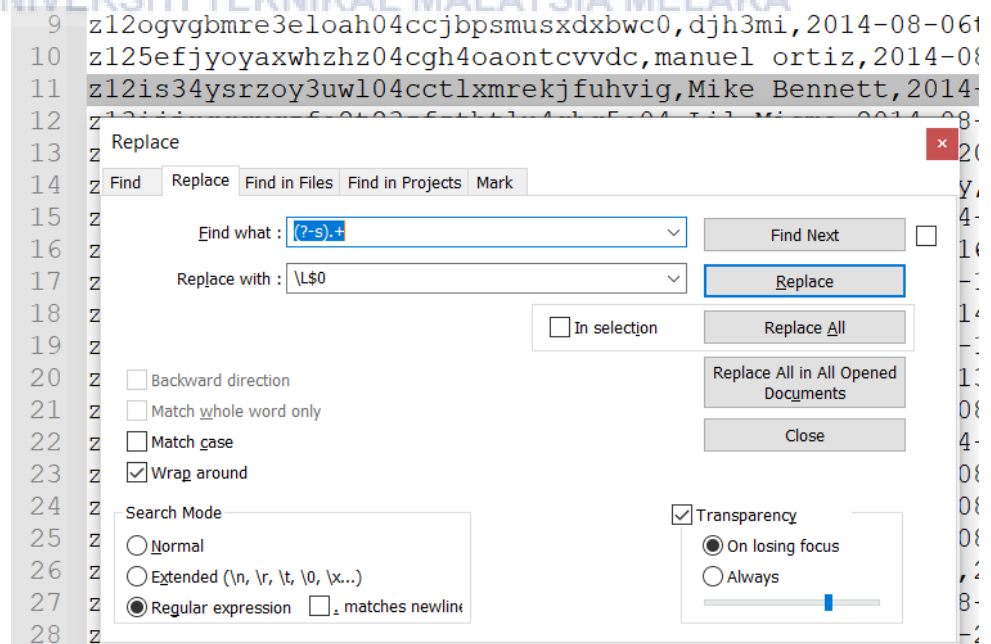


Figure 5.36: Convert To Lowercase

d) Preprocessing Execution Script

The next step is to execute the script using a batch file to prepare the Psy dataset, the script running in the Command Prompt. As shown in Figure 5.37, the example of the batch file. Execution may require some time process running for 1n to 5n. Once executed, the program creates multiple files by the script included in the batch file. The file created as a result of running the batch file is shown in Figure 5.38.

```

java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 1 D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\contentKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 2 D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\contentKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 3 D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\contentKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 4 D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\contentKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 5 D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\contentKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\contentKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\contentKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\contentKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\contentKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\contentKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureDescriptorKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureDescriptorKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureDescriptorKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureDescriptorKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureDescriptorKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry.txt

```

Figure 5.37: Sample scriptKatyPerry.bat

contentKatyPerry.1n	11/8/2021 1:01 AM	1N File	73 KB
contentKatyPerry.2n	11/8/2021 1:01 AM	2N File	109 KB
contentKatyPerry.3n	11/8/2021 1:01 AM	3N File	143 KB
contentKatyPerry.4n	11/8/2021 1:01 AM	4N File	177 KB
contentKatyPerry.5n	11/8/2021 1:01 AM	5N File	210 KB
contentKatyPerry	10/8/2021 3:10 PM	Text Document	37 KB
featureDescriptorKatyPerry.1n	11/8/2021 1:01 AM	1N File	1 KB
featureDescriptorKatyPerry.2n	11/8/2021 1:01 AM	2N File	6 KB
featureDescriptorKatyPerry.3n	11/8/2021 1:01 AM	3N File	28 KB
featureDescriptorKatyPerry.4n	11/8/2021 1:01 AM	4N File	66 KB
featureDescriptorKatyPerry.5n	11/8/2021 1:01 AM	5N File	111 KB
featureVectorKatyPerry.1n	11/8/2021 10:38 AM	1N File	45 KB
featureVectorKatyPerry.2n	11/8/2021 10:40 AM	2N File	1,198 KB
featureVectorKatyPerry.3n	11/8/2021 10:41 AM	3N File	4,903 KB
featureVectorKatyPerry.4n	11/8/2021 10:41 AM	4N File	9,285 KB
featureVectorKatyPerry.5n	11/8/2021 1:02 AM	5N File	12,958 KB

Figure 5.38: Sample File Created for Katy Perry

5.3.3 Feature Extraction

N-gram will be used to create features in the feature extraction method. An n-gram is a sequence of n words extracted from such a single text stream. Also,

$n=1,2,3,4,5$ will be utilized as the n-gram size. These methods produce a feature descriptor by each dataset, and the delimiter is a pipe symbol '|'.

5.3.3.1 Eminem Dataset Feature Extraction

This part will fully describe the outcome of feature extraction from the dataset Eminem and the parameters used to run the scripts in Eclipse. For the Eminem dataset, there are two kinds of experiments: single classifier and multiple classifiers. Each experiment has five different n-gram sizes. So, first, utilize the Java program and export it into a runnable jar file. The batch file used to create N-grams ($n=1,2,3,4,5$) for both experiment 1 and experiment 2 is shown in figures 5.39.

```
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 1 D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 2 D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 3 D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 4 D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 5 D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.txt
```

Figure 5.39: Generate Ngram Eminem

Following the execution, the script produces various files depending on the batch file's arguments. The file created as a result of executing the batch file for both experiments is shown in Table 5.1.

Table 5.A: Generate Ngram Eminem

Ngram	Generate Ngram
1	h e y g u y s i m a 1 7 y r o l d + 4 4 7 9 3 5 4 5 4 1 5 0 l o v e l y g i m y s i s t e r j u s t r e c e i v e d
2	h e y y g g u y y s s i i m m a a 1 1 7 7 7 +4 44 47 79 93 35 54 45 54 41 15 50 0 1 1 o o c m y y s s i i s s t t e e r r j j u u s s t t r r
3	h e y y y g g g u g u y y s y s s i i i m i m m a a a +44 447 479 793 935 354 545 454 541 415 150 50 m y y s s i s s i s s t t e e r r r j j u u s s u s t

4	hey ey g y gu guy guys uys ys i s im im im +447 4479 4793 7935 9354 3545 5454 4541 5415 4154 my s y si sis sist iste ster ter er j r ju jv
5	hey g ey gu y guy guys guys uys i ys im s i +4479 44793 47935 79354 93545 35454 54541 454 my si y sis sist siste ister ster ter j er

Once finishing both experiments, create the Ngram feature descriptor. Figure 5.40 shows the batch file for generating Ngram feature descriptors (1,2,3,4,5) for both experiments.

```
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.1n D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureDescriptorEminem.1n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.2n D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureDescriptorEminem.2n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.3n D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureDescriptorEminem.3n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.4n D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureDescriptorEminem.4n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.5n D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureDescriptorEminem.5n
```

Figure 5.40: Batch File Eminem

Table 5.B: Feature Descriptor Eminem

Ngram	Generate Ngram
1	! " # \$ % & () * + , - . / 0 1 2 3 4 5 6 7 8
2	! " # \$ % & (, - . / 1 2 3 4 5 6 7 8
3	& - . / 2 : < a b c d e f
4	- . b c d g i l m p r s
5	- c m p r -- -n . be ch cr

5.3.3.2 Psy Dataset Feature Extraction

Here we'll go through the feature extraction results and how to execute the scripts in Eclipse. There are two types of experiments for the Psy dataset: single and multiple classifiers. The n-gram sizes vary across experiments. Use the Java application and export it as a runnable Jar. Figure 5.41 shows the batch file used to generate N-grams (n=1,2,3,4,5).

```
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgram 1 D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgram 2 D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgram 3 D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgram 4 D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgram 5 D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.txt
```

Figure 5.41: Generate Ngram Psy

Following the execution, the script generates various files based on the parameters specified in the batch file. Table 5.3 illustrates the file produced as a consequence of running the batch file for such tests.

Table 5.C: Parameter Details of Generate Ngram Psy

Ngram	Generate Ngram
1	h u h , a n y w a y c h e c k o u t t h i s y h e y g u y s c h e c k o u t m y n e w c h j u s t f o r t e s t i h a v e t o s a y
2	h u h h, , a a n n y w w a a y y c c h h e e c c k k o o u h e e y y g g u u y y s s c c h h e e c c k k o o u u t t m j u u s s t t f f o o r r t t e e s s t t i i h h a a v v e
3	huh uh, h, , a a n any nyw ywa way ay y c c h che hec hey ey y g g u gu uys ys s c c h che hec eck ck k c jus ust st t f f o for or r t t e tes est st t i i
4	huh, uh, h, a , a n any anyw nywa yway way lay c y c h che hey ey g y g u g u y g u y s u y s c s c h che chec heck eck just ust st f t fo for for or t r te tes test est st i
5	huh, uh, a h, a n , a n any anyw anywa nyway yway way c ay c h y ch hey g ey g u y g u y g u y s g u y s u y s c y s c h s che chec check heck just ust fst fo t for for for t or te r tes test test est

Once finishing both experiments, create the Ngram feature descriptor. Figure 5.42 shows the batch file for generating Ngram feature descriptors (1,2,3,4,5) for both experiments.

```
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.1n D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureDescriptorPsy.1n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.2n D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureDescriptorPsy.2n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.3n D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureDescriptorPsy.3n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.4n D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureDescriptorPsy.4n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.5n D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureDescriptorPsy.5n
```

Figure 5.42: Batch File Psy

Table 5.D: Feature Descriptor Psy

Ngram	Generate Ngram
1	! " # \$ % & ' () * + , - . / 0 1 2 3 4 5 6 7
2	! " # \$ % & ' ()) * , - . / 0 1 2 3 4 5 6 7
3	! " # & (- . / 2 3 4 5 7 @ a
4	! " # 3 5 @ c h p t y ! 5 # s
5	_e h p t y _3 _5m _e _ch _co

5.3.3.3 Shakira Dataset Feature Extraction

This part will fully describe the outcome of feature extraction from the dataset Shakira and the parameters used to run the scripts in Eclipse. For the Shakira dataset, there are two kinds of experiments: single classifier and multiple classifiers. Each experiment has five different n-gram sizes. So, first, utilize the Java program and export it into a runnable jar file. The batch file used to create N-grams (n=1,2,3,4,5) for both experiment 1 and experiment 2 is shown in figures 5.43.

```
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 1 D:\PSM2\YouTube-Spam-Collection-v1\Shakira\contentShakira.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 2 D:\PSM2\YouTube-Spam-Collection-v1\Shakira\contentShakira.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 3 D:\PSM2\YouTube-Spam-Collection-v1\Shakira\contentShakira.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 4 D:\PSM2\YouTube-Spam-Collection-v1\Shakira\contentShakira.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 5 D:\PSM2\YouTube-Spam-Collection-v1\Shakira\contentShakira.txt
```

Figure 5.43: Generate Ngram Shakira

Following the execution, the script produces various files depending on the batch file's arguments. The file created as a result of executing the batch file for such experiments is shown in Table 5.5.

Table 5.E: Generate Ngram Shakira

Ngram	Generate Ngram
1	s e e e s o m e m o r e s o n g o p e n g o o g l e a n d c h e c k o u t t h i s p l a y l i s t o n y o u t u b e s u p p o r t t h e f i g h t f o r y o u r 4 t h a m e n
2	s e e e e s s o m e e m m o r e r e s s o n ng o o p e n r c h e c c k k o o u t t t h h i i s s p p l a a y y l i i s s t t s u u p p p o r r t t t t h h e e f f i i g g h h t t f f o o r r y y
3	s e e e e e s s s o s o m o m e e e m m m o r o r e r e e s s s o s o n o ng ng c h e h e c c k c k k o o u o u t t t t t h t h i h i s s s p p l p l a l a s u u p p p p o p o r o r t t t t t h t h e h e e f f i f i g g h h t t
4	s e e e e s e s o s o m s o m e e e m e m m o r o r e r e r e s e s o s o n s c h e c h e c h e c c c k o k o u o u t t t t t h t h i h i s s i s p s p l s u u p p p o p p o r o r t t t t t h t h e h e e f e f i f i g g h h t t
5	s e e s e e s o e s o m s o m e o m e m m e m m e m m o r m m o r e r e o r e s r e s o c h e c h e c h e c c c k o c k o u o u t t t t t h t h i h i s s i h i s p s u u p p p o r p p o r o r t t t t t h t h e h e t h e t h e f h e f i e f i g f i h

Once finishing both experiments, create the Ngram feature descriptor. Figure 5.44 shows the batch file for generating Ngram feature descriptors (1,2,3,4,5) for both experiments.

```
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\contentShakira.1n D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureDescriptorShakira.1n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\contentShakira.2n D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureDescriptorShakira.2n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\contentShakira.3n D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureDescriptorShakira.3n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\contentShakira.4n D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureDescriptorShakira.4n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\contentShakira.5n D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureDescriptorShakira.5n
```

Figure 5.44: Batch File Shakira

Table 5.F: Feature Descriptor Shakira

Ngram	Generate Ngram
1	! " # \$ % & () * + , - . / 0 1 2 3 4 5 6 7 8
2	! \$ & () * + , - . / 0 1 2 3 4 5 6
3	\$ & + - . : ; < a b c e f
4	\$. < b c g i l m p s t u
5	b p \$2 . <b bi ch go gt

5.3.3.4 LMFAO Dataset Feature Extraction

Here we'll go through the feature extraction results and how to execute the scripts in Eclipse. There are two types of experiments for the LMFAO dataset: single and multiple classifiers. The n-gram sizes vary across experiments. Use the Java application and export it as a runnable Jar. Figure 5.45 shows the batch file used to generate N-grams (n=1,2,3,4,5).

```
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgram 1 D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\contentLmfao.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgram 2 D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\contentLmfao.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgram 3 D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\contentLmfao.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgram 4 D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\contentLmfao.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgram 5 D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\contentLmfao.txt
```

Figure 5.45: Generate Ngram LMFAO

Following the execution, the script generates various files based on the parameters specified in the batch file. Table 5.7 illustrates the file produced as a consequence of running the batch file for such tests.

Table 5.G: Parameter Details of Generate Ngram LMFAO

Ngram	Generate Ngram
1	h e y g u y s , 1 & # 3 9 ; m a h u m a n . < b r / c h e c k o u t t h i s v i d e o o n y o u t u b l e c h e c k o u t t h i s v i d e o o n y o u t u b l e

2	he ey y g gu uy ys s, , i i &# 39 9; ;m m a a h h u u m ma ch he ec ck k o ou ut t t th hi is s v vi id de eo o o on n ch he ec ck k o ou ut t t th hi is s v vi id de eo o o on n
3	hey ey y g gu guy uys ys, s, , i i i &# ' 39; 9;m m che hec eck ck k o ou out ut t t th thi his is s v vi vid che hec eck ck k o ou out ut t t th thi his is s v vi vid
4	hey ey g y gu gu guys uys, ys, s, i , i i i &# i i ' i ' 3 chec heck eck ck o k ou out out ut t t th thi this his is chec heck eck ck o k ou out out ut t t th thi this his is
5	hey g ey gu y gu guys guys, uys, ys, s, i s, i i , i &# i i ' 3 check heck eck o ck ou k out out out t ut th t thi this this check heck eck o ck ou k out out out t ut th t thi this this

```
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\contentLmfao.1n D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureDescriptorLmfao.1n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\contentLmfao.2n D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureDescriptorLmfao.2n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\contentLmfao.3n D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureDescriptorLmfao.3n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\contentLmfao.4n D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureDescriptorLmfao.4n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\contentLmfao.5n D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureDescriptorLmfao.5n
```

Figure 5.46: Batch File LMFAO

Table 5.H: Feature Descriptor LMFAO

Ngram	Generate Ngram
1	! " # \$ % & () + , - . / 0 1 2 3 4 5 6 7 8
2	! \$ & () , - / 1 2 3 4 5 6
3	< \ g h m p s t y ! ! ! \$ &
4	< \ y <a <b \ \ < go he ma pe
5	< \ y <b \ \ yo <a <br

5.3.3.5 Katy Perry Dataset Feature Extraction

This part will fully describe the outcome of feature extraction from the dataset Katy Perry and the parameters used to run the scripts in Eclipse. For the Katy Perry dataset, there are two kinds of experiments: single classifier and multiple classifiers. Each experiment has five different n-gram sizes. So, first, utilize the Java program and export it into a runnable jar file. The batch file used to create N-grams (n=1,2,3,4,5) for experiments 1 and 2 is shown in figures 5.47.

Figure 5.47: Generate Ngram Katy Perry

Following the execution, the script produces various files depending on the batch file's arguments. The file created as a result of executing the batch file for such experiments is shown in Table 5.9.

```
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 1 D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\contentKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 2 D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\contentKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 3 D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\contentKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 4 D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\contentKatyPerry.txt
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGram 5 D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\contentKatyPerry.txt
```

Table 5.I: Generate Ngram Katy Perry

Ngram	Generate Ngram
1	i l lo ov e t h h i s s o m u c h . a n d a l s o i g e n h t t p : / / w w w . b i i l l b o a r d . c o m / a r t i c l e s / c h e y g u y s ! p l e a s e j o i n m e i n m y f i g h
2	i l lo ov ve e t th h i s s s o o m m u c h h . a a n n d d h t t t p : : / / / w w w w . b b i i l l l b b o o a r r d . c c o o m m/ h e y y g g u u y y s s! ! p p l l e e a a s s e e j j o o i i n n m m e e
3	i l lo lov ove ve e t th thi his is s s so so o m mu muc uch h t t t p t p : p : / : / / w w w w w . w .b .b i bil ill llb lbo boa oar h e y e y y g g u guy uys ys !s ! ! p p l ple lea leas ase se e j j o
4	i lo lov love ove ve t e t h thi this his is s s so so so m o m h t t t p : t p : / p : / / w //w //w //w w w . w .b w .b i .bil bill illb llb h e y e y g y g u g u y guys uys !ys ! s ! p ! p l ple plea leas ease ase
5	i lov love love ove t ve t h e thi this this his s is s s so so m s h t t t p : t t p : / t p : / / p ://w ://w //w //w w w . w .w .b w .b i .bil bill illb llb h e y g e y g u y g u y guys guys !uys ! ys ! p s ! p l ! ple plea pleas lease e e

Once finishing both experiments, create the Ngram feature descriptor. Figure 5.48 shows the batch file for generating Ngram feature descriptors (1,2,3,4,5) for both experiments.

```
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\content\KatyPerry.1n D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureDescriptor\KatyPerry.1n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\content\KatyPerry.2n D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureDescriptor\KatyPerry.2n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\content\KatyPerry.3n D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureDescriptor\KatyPerry.3n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\content\KatyPerry.4n D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureDescriptor\KatyPerry.4n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureDescriptor
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\content\KatyPerry.5n D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureDescriptor\KatyPerry.5n
```

Figure 5.48: Batch File Katy Perry

Table 5.J: Feature Descriptor Katy Perry

Ngram	Generate Ngram
1	! " # \$ % & ' () * + , - . / 0 1 2 3 4 5 6 7 8
2	! " # \$ % & ' () * + , - . / 0 1
3	! " # & ' () . / 1 6 9 ^
4	! " # / 6 b d f g h i l n
5	! b d f g h i l

5.3.4 Feature Vector

As previously discussed in the previous chapter, there seem to be four steps in executing the feature vector module. Figure 5.49 illustrates the actions to be taken.



Figure 5.49: Feature Vector Module

5.3.4.1 Dataset Eminem Feature Vector

Following the methods outlined in the feature vector module, this chapter will cover how to construct Ngram feature vectors for the Eminem dataset. When carrying out this procedure, will use the Java code exported into a runnable jar file that it may use for batch execution of a program. Figures 5.50 resembles the batch file used to create the feature vectors for the experiments, respectively.

```
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureDescriptorEminem.1n D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVectorEminem.1n
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.1n 245 D:\PSM2\YouTube-Spam-Collection-v1\Eminem\renameFeatureVectorEminem.1n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureDescriptorEminem.2n D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVectorEminem.2n
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.2n 245 D:\PSM2\YouTube-Spam-Collection-v1\Eminem\renameFeatureVectorEminem.2n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureDescriptorEminem.3n D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVectorEminem.3n
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.3n 245 D:\PSM2\YouTube-Spam-Collection-v1\Eminem\renameFeatureVectorEminem.3n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureDescriptorEminem.4n D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVectorEminem.4n
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.4n 245 D:\PSM2\YouTube-Spam-Collection-v1\Eminem\renameFeatureVectorEminem.4n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureDescriptorEminem.5n D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVectorEminem.5n
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.5n 245 D:\PSM2\YouTube-Spam-Collection-v1\Eminem\renameFeatureVectorEminem.5n
```

Figure 5.50: Batch File for Ngram Feature Vector Eminem

Table 5.11 provides a detailed explanation of each command that was used to create the Ngram feature vector. Next, run the batch file in command prompt to see whether it works properly.

Table 5.K: Details Parameter of Feature Vector Eminem

Command	Description
-Xmx2G	RAM size
-cp	Execute
Classify.jar	Executable .jar file
psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeature Vector	Package and file name
D:\PSM2\YouTube-Spam-Collection v1\Eminem\featureDescriptorEminem.1n	Path for file descriptor, Ngram (n=1,2,3,4,5)

D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVectorEminem.1n	Path for file feature vector, Ngram (n=1,2,3,4,5)
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\contentEminem.1n	Path for file content, Ngram (n=1,2,3,4,5)
245	The last spam, boundary between spam and legitimate
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\renameFeatureVectorEminem.1n	The path for rename feature descriptor, Ngram (n=1,2,3,4,5)

In Table 5.12, the experiments' batch files are shown.

Table 5.L: Generate Feature Vector Eminem for Both Experiments

Experiment	Gram	Feature Vector	Rename Feature vector																																																															
Single classifier and Multiple Classifier	1	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>AA</th> <th>AB</th> <th>AC</th> <th>BO</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f27</td> <td>f28</td> <td>f29</td> <td>class</td> </tr> <tr> <td>2</td> <td>103</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>3</td> <td>6</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>247</td> <td>9</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>248</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		A	B	C	AA	AB	AC	BO	1	f1	f2	f3	f27	f28	f29	class	2	103	0	1	0	0	0	1	3	6	0	0	0	0	0	1	247	9	0	0	0	1	0	0	248	0	0	0	0	0	0	0	<table border="1"> <tbody> <tr> <td>1</td> <td>f1</td> <td></td> </tr> <tr> <td>2</td> <td>f2</td> <td>!</td> </tr> <tr> <td>65</td> <td>f65</td> <td>}</td> </tr> <tr> <td>66</td> <td>f66</td> <td>~</td> </tr> </tbody> </table>	1	f1		2	f2	!	65	f65	}	66	f66	~			
		A	B	C	AA	AB	AC	BO																																																										
1	f1	f2	f3	f27	f28	f29	class																																																											
2	103	0	1	0	0	0	1																																																											
3	6	0	0	0	0	0	1																																																											
247	9	0	0	0	1	0	0																																																											
248	0	0	0	0	0	0	0																																																											
1	f1																																																																	
2	f2	!																																																																
65	f65	}																																																																
66	f66	~																																																																
	2	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>AA</th> <th>AB</th> <th>AC</th> <th>AOK</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f27</td> <td>f28</td> <td>f29</td> <td>class</td> </tr> <tr> <td>2</td> <td>6</td> <td>0</td> <td>1</td> <td>0</td> <td>9</td> <td>5</td> <td>1</td> </tr> <tr> <td>3</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>453</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>454</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		A	B	C	AA	AB	AC	AOK	1	f1	f2	f3	f27	f28	f29	class	2	6	0	1	0	9	5	1	3	0	0	0	0	0	0	1	453	0	1	0	0	1	0	0	454	0	0	0	0	0	0	0	<table border="1"> <tbody> <tr> <td>1</td> <td>f1</td> <td></td> </tr> <tr> <td>2</td> <td>f2</td> <td>!</td> </tr> <tr> <td>3</td> <td>f3</td> <td></td> </tr> <tr> <td>1075</td> <td>f1075</td> <td>~!</td> </tr> <tr> <td>1076</td> <td>f1076</td> <td>~t</td> </tr> </tbody> </table>	1	f1		2	f2	!	3	f3		1075	f1075	~!	1076	f1076	~t
	A	B	C	AA	AB	AC	AOK																																																											
1	f1	f2	f3	f27	f28	f29	class																																																											
2	6	0	1	0	9	5	1																																																											
3	0	0	0	0	0	0	1																																																											
453	0	1	0	0	1	0	0																																																											
454	0	0	0	0	0	0	0																																																											
1	f1																																																																	
2	f2	!																																																																
3	f3																																																																	
1075	f1075	~!																																																																
1076	f1076	~t																																																																

3			A	B	C	D	FWI	FWJ		
	1	f1	f2	f3	f4	f4663	class		1	f1
	2	1	0	1	0	0	0	1	2	f2
	3	0	0	0	0	0	0	1	4662	f4662
	453	0	0	0	0	0	0	0	4663	f4663
454	0	0	0	0	0	0	0			
4			A	B	C	JBL	OTS			
	1	f1	f2	f3	f6824	class			1	f1
	2	0	1	0	0	1			2	f2
	453	0	0	0	0	0			10677	f10677
	454	0	0	0	0	0			10678	f10678
5			A	B	C	CT	RL			
	1	f1	f2	f3	f98	class			1	f1
	2	0	0	0	0	1			2	f2
	453	0	0	0	0	0			16474	f16474
	454	0	0	0	0	0			16475	f16475

To execute a batch script, transform the Ngram feature vector obtained from a CSV file to an ARFF file via Java code. Then run the batch file from the command prompt. The batch file for the experiments is given in figure 5.51. Table 5.13 lists the commands being used to convert CSV to ARFF.

```

java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVector\Eminem.1n D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVector\Eminem1n.arff
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVector\Eminem.2n D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVector\Eminem2n.arff
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVector\Eminem.3n D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVector\Eminem3n.arff
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVector\Eminem.4n D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVector\Eminem4n.arff
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVector\Eminem.5n D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVector\Eminem5n.arff

```

Figure 5.51: Batch File Convert CSV to ARFF Eminem

Table 5.M: Details Parameter of Convert CSV to ARFF Eminem

Command	Description
-Xmx2G	RAM size

-cp	Execute
Classify.jar	Executable .jar file
psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF	Package and file name
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVectorEminem.1n	Path for file vector, Ngram (n=1,2,3,4,5)
D:\PSM2\YouTube-Spam-Collection-v1\Eminem\featureVectorEminem1n.arff	Path for file feature vector ARFF, Ngram (n=1,2,3,4,5)

5.3.4.2 Dataset Psy Feature Vector

Following the methods outlined in the feature vector module, this chapter will cover how to construct Ngram feature vectors for the Psy dataset. When carrying out this procedure, will use the Java code exported into a runnable jar file that it may use for batch execution of a program. Figures 5.52 resembles the batch file used to create the feature vectors for the experiments, respectively.

```

java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureDescriptorPsy.1n D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy.1n
D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.1n 175 D:\PSM2\YouTube-Spam-Collection-v1\Psy\renameFeatureVectorPsy.1n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureDescriptorPsy.2n D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy.2n
D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.2n 175 D:\PSM2\YouTube-Spam-Collection-v1\Psy\renameFeatureVectorPsy.2n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureDescriptorPsy.3n D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy.3n
D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.3n 175 D:\PSM2\YouTube-Spam-Collection-v1\Psy\renameFeatureVectorPsy.3n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureDescriptorPsy.4n D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy.4n
D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.4n 175 D:\PSM2\YouTube-Spam-Collection-v1\Psy\renameFeatureVectorPsy.4n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureDescriptorPsy.5n D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy.5n
D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.5n 175 D:\PSM2\YouTube-Spam-Collection-v1\Psy\renameFeatureVectorPsy.5n

```

Figure 5.52: Batch File for Ngram Feature Vector Psy

Table 5.14 provides a detailed explanation of each command that was used to create the Ngram feature vector. Next, run the batch file in command prompt to see whether it works properly.

Table 5.N: Details Parameter of Feature Vector Psy

Command	Description
-Xmx2G	RAM size
-cp	Execute
Classify.jar	Executable .jar file
psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector	Package and file name
D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureDescriptorPsy.1n	Path for file descriptor, Ngram (n=1,2,3,4,5)
D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy.1n	Path for file feature vector, Ngram (n=1,2,3,4,5)
D:\PSM2\YouTube-Spam-Collection-v1\Psy\contentPsy.1n	Path for file content, Ngram (n=1,2,3,4,5)
175	The last spam, boundary between spam and legitimate
D:\PSM2\YouTube-Spam-Collection-v1\Psy\renameFeatureVectorPsy.1n	The path for rename feature descriptor, Ngram (n=1,2,3,4,5)

In Table 5.15, the experiments' batch files are shown.

Table 5.O: Generate Feature Vector Psy for Both Experiments

Experiment	Gram	Feature Vector	Rename Feature vector																																																														
Single classifier and Multiple Classifiers	1	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>BN</th> <th>BO</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f66</td> <td>class</td> </tr> <tr> <td>2</td> <td>103</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>453</td> <td>3</td> <td>42</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>454</td> <td>6</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		A	B	C	BN	BO	1	f1	f2	f3	f66	class	2	103	0	1	0	1	453	3	42	0	0	0	454	6	0	0	0	0	<table border="1"> <tbody> <tr> <td>2</td> <td>f2</td> <td>!</td> </tr> <tr> <td>3</td> <td>f3</td> <td></td> </tr> <tr> <td>65</td> <td>f65</td> <td>}</td> </tr> <tr> <td>66</td> <td>f66</td> <td>~</td> </tr> </tbody> </table>	2	f2	!	3	f3		65	f65	}	66	f66	~																				
		A	B	C	BN	BO																																																											
	1	f1	f2	f3	f66	class																																																											
	2	103	0	1	0	1																																																											
	453	3	42	0	0	0																																																											
454	6	0	0	0	0																																																												
2	f2	!																																																															
3	f3																																																																
65	f65	}																																																															
66	f66	~																																																															
2	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>BG</th> <th>BH</th> <th>AOK</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f59</td> <td>f60</td> <td>class</td> </tr> <tr> <td>2</td> <td>6</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>3</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>453</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>454</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		A	B	C	BG	BH	AOK	1	f1	f2	f3	f59	f60	class	2	6	0	1	0	0	1	3	0	0	0	0	0	1	453	0	1	0	0	0	0	454	0	0	0	0	0	0	<table border="1"> <tbody> <tr> <td>1</td> <td>f1</td> <td></td> </tr> <tr> <td>2</td> <td>f2</td> <td>!</td> </tr> <tr> <td>3</td> <td>f3</td> <td></td> </tr> <tr> <td>1075</td> <td>f1075</td> <td>~!</td> </tr> <tr> <td>1076</td> <td>f1076</td> <td>~t</td> </tr> </tbody> </table>	1	f1		2	f2	!	3	f3		1075	f1075	~!	1076	f1076	~t						
	A	B	C	BG	BH	AOK																																																											
1	f1	f2	f3	f59	f60	class																																																											
2	6	0	1	0	0	1																																																											
3	0	0	0	0	0	1																																																											
453	0	1	0	0	0	0																																																											
454	0	0	0	0	0	0																																																											
1	f1																																																																
2	f2	!																																																															
3	f3																																																																
1075	f1075	~!																																																															
1076	f1076	~t																																																															
3	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>FH</th> <th>HCZ</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f164</td> <td>class</td> </tr> <tr> <td>2</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>3</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>350</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>351</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		A	B	C	FH	HCZ	1	f1	f2	f3	f164	class	2	0	0	0	0	1	3	0	0	0	0	1	350	0	0	0	0	0	351	0	0	0	0	0	<table border="1"> <tbody> <tr> <td>2</td> <td>f2</td> <td>&</td> </tr> <tr> <td>3</td> <td>f3</td> <td>-</td> </tr> <tr> <td>4662</td> <td>f4662</td> <td>~!~</td> </tr> <tr> <td>4663</td> <td>f4663</td> <td>~th</td> </tr> </tbody> </table>	2	f2	&	3	f3	-	4662	f4662	~!~	4663	f4663	~th															
	A	B	C	FH	HCZ																																																												
1	f1	f2	f3	f164	class																																																												
2	0	0	0	0	1																																																												
3	0	0	0	0	1																																																												
350	0	0	0	0	0																																																												
351	0	0	0	0	0																																																												
2	f2	&																																																															
3	f3	-																																																															
4662	f4662	~!~																																																															
4663	f4663	~th																																																															
4	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>ABE</th> <th>HBB</th> <th>OQN</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f733</td> <td>f5462</td> <td>class'</td> </tr> <tr> <td>2</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>3</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>350</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>351</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		A	B	C	ABE	HBB	OQN	1	f1	f2	f3	f733	f5462	class'	2	0	0	0	0	0	1	3	0	0	0	0	0	1	350	0	0	0	0	0	0	351	0	0	0	0	0	0	<table border="1"> <tbody> <tr> <td>1</td> <td>f1</td> <td></td> </tr> <tr> <td>2</td> <td>f2</td> <td>-</td> </tr> <tr> <td>3</td> <td>f3</td> <td>.</td> </tr> <tr> <td>4</td> <td>f4</td> <td>b</td> </tr> <tr> <td>5</td> <td>f5</td> <td>c</td> </tr> <tr> <td>10677</td> <td>f10677</td> <td>}tha</td> </tr> <tr> <td>10678</td> <td>f10678</td> <td>~tha</td> </tr> </tbody> </table>	1	f1		2	f2	-	3	f3	.	4	f4	b	5	f5	c	10677	f10677	}tha	10678	f10678	~tha
	A	B	C	ABE	HBB	OQN																																																											
1	f1	f2	f3	f733	f5462	class'																																																											
2	0	0	0	0	0	1																																																											
3	0	0	0	0	0	1																																																											
350	0	0	0	0	0	0																																																											
351	0	0	0	0	0	0																																																											
1	f1																																																																
2	f2	-																																																															
3	f3	.																																																															
4	f4	b																																																															
5	f5	c																																																															
10677	f10677	}tha																																																															
10678	f10678	~tha																																																															
5	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>VN</th> <th>FQB</th> <th>OQN</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f586</td> <td>f4500</td> <td>class</td> </tr> <tr> <td>2</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>3</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>350</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>351</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		A	B	C	VN	FQB	OQN	1	f1	f2	f3	f586	f4500	class	2	0	0	0	0	0	1	3	0	0	0	1	0	1	350	0	0	0	0	0	0	351	0	0	0	0	0	0	<table border="1"> <tbody> <tr> <td>1</td> <td>f1</td> <td></td> </tr> <tr> <td>2</td> <td>f2</td> <td>-</td> </tr> <tr> <td>3</td> <td>f3</td> <td>c</td> </tr> <tr> <td>16474</td> <td>f16474</td> <td>}than</td> </tr> <tr> <td>16475</td> <td>f16475</td> <td>~than</td> </tr> </tbody> </table>	1	f1		2	f2	-	3	f3	c	16474	f16474	}than	16475	f16475	~than						
	A	B	C	VN	FQB	OQN																																																											
1	f1	f2	f3	f586	f4500	class																																																											
2	0	0	0	0	0	1																																																											
3	0	0	0	1	0	1																																																											
350	0	0	0	0	0	0																																																											
351	0	0	0	0	0	0																																																											
1	f1																																																																
2	f2	-																																																															
3	f3	c																																																															
16474	f16474	}than																																																															
16475	f16475	~than																																																															

To execute a batch script, transform the Ngram feature vector obtained from a CSV file to an ARFF file via Java code. Then run the batch file in from the command prompt. The batch file for the experiments is given in figure 5.53. Table 5.16 lists the commands being used to convert CSV to ARFF.

```
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy.1n D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy1n.arff
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy.2n D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy2n.arff
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy.3n D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy3n.arff
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy.4n D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy4n.arff
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy.5n D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy5n.arff
```

Figure 5.53: Batch File Convert CSV to ARFF Psy

Table 5.P: Details Parameter of Convert CSV to ARFF Psy

Command	Description
-Xmx2G	RAM size
-cp	Execute
Classify.jar	Executable .jar file
psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF	Package and file name
D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy.1n	Path for file vector, Ngram (n=1,2,3,4,5)
D:\PSM2\YouTube-Spam-Collection-v1\Psy\featureVectorPsy1n.arff	Path for file feature vector ARFF, Ngram (n=1,2,3,4,5)

5.3.4.3 Dataset Shakira Feature Vector

Following the methods outlined in the feature vector module, this chapter will cover how to construct Ngram feature vectors for the Shakira dataset. When carrying out this procedure, will use the Java code exported into a runnable jar file that it may

use for batch execution of a program. Figures 5.54 resembles the batch file used to create the feature vectors for the experiments, respectively.

```
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureDescriptorShakira.1n D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShakira.1n
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\contentShakira.1n 174 D:\PSM2\YouTube-Spam-Collection-v1\Shakira\renameFeatureVectorShakira.1n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureDescriptorShakira.2n D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShakira.2n
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\contentShakira.2n 174 D:\PSM2\YouTube-Spam-Collection-v1\Shakira\renameFeatureVectorShakira.2n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureDescriptorShakira.3n D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShakira.3n
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\contentShakira.3n 174 D:\PSM2\YouTube-Spam-Collection-v1\Shakira\renameFeatureVectorShakira.3n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureDescriptorShakira.4n D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShakira.4n
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\contentShakira.4n 174 D:\PSM2\YouTube-Spam-Collection-v1\Shakira\renameFeatureVectorShakira.4n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureDescriptorShakira.5n D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShakira.5n
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\contentShakira.5n 174 D:\PSM2\YouTube-Spam-Collection-v1\Shakira\renameFeatureVectorShakira.5n
```

Figure 5.54: Batch File for Ngram Feature Vector Shakira

Table 5.17 provides a detailed explanation of each command that was used to create the Ngram feature vector. Next, run the batch file in command prompt to see whether it works properly.

Table 5.Q: Details Parameter of Feature Vector Shakira

Command	Description
-Xmx2G	RAM size
-cp	Execute
Classify.jar	Executable .jar file
psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector	Package and file name
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureDescriptorShakira.1n	Path for file descriptor, Ngram (n=1,2,3,4,5)
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShakira.1n	Path for file feature vector, Ngram (n=1,2,3,4,5)

D:\PSM2\YouTube-Spam-Collection-v1\Shakira\contentShakira.1n	Path for file content, Ngram (n=1,2,3,4,5)
175	The last spam, boundary between spam and legitimate
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\renameFeatureVector Shakira.1n	The path for rename feature descriptor, Ngram (n=1,2,3,4,5)

In Table 5.18, the experiments' batch files are shown.

Table 5.R: Generate Feature Vector Shakira for Both Experiments

Experiment	Gram	Feature Vector	Rename Feature vector																																																								
Single classifier and Multiple Classifiers	1	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>AC</th> <th>BL</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f29</td> <td>class</td> </tr> <tr> <td>2</td> <td>9</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>3</td> <td>5</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>370</td> <td>4</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>371</td> <td>4</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		A	B	C	AC	BL	1	f1	f2	f3	f29	class	2	9	0	0	0	1	3	5	0	0	0	1	370	4	0	0	0	0	371	4	0	0	0	0	<table border="1"> <tbody> <tr> <td>1</td> <td>f1</td> <td></td> </tr> <tr> <td>2</td> <td>f2</td> <td>!</td> </tr> <tr> <td>3</td> <td>f3</td> <td></td> </tr> <tr> <td>62</td> <td>f62</td> <td>z</td> </tr> <tr> <td>63</td> <td>f63</td> <td>~</td> </tr> </tbody> </table>	1	f1		2	f2	!	3	f3		62	f62	z	63	f63	~					
		A	B	C	AC	BL																																																					
	1	f1	f2	f3	f29	class																																																					
2	9	0	0	0	1																																																						
3	5	0	0	0	1																																																						
370	4	0	0	0	0																																																						
371	4	0	0	0	0																																																						
1	f1																																																										
2	f2	!																																																									
3	f3																																																										
62	f62	z																																																									
63	f63	~																																																									
2	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>BB</th> <th>AHA</th> <th>AMH</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f54</td> <td>f85</td> <td>class</td> </tr> <tr> <td>2</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>3</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>370</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>371</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		A	B	C	BB	AHA	AMH	1	f1	f2	f3	f54	f85	class	2	0	0	0	0	0	1	3	0	0	0	0	0	1	370	0	0	0	0	0	0	371	0	0	0	0	0	0	<table border="1"> <tbody> <tr> <td>1</td> <td>f1</td> <td></td> </tr> <tr> <td>2</td> <td>f2</td> <td>!</td> </tr> <tr> <td>3</td> <td>f3</td> <td>\$</td> </tr> <tr> <td>1020</td> <td>f1020</td> <td>zz</td> </tr> <tr> <td>1021</td> <td>f1021</td> <td>~a</td> </tr> </tbody> </table>	1	f1		2	f2	!	3	f3	\$	1020	f1020	zz	1021	f1021	~a
	A	B	C	BB	AHA	AMH																																																					
1	f1	f2	f3	f54	f85	class																																																					
2	0	0	0	0	0	1																																																					
3	0	0	0	0	0	1																																																					
370	0	0	0	0	0	0																																																					
371	0	0	0	0	0	0																																																					
1	f1																																																										
2	f2	!																																																									
3	f3	\$																																																									
1020	f1020	zz																																																									
1021	f1021	~a																																																									
3	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>NO</th> <th>EXP</th> <th>FJF</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f379</td> <td>f4020</td> <td>class</td> </tr> <tr> <td>2</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>3</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>370</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>371</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		A	B	C	NO	EXP	FJF	1	f1	f2	f3	f379	f4020	class	2	0	0	0	0	0	1	3	0	0	0	0	0	1	370	0	0	0	0	0	0	371	0	0	0	0	0	0	<table border="1"> <tbody> <tr> <td>1</td> <td>f1</td> <td></td> </tr> <tr> <td>2</td> <td>f2</td> <td>\$</td> </tr> <tr> <td>3</td> <td>f3</td> <td>&</td> </tr> <tr> <td>4320</td> <td>f4320</td> <td>zzm</td> </tr> <tr> <td>4321</td> <td>f4321</td> <td>~ax</td> </tr> </tbody> </table>	1	f1		2	f2	\$	3	f3	&	4320	f4320	zzm	4321	f4321	~ax
	A	B	C	NO	EXP	FJF																																																					
1	f1	f2	f3	f379	f4020	class																																																					
2	0	0	0	0	0	1																																																					
3	0	0	0	0	0	1																																																					
370	0	0	0	0	0	0																																																					
371	0	0	0	0	0	0																																																					
1	f1																																																										
2	f2	\$																																																									
3	f3	&																																																									
4320	f4320	zzm																																																									
4321	f4321	~ax																																																									

4	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>DSX</th> <th>MUK</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f3222</td> <td>class</td> </tr> <tr> <td>2</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>3</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>370</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>371</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		A	B	C	DSX	MUK	1	f1	f2	f3	f3222	class	2	0	0	0	0	1	3	0	0	0	0	1	370	0	0	0	0	0	371	0	0	0	0	0	<table border="1"> <tbody> <tr> <td>1</td> <td>f1</td> <td></td> </tr> <tr> <td>2</td> <td>f2</td> <td>\$</td> </tr> <tr> <td>3</td> <td>f3</td> <td>.</td> </tr> <tr> <td>9343</td> <td>f9343</td> <td>zzmt</td> </tr> <tr> <td>9344</td> <td>f9344</td> <td>~axy</td> </tr> </tbody> </table>	1	f1		2	f2	\$	3	f3	.	9343	f9343	zzmt	9344	f9344	~axy
		A	B	C	DSX	MUK																																															
1	f1	f2	f3	f3222	class																																																
2	0	0	0	0	1																																																
3	0	0	0	0	1																																																
370	0	0	0	0	0																																																
371	0	0	0	0	0																																																
1	f1																																																				
2	f2	\$																																																			
3	f3	.																																																			
9343	f9343	zzmt																																																			
9344	f9344	~axy																																																			
5	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>ETA</th> <th>TCW</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f3901</td> <td>class</td> </tr> <tr> <td>2</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>3</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>370</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>371</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		A	B	C	ETA	TCW	1	f1	f2	f3	f3901	class	2	0	0	0	0	1	3	0	0	0	0	1	370	0	0	0	0	0	371	0	0	0	0	0	<table border="1"> <tbody> <tr> <td>1</td> <td>f1</td> <td></td> </tr> <tr> <td>2</td> <td>f2</td> <td>b</td> </tr> <tr> <td>3</td> <td>f3</td> <td>p</td> </tr> <tr> <td>13619</td> <td>f13619</td> <td>zzmta</td> </tr> <tr> <td>13620</td> <td>f13620</td> <td>~axy6</td> </tr> </tbody> </table>	1	f1		2	f2	b	3	f3	p	13619	f13619	zzmta	13620	f13620	~axy6
		A	B	C	ETA	TCW																																															
1	f1	f2	f3	f3901	class																																																
2	0	0	0	0	1																																																
3	0	0	0	0	1																																																
370	0	0	0	0	0																																																
371	0	0	0	0	0																																																
1	f1																																																				
2	f2	b																																																			
3	f3	p																																																			
13619	f13619	zzmta																																																			
13620	f13620	~axy6																																																			

To execute a batch script, transform the Ngram feature vector obtained from a CSV file to an ARFF file via Java code. Then run the batch file in from the command prompt. The batch file for the experiments is given in figure 5.55. Table 5.19 lists the commands being used to convert CSV to ARFF.

```

java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShakira.1n D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShakira1n.arff
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShakira.2n D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShakira2n.arff
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShakira.3n D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShakira3n.arff
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShakira.4n D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShakira4n.arff
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShakira.5n D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShakira5n.arff

```

Figure 5.55: Batch File Convert CSV to ARFF Shakira

Table 5.S: Details Parameter of Convert CSV to ARFF Shakira

Command	Description
-Xmx2G	RAM size
-cp	Execute
Classify.jar	Executable .jar file

psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF	Package and file name
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShakira.1n	Path for file vector, Ngram (n=1,2,3,4,5)
D:\PSM2\YouTube-Spam-Collection-v1\Shakira\featureVectorShakira.1n.arff	Path for file feature vector ARFF, Ngram (n=1,2,3,4,5)

5.3.4.4 Dataset LMFAO Feature Vector

Following the methods outlined in the feature vector module, this chapter will cover how to construct Ngram feature vectors for the LMFAO dataset. When carrying out this procedure, will use the Java code exported into a runnable jar file that it may use for batch execution of a program. Figures 5.56 resembles the batch file used to create the feature vectors for the experiments, respectively.

```

java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureDescriptorLmfao.1n D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmfao.1n
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\contentLmfao.1n 236 D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\renameFeatureVectorLMFAO.1n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureDescriptorLmfao.2n D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmfao.2n
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\contentLmfao.2n 236 D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\renameFeatureVectorLMFAO.2n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureDescriptorLmfao.3n D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmfao.3n
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\contentLmfao.3n 236 D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\renameFeatureVectorLMFAO.3n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureDescriptorLmfao.4n D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmfao.4n
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\contentLmfao.4n 236 D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\renameFeatureVectorLMFAO.4n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNgramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureDescriptorLmfao.5n D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmfao.5n
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\contentLmfao.5n 236 D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\renameFeatureVectorLMFAO.5n

```

Figure 5.56: Batch File for Ngram Feature Vector LMFAO

Table 5.20 provides a detailed explanation of each command that was used to create the Ngram feature vector. Next, run the batch file in command prompt to see whether it works properly.

Table 5.T: Details Parameter of Feature Vector LMFAO

Command	Description
-Xmx2G	RAM size

-cp	Execute
Classify.jar	Executable .jar file
psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector	Package and file name
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureDescriptorLmfao.1n	Path for file descriptor, Ngram (n=1,2,3,4,5)
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmfao.1n	Path for file feature vector, Ngram (n=1,2,3,4,5)
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\contentLmfao.1n	Path for file content, Ngram (n=1,2,3,4,5)
236	The last spam, boundary between spam and legitimate
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\renameFeatureVectorLmfao.1n	The path for rename feature descriptor, Ngram (n=1,2,3,4,5)

In Table 5.21, the experiments' batch files are shown.

Table 5.U: Generate Feature Vector LMFAO for Both Experiments

Experiment	Gram	Feature Vector	Rename Feature vector

Single classifier and Multiple Classifiers	1	<table border="1"> <thead> <tr> <th> </th> <th>A</th> <th>B</th> <th>C</th> <th>AI</th> <th>BK</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f35</td> <td>class</td> </tr> <tr> <td>2</td> <td>77</td> <td>6</td> <td>6</td> <td>25</td> <td>1</td> </tr> <tr> <td>3</td> <td>5</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>438</td> <td>2</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>439</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		A	B	C	AI	BK	1	f1	f2	f3	f35	class	2	77	6	6	25	1	3	5	0	0	0	1	438	2	0	0	0	0	439	1	0	0	0	0	<table border="1"> <tbody> <tr> <td>1</td> <td>f1</td> <td></td> </tr> <tr> <td>2</td> <td>f2</td> <td>!</td> </tr> <tr> <td>3</td> <td>f3</td> <td></td> </tr> <tr> <td>61</td> <td>f61</td> <td>z</td> </tr> <tr> <td>62</td> <td>f62</td> <td>~</td> </tr> </tbody> </table>	1	f1		2	f2	!	3	f3		61	f61	z	62	f62	~					
			A	B	C	AI	BK																																																				
		1	f1	f2	f3	f35	class																																																				
		2	77	6	6	25	1																																																				
		3	5	0	0	0	1																																																				
		438	2	0	0	0	0																																																				
439	1	0	0	0	0																																																						
1	f1																																																										
2	f2	!																																																									
3	f3																																																										
61	f61	z																																																									
62	f62	~																																																									
2	<table border="1"> <thead> <tr> <th> </th> <th>A</th> <th>B</th> <th>C</th> <th>EZ</th> <th>AJH</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f156</td> <td>class</td> </tr> <tr> <td>2</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>3</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>438</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>439</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		A	B	C	EZ	AJH	1	f1	f2	f3	f156	class	2	0	0	0	0	1	3	0	0	0	0	1	438	0	0	0	0	0	439	0	0	0	0	0	<table border="1"> <tbody> <tr> <td>1</td> <td>f1</td> <td></td> </tr> <tr> <td>2</td> <td>f2</td> <td></td> </tr> <tr> <td>3</td> <td>f3</td> <td>!</td> </tr> <tr> <td>942</td> <td>f942</td> <td>zy</td> </tr> <tr> <td>943</td> <td>f943</td> <td>zz</td> </tr> </tbody> </table>	1	f1		2	f2		3	f3	!	942	f942	zy	943	f943	zz						
		A	B	C	EZ	AJH																																																					
	1	f1	f2	f3	f156	class																																																					
	2	0	0	0	0	1																																																					
	3	0	0	0	0	1																																																					
	438	0	0	0	0	0																																																					
439	0	0	0	0	0																																																						
1	f1																																																										
2	f2																																																										
3	f3	!																																																									
942	f942	zy																																																									
943	f943	zz																																																									
3	<table border="1"> <thead> <tr> <th> </th> <th>A</th> <th>B</th> <th>C</th> <th>OK</th> <th>DUD</th> <th>EBM</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f401</td> <td>f3254</td> <td>class</td> </tr> <tr> <td>2</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>3</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>438</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>439</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		A	B	C	OK	DUD	EBM	1	f1	f2	f3	f401	f3254	class	2	0	0	0	0	0	1	3	0	0	0	0	0	1	438	0	0	0	0	0	0	439	0	0	0	0	0	0	<table border="1"> <tbody> <tr> <td>1</td> <td>f1</td> <td></td> </tr> <tr> <td>2</td> <td>f2</td> <td></td> </tr> <tr> <td>3</td> <td>f3</td> <td><</td> </tr> <tr> <td>3443</td> <td>f3443</td> <td>zz</td> </tr> <tr> <td>3444</td> <td>f3444</td> <td>zzz</td> </tr> </tbody> </table>	1	f1		2	f2		3	f3	<	3443	f3443	zz	3444	f3444	zzz
		A	B	C	OK	DUD	EBM																																																				
	1	f1	f2	f3	f401	f3254	class																																																				
	2	0	0	0	0	0	1																																																				
	3	0	0	0	0	0	1																																																				
	438	0	0	0	0	0	0																																																				
439	0	0	0	0	0	0																																																					
1	f1																																																										
2	f2																																																										
3	f3	<																																																									
3443	f3443	zz																																																									
3444	f3444	zzz																																																									
4	<table border="1"> <thead> <tr> <th> </th> <th>A</th> <th>B</th> <th>C</th> <th>DFA</th> <th>ITP</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f2861</td> <td>class</td> </tr> <tr> <td>2</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>3</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>438</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>439</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		A	B	C	DFA	ITP	1	f1	f2	f3	f2861	class	2	0	0	0	0	1	3	0	0	0	0	1	438	0	0	0	0	0	439	0	0	0	0	0	<table border="1"> <tbody> <tr> <td>1</td> <td>f1</td> <td></td> </tr> <tr> <td>2</td> <td>f2</td> <td></td> </tr> <tr> <td>3</td> <td>f3</td> <td><</td> </tr> <tr> <td>6618</td> <td>f6618</td> <td>zzz</td> </tr> <tr> <td>6619</td> <td>f6619</td> <td>zzzz</td> </tr> </tbody> </table>	1	f1		2	f2		3	f3	<	6618	f6618	zzz	6619	f6619	zzzz						
		A	B	C	DFA	ITP																																																					
	1	f1	f2	f3	f2861	class																																																					
	2	0	0	0	0	1																																																					
	3	0	0	0	0	1																																																					
	438	0	0	0	0	0																																																					
439	0	0	0	0	0																																																						
1	f1																																																										
2	f2																																																										
3	f3	<																																																									
6618	f6618	zzz																																																									
6619	f6619	zzzz																																																									
5	<table border="1"> <thead> <tr> <th> </th> <th>A</th> <th>B</th> <th>C</th> <th>BOW</th> <th>MKM</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f1765</td> <td>class</td> </tr> <tr> <td>2</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>3</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>438</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>439</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		A	B	C	BOW	MKM	1	f1	f2	f3	f1765	class	2	0	0	0	0	1	3	0	0	0	0	1	438	0	0	0	0	0	439	0	0	0	0	0	<table border="1"> <tbody> <tr> <td>1</td> <td>f1</td> <td></td> </tr> <tr> <td>2</td> <td>f2</td> <td></td> </tr> <tr> <td>3</td> <td>f3</td> <td><</td> </tr> <tr> <td>9085</td> <td>f9085</td> <td>zzzz</td> </tr> <tr> <td>9086</td> <td>f9086</td> <td>zzzzz</td> </tr> </tbody> </table>	1	f1		2	f2		3	f3	<	9085	f9085	zzzz	9086	f9086	zzzzz						
		A	B	C	BOW	MKM																																																					
	1	f1	f2	f3	f1765	class																																																					
	2	0	0	0	0	1																																																					
	3	0	0	0	0	1																																																					
	438	0	0	0	0	0																																																					
439	0	0	0	0	0																																																						
1	f1																																																										
2	f2																																																										
3	f3	<																																																									
9085	f9085	zzzz																																																									
9086	f9086	zzzzz																																																									

To execute a batch script, transform the Ngram feature vector obtained from a CSV file to an ARFF file via Java code. Then run the batch file in from the command prompt. The batch file for the experiments is given in figure 5.57. Table 5.22 lists the commands being used to convert CSV to ARFF.

```

java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmfao.1n D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmfao1n.arff
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmfao.2n D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmfao2n.arff
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmfao.3n D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmfao3n.arff
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmfao.4n D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmfao4n.arff
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmfao.5n D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmfao5n.arff

```

Figure 5.57: Batch File Convert CSV to ARFF LMFAO

Table 5.V: Details Parameter of Convert CSV to ARFF LMFAO

Command	Description
-Xmx2G	RAM size
-cp	Execute
Classify.jar	Executable .jar file
psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF	Package and file name
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmfao.1n	Path for file vector, Ngram (n=1,2,3,4,5)
D:\PSM2\YouTube-Spam-Collection-v1\LMFAO\featureVectorLmfao1n.arff	Path for file feature vector ARFF, Ngram (n=1,2,3,4,5)

5.3.4.5 Dataset Katy Perry Feature Vector

Following the methods outlined in the feature vector module, this chapter will cover how to construct Ngram feature vectors for the Katy Perry dataset. When carrying out this procedure, will use the Java code exported into a runnable jar file that it may use for batch execution of a program. Figures 5.58 resembles the batch file used to create the feature vectors for the experiments, respectively.

```

java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureDescriptorKatyPerry.1n D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry.1n
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\contentKatyPerry.1n 175 D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\renameFeatureVectorKatyPerry.1n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureDescriptorKatyPerry.2n D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry.2n
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\contentKatyPerry.2n 175 D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\renameFeatureVectorKatyPerry.2n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureDescriptorKatyPerry.3n D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry.3n
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\contentKatyPerry.3n 175 D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\renameFeatureVectorKatyPerry.3n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureDescriptorKatyPerry.4n D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry.4n
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\contentKatyPerry.4n 175 D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\renameFeatureVectorKatyPerry.4n
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureDescriptorKatyPerry.5n D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry.5n
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\contentKatyPerry.5n 175 D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\renameFeatureVectorKatyPerry.5n

```

Figure 5.58: Batch File for Ngram Feature Vector Katy Perry

Table 5.23 provides a detailed explanation of each command that was used to create the Ngram feature vector. Next, run the batch file in command prompt to see whether it works properly.

Table 5.W: Details Parameter of Feature Vector Katy Perry

Command	Description
-Xmx2G	RAM size
-cp	Execute
Classify.jar	Executable .jar file
psm.sem32021.syazaliyanamuhamadshapee.GenerateNGramFeatureVector	Package and file name
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureDescriptorKatyPerry.1n	Path for file descriptor, Ngram (n=1,2,3,4,5)
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry.1n	Path for file feature vector, Ngram (n=1,2,3,4,5)
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\contentKatyPerry.1n	Path for file content, Ngram (n=1,2,3,4,5)

175	The last spam, boundary between spam and legitimate
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry \renameFeatureVectorLKatyPerry.1n	The path for rename feature descriptor, Ngram (n=1,2,3,4,5)

In Table 5.24, the experiments' batch files are shown.

Table 5.X: Generate Feature Vector Katy Perry for Both Experiments

Experiment	Gram	Feature Vector	Rename Feature vector																																																			
Single classifier and Multiple Classifiers	1	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>AA</th> <th>BK</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f27</td> <td>class</td> </tr> <tr> <td>2</td> <td>19</td> <td>1</td> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>3</td> <td>9</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>350</td> <td>35</td> <td>3</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>351</td> <td>12</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> </tr> </tbody> </table>		A	B	C	AA	BK	1	f1	f2	f3	f27	class	2	19	1	0	1	1	3	9	0	0	1	1	350	35	3	0	0	0	351	12	0	0	1	0	<table border="1"> <tbody> <tr> <td>1</td> <td>f1</td> <td></td> </tr> <tr> <td>2</td> <td>f2</td> <td>!</td> </tr> <tr> <td>3</td> <td>f3</td> <td></td> </tr> <tr> <td>61</td> <td>f61</td> <td>y</td> </tr> <tr> <td>62</td> <td>f62</td> <td>z</td> </tr> </tbody> </table>	1	f1		2	f2	!	3	f3		61	f61	y	62	f62	z
		A	B	C	AA	BK																																																
	1	f1	f2	f3	f27	class																																																
2	19	1	0	1	1																																																	
3	9	0	0	1	1																																																	
350	35	3	0	0	0																																																	
351	12	0	0	1	0																																																	
1	f1																																																					
2	f2	!																																																				
3	f3																																																					
61	f61	y																																																				
62	f62	z																																																				
2	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>CD</th> <th>BNW</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f82</td> <td>class</td> </tr> <tr> <td>2</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>3</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>350</td> <td>2</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>351</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		A	B	C	CD	BNW	1	f1	f2	f3	f82	class	2	1	0	0	0	1	3	0	0	0	0	1	350	2	0	0	0	0	351	0	0	0	0	0	<table border="1"> <tbody> <tr> <td>1</td> <td>f1</td> <td></td> </tr> <tr> <td>2</td> <td>f2</td> <td>!</td> </tr> <tr> <td>3</td> <td>f3</td> <td></td> </tr> <tr> <td>1737</td> <td>f1737</td> <td>zy</td> </tr> <tr> <td>1738</td> <td>f1738</td> <td>zz</td> </tr> </tbody> </table>	1	f1		2	f2	!	3	f3		1737	f1737	zy	1738	f1738	zz	
	A	B	C	CD	BNW																																																	
1	f1	f2	f3	f82	class																																																	
2	1	0	0	0	1																																																	
3	0	0	0	0	1																																																	
350	2	0	0	0	0																																																	
351	0	0	0	0	0																																																	
1	f1																																																					
2	f2	!																																																				
3	f3																																																					
1737	f1737	zy																																																				
1738	f1738	zz																																																				
3	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>ADK</th> <th>JMN</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f791</td> <td>class</td> </tr> <tr> <td>2</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>3</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>350</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>351</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		A	B	C	ADK	JMN	1	f1	f2	f3	f791	class	2	0	0	0	0	1	3	0	0	0	0	1	350	0	0	0	0	0	351	0	0	0	0	0	<table border="1"> <tbody> <tr> <td>1</td> <td>f1</td> <td></td> </tr> <tr> <td>2</td> <td>f2</td> <td>!</td> </tr> <tr> <td>3</td> <td>f3</td> <td></td> </tr> <tr> <td>7110</td> <td>f7110</td> <td>zzq</td> </tr> <tr> <td>7111</td> <td>f7111</td> <td>zzz</td> </tr> </tbody> </table>	1	f1		2	f2	!	3	f3		7110	f7110	zzq	7111	f7111	zzz	
	A	B	C	ADK	JMN																																																	
1	f1	f2	f3	f791	class																																																	
2	0	0	0	0	1																																																	
3	0	0	0	0	1																																																	
350	0	0	0	0	0																																																	
351	0	0	0	0	0																																																	
1	f1																																																					
2	f2	!																																																				
3	f3																																																					
7110	f7110	zzq																																																				
7111	f7111	zzz																																																				

	4	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>BUN</th> <th>SWU</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f1912</td> <td>class</td> </tr> <tr> <td>2</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>3</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>350</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>351</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		A	B	C	BUN	SWU	1	f1	f2	f3	f1912	class	2	0	0	0	0	1	3	0	0	0	0	1	350	0	0	0	0	0	351	0	0	0	0	0	<table border="1"> <tbody> <tr> <td>1</td> <td>f1</td> <td></td> </tr> <tr> <td>2</td> <td>f2</td> <td>!</td> </tr> <tr> <td>3</td> <td>f3</td> <td>#</td> </tr> <tr> <td>13461</td> <td>f13461</td> <td>zzqo</td> </tr> <tr> <td>13462</td> <td>f13462</td> <td>zzz</td> </tr> </tbody> </table>	1	f1		2	f2	!	3	f3	#	13461	f13461	zzqo	13462	f13462	zzz
		A	B	C	BUN	SWU																																																
1	f1	f2	f3	f1912	class																																																	
2	0	0	0	0	1																																																	
3	0	0	0	0	1																																																	
350	0	0	0	0	0																																																	
351	0	0	0	0	0																																																	
1	f1																																																					
2	f2	!																																																				
3	f3	#																																																				
13461	f13461	zzqo																																																				
13462	f13462	zzz																																																				
	5	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>AWS</th> <th>GEJ</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>f1</td> <td>f2</td> <td>f3</td> <td>f1293</td> <td>f4872</td> </tr> <tr> <td>2</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>3</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>350</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>351</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		A	B	C	AWS	GEJ	1	f1	f2	f3	f1293	f4872	2	0	0	0	0	1	3	0	0	0	0	1	350	0	0	0	0	0	351	0	0	0	0	0	<table border="1"> <tbody> <tr> <td>1</td> <td>f1</td> <td></td> </tr> <tr> <td>2</td> <td>f2</td> <td>!</td> </tr> <tr> <td>3</td> <td>f3</td> <td>b</td> </tr> <tr> <td>18781</td> <td>f18781</td> <td>zzqo/</td> </tr> <tr> <td>18782</td> <td>f18782</td> <td>zzz</td> </tr> </tbody> </table>	1	f1		2	f2	!	3	f3	b	18781	f18781	zzqo/	18782	f18782	zzz
	A	B	C	AWS	GEJ																																																	
1	f1	f2	f3	f1293	f4872																																																	
2	0	0	0	0	1																																																	
3	0	0	0	0	1																																																	
350	0	0	0	0	0																																																	
351	0	0	0	0	0																																																	
1	f1																																																					
2	f2	!																																																				
3	f3	b																																																				
18781	f18781	zzqo/																																																				
18782	f18782	zzz																																																				

To execute a batch script, transform the Ngram feature vector obtained from a CSV file to an ARFF file via Java code. Then run the batch file in from the command prompt. The batch file for the experiments is given in figure 5.59. Table 5.25 lists the commands being used to convert CSV to ARFF.

```
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry.1n D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry1n.arff
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry.2n D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry2n.arff
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry.3n D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry3n.arff
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry.4n D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry4n.arff
java -Xmx2G -cp Classify.jar psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry.5n D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry5n.arff
```

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 5.59: Batch File Convert CSV to ARFF Katy Perry

Table 5.Y: Details Parameter of Convert CSV to ARFF Katy Perry

Command	Description
-Xmx2G	RAM size
-cp	Execute
Classify.jar	Executable .jar file
psm.sem32021.syazaliyanamuhamadshapee.ConvertCSVtoARFF	Package and file name

D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry.1n	Path for file vector, Ngram (n=1,2,3,4,5)
D:\PSM2\YouTube-Spam-Collection-v1\KatyPerry\featureVectorKatyPerry1n.arff	Path for file feature vector ARFF, Ngram (n=1,2,3,4,5)

5.3.5 Data Training and Testing

This part necessitates the use of four classes: split dataset, model file, prediction file, and script execution for the test, among other things. The model would then be input into the algorithm code, with the prediction accuracy produced as the result of the algorithm.

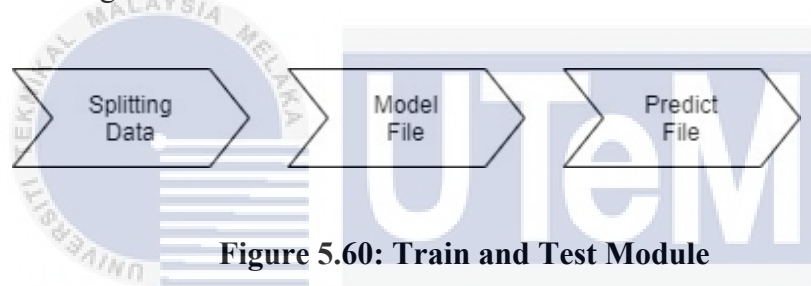


Figure 5.60: Train and Test Module

5.3.5.1 Splitting Data

There is a variable quantity of spam and legitimate data in each YouTube spam dataset collection, categorized in the same way as the table.

Table 5.Z: Sorted Number of Instances of Each Dataset

Dataset	Spam		Legitimate	
	Start	End	Start	End
Eminem	0	244	245	453
Psy	0	174	175	350

Shakira	0	174	175	350
LMFAO	0	235	236	202
Katy Perry	0	173	174	370

Following then, both train and test data for the dataset are 80% 20% dataset plotting sin size based on the total number of instances to every dataset as shown earlier in the table. Table 5.27 represents the train and test data for every dataset. As a result, the pattern of the training and testing data about each dataset is shown in figure 5.61.

Table 5.AA: Training and Testing Size

Dataset	Train	Test
Eminem	176	44
Psy	132	33
Shakira	136	34
LMFAO	160	40
Katy Perry	140	35

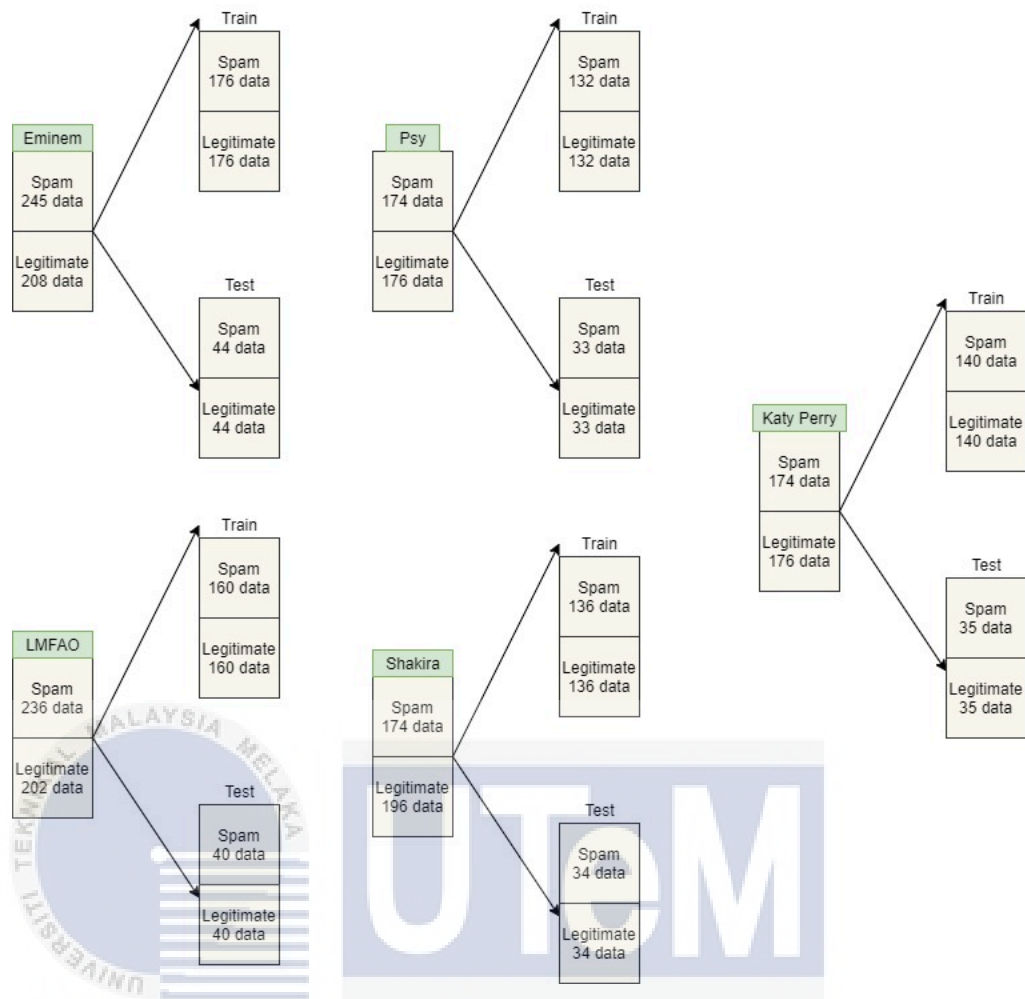


Figure 5.61: tern and Structure Training and Testing Data

The selection of instances for training or testing will be based on a data split of over 80% and 20%. The researchers previously discussed this situation in earlier chapters. To determine whether instances are train or test, they are assigned a sequence number, divided to create test and train data.

5.3.5.2 Model File

Instances to develop a train scale model for BOW, IG, and CS feature selection. The training scale gives a value to weightage, which is then used to generate prediction files for trains. The train.arff files will be used in conjunction with the train.scale and processed in SVM to create the train.

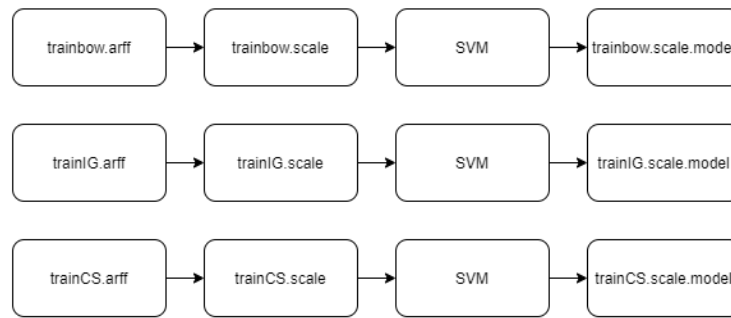


Figure 5.62: Train Process of Model and Scale

5.3.5.3 Predict File

As previously stated, weights are given to the models, which they will utilise in the test.arff file to generate the prediction file. The weightage value from the training scale will be used in the test.arff file for prediction in the test.arff file. It also contains information on the accuracy of BOW, IG, and CS predictions. Figure 5.63 portrays the steps involved in creating a predicted file for use in modelling.

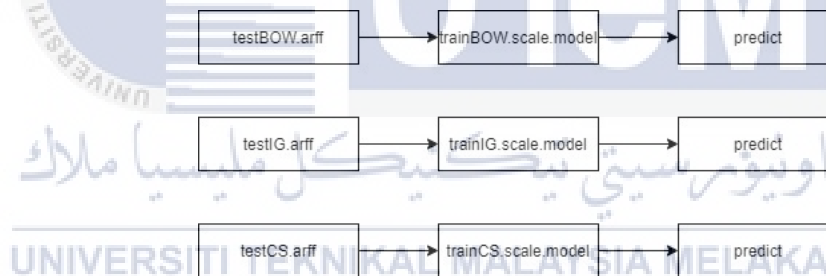


Figure 5.63: Process Generating Predict File

5.4 Script Execution

5.4.1 Eminem Dataset Scripts

The code runs the program ten times and produces a variety of files. The experiment will be repeated five times for 1n-5n, depending on the grams. Figures 5.64 show the run command for the Eminem dataset of experiment 1 and figure 5.65 shows the run command for the experiment 2.

```
D:\PSM2\Experiment\liblinear-2.1\liblinear-2.1\windows\ D:\PSM2\Experiment\libsvm-
3.23\libsvm-3.23\ "C:\Program Files\gnuplot\bin\\" C:\Users\ASUS\AppData\Local\Programs
\Python\Python39\ 453 176 44 0 244 245 452 1 5 1 10
```

Figure 5.64: Run code program EMINEM experiment 1

Figure 5.65 represents experiment 2 in ranking accuracy Ngram, $n=1,2,3,4,5$ by using weight. The EMINEM dataset shows that $3n$, $2n$, $1n$, $4n$, $5n$ are the ranking accuracy and $3n$ are the highest.

0.33 0.5 1 0.25 0.2

Figure 5.65: Run code program EMINEM experiment 2

Execution may take many times, depending on the number of runs and the size of the gram. The program produces various files depending on the argument, including results, predict, and scale. Table 5.28 provides information on every command used in the figure.

Table 5.28: Parameter description EMINEM

Experiment	Command	Description
Single Classifier	liblinear-2.1	Liblinear path
	libsvm-3.21	Libsvm path
	gnuplot	GNU plot path
	Python39	Python path
	453	Total number of instances
	176	Train size
	44	Test size
	0 244	Initial and range number of instances for class 1
	245 452	Initial and range number of instances for class 0
	1 5	Start and end gram (1n-5n)
	1 10	Start and end run (Number of run train)
Multiple Classifier	0.3 0.5 1 0.25 0.2	Weight to compare ranking accuracy (1n-5n)

5.4.2 Psy Dataset Scripts

The code runs the program ten times and produces a variety of files. The experiment will be repeated five times for 1n-5n, depending on the grams. Figures 5.66 show the run command for the Psy dataset of experiment 1 and figure 5.67 shows the run command for the experiment 2.

```
D:\PSM2\Experiment\liblinear-2.1\liblinear-2.1\windows\ D:\PSM2\Experiment\libsvm-3.23\libsvm-3.23\ "C:\Program Files\gnuplot\bin\\" C:\Users\ASUS\AppData\Local\Programs\Python\Python39\ 350 132 33 0 174 175 349 1 5 1 10
```

Figure 5.66: Run code program Psy experiment 1

Figure 5.67 represents experiment 2 in ranking accuracy Ngram, n=1,2,3,4,5 by using weight. The Psy dataset shows that 3n, 4n, 5n, 2n, 1n are the ranking accuracy and 3n are the highest.



Figure 5.67: Run code program Psy experiment 2

Execution may take many times, depending on the number of runs and the size of the gram. The program produces various files depending on the argument, including results, predict, and scale. Table 5.29 provides information on every command used in the figure.

Table 5.CC: Parameter description Psy

Experiment	Command	Description
Single Classifier	liblinear-2.1	Liblinear path
	libsvm-3.21	Libsvm path
	gnuplot	GNU plot path
	Python39	Python path
	350	Total number of instances
	132	Train size
	33	Test size
	0 174	Initial and range number of instances for class 1
	175 349	Initial and range number of instances for class 0

	1 5	Start and end gram (1n-5n)
	1 10	Start and end run (Number of run train)
Multiple Classifier	0.2 0.25 1 0.5 0.33	Weight to compare ranking accuracy (1n-5n)

5.4.3 Shakira Dataset Scripts

The code runs the program ten times and produces a variety of files. The experiment will be repeated five times for 1n-5n, depending on the grams. Figures 5.68 show the run command for the Shakira dataset of experiment 1 and figure 5.69 shows the run command for the experiment 2.

```
D:\PSM2\Experiment\liblinear-2.1\liblinear-2.1\windows\ D:\PSM2\Experiment\libsvm-3.23\libsvm-3.23\ "C:\Program Files\gnuplot\bin\\" C:\Users\ASUS\AppData\Local\Programs\Python\Python39\ 370 136 34 0 173 174 369 1 5 1 10
```

Figure 5.68: Run code program Shakira experiment 1

Figure 5.69 represents experiment 2 in ranking accuracy Ngram, n=1,2,3,4,5 by using weight. The Shakira dataset shows that 3n, 2n, 4n, 5n, 1n are the ranking accuracy and 3n are the highest.

```
0.2 0.5 1 0.33 0.25
```

Figure 5.69: Run code program Shakira experiment 2

Execution may take many times, depending on the number of runs and the size of the gram. The program produces various files depending on the argument, including results, predict, and scale. Table 5.30 provides information on every command used in the figure.

Table 5.DD: Parameter description Shakira for experiment 1

Experiment	Command	Description
Single Classifier	liblinear-2.1	Liblinear path
	libsvm-3.21	Libsvm path
	gnuplot	GNU plot path
	Python39	Python path

	370	Total number of instances
	136	Train size
	34	Test size
	0 173	Initial and range number of instances for class 1
	174 369	Initial and range number of instances for class 0
	1 5	Start and end gram (1n-5n)
	1 10	Start and end run (Number of run train)
Multiple Classifier	0.2 0.5 1 0.33 0.25	Weight to compare ranking accuracy (1n-5n)

5.4.4 LMFAO Dataset Scripts

The code runs the program ten times and produces a variety of files. The experiment will be repeated five times for 1n-5n, depending on the grams. Figures 5.70 show the run command for the LMFAO dataset of experiment 1 and figure 5.71 shows the run command for the experiment 2.

```
D:\PSM2\Experiment\liblinear-2.1\liblinear-2.1\windows\ D:\PSM2\Experiment\libsvm-3.23\libsvm-3.23\ "C:\Program Files\gnuplot\bin\\" C:\Users\ASUS\AppData\Local\Programs\Python\Python39\ 438 160 40 0 235 236 437 1 5 1 10
```

Figure 5.70: Run code program LMFAO experiment 1

Figure 5.71 represents experiment 2 in ranking accuracy Ngram, n=1,2,3,4,5 by using weight. The LMFAO dataset shows that 4n, 5n, 3n, 2n, 1n are the ranking accuracy and 4n are the highest.

```
0.2 0.25 0.33 1 0.5
```

Figure 5.71: Run code program LMFAO experiment 2

Execution may take many times, depending on the number of runs and the size of the gram. The program produces various files depending on the argument, including results, predict, and scale. Table 5.31 provides information on every command used in the figure.

Table 5.EE: Parameter description LMFAO for experiment 1

Experiment	Command	Description
Single Classifier	liblinear-2.1	Liblinear path
	libsvm-3.21	Libsvm path
	gnuplot	GNU plot path
	Python39	Python path
	438	Total number of instances
	160	Train size
	40	Test size
	0 235	Initial and range number of instances for class 1
	236 437	Initial and range number of instances for class 0
	1 5	Start and end gram (1n-5n)
	1 10	Start and end run (Number of run train)
Multiple Classifier	0.2 0.25 0.33 1 0.5	Weight to compare ranking accuracy (1n-5n)

5.4.5 Katy Perry Dataset Scripts

The code runs the program ten times and produces a variety of files. The experiment will be repeated five times for 1n-5n, depending on the grams. Figures 5.72 show the run command for the Katy Perry dataset of experiment 1 and figure 5.73 shows the run command for the experiment 2.

```
D:\PSM2\Experiment\liblinear-2.1\liblinear-2.1\windows\ D:\PSM2\Experiment\libsvm-3.23\libsvm-3.23\ "C:\Program Files\gnuplot\bin\\" C:\Users\ASUS\AppData\Local\Programs\Python\Python39\ 350 140 35 0 174 175 349 1 5 1 10
```

Figure 5.72: Run code program Katy Perry experiment 1

Figure 5.73 represents experiment 2 in ranking accuracy Ngram, n=1,2,3,4,5 by using weight. The Katy Perry dataset shows that 3n, 4n, 2n, 5n, 1n are the ranking accuracy and 3n are the highest.

0.2 0.33 1 0.5 0.25

Figure 5.73: Run code program Katy Perry experiment 2

Execution may take many times, depending on the number of runs and the size of the gram. The program produces various files depending on the argument, including results, predict, and scale. Table 5.32 provides information on every command used in the figure.

Table 5.FF: Parameter description Katy Perry for experiment 1

Experiment	Command	Description
Single Classifier	liblinear-2.1	Liblinear path
	libsvm-3.21	Libsvm path
	gnuplot	GNU plot path
	Python39	Python path
	350	Total number of instances
	140	Train size
	35	Test size
	0 174	Initial and range number of instances for class 1
	175 350	Initial and range number of instances for class 0
	1 5	Start and end gram (1n-5n)
	1 10	Start and end run (Number of run train)
Multiple Classifier	0.2 0.33 1 0.5 0.25	Weight to compare ranking accuracy (1n-5n)

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

5.5 Result

This experiment will be carried out in two phases, with the best predictions technique being determined for each step and the final result. The first step consists of an experiment with a single classifier, followed by an experiment with multiple classifiers in the second step. When it comes to model outcomes, it is split into three kinds of results: BOW, Chi-Square, and Information Gain. As a result, the tests with grams ranging from $n=1,2,3,4,5$ are carried out in both experiments utilizing Linear and RBF Kernels. For each of the various datasets, Eminem, Katy Perry, LMFAO, Psy, and Shakira.

5.5.1 Eminem Dataset Result

The estimated number of instances of YouTube spam for the Eminem dataset is 453, with 245 instances being spam and 208 instances being legitimate.

5.5.1.1 Attribute

It takes 220 instances to achieve the BOW's accuracy for every Ngram in both experiments. Thus, a total of 220 instances are used. Process feature selection was carried out using the specified attributes and ranks, with the degree of relevance for predictions being referred to as a benchmark. When it comes to the chosen attributes, the researcher previously explained in the previous chapter that the study had chosen 80 percent of the total set of attributes for this particular project before starting. Table 5.33 indicates the proportion of 80 percent characteristics for both the algorithm's single classifier and multiple classifier versions.

Table 5.GG: 80% of Total Instances Attributes Eminem

Gram	Total Attribute	80% of Total Attribute
1	462	370
2	1472	1178
3	5059	4048
4	11074	8860
5	16871	13497

5.5.1.2 10 Runs

Figure 5.74 signifies the ten run-time train and test results produced by BOW, IG, and CS for each kernel. TrainBOWL, TrainIGL and TrainCSL are the

abbreviations for train BOW, IG and CS linear kernel, meanwhile TrainBOWR, TrainIGR and TrainCSR are abbreviations for train BOW, IG and CS RBF kernel. The same goes for the others. TestBOWL, TestIGL and TestCSL are abbreviations for linear test kernel, and TestBOWR, TestIGR and TestCSR are abbreviations for test RBF kernel.

Run	Gram	TrainBOWL	TestBOWL	TrainBOWR	TestBOWR	TrainIGL	TestIGL	TrainIGR	TestIGR	TrainCSL	TestCSL	TrainCSR	TestCR
1	1	0.994318	0.977273	99.431818	97.727273	0.994318	0.977273	99.431818	97.727273	0.994318	0.977273	99.431818	97.727273
2	1	1.000000	1.000000	100.000000	100.000000	1.000000	1.000000	100.000000	100.000000	1.000000	1.000000	100.000000	100.000000
3	1	1.000000	1.000000	100.000000	100.000000	1.000000	1.000000	100.000000	100.000000	1.000000	1.000000	100.000000	100.000000
4	1	1.000000	1.000000	100.000000	100.000000	1.000000	1.000000	100.000000	100.000000	1.000000	1.000000	100.000000	100.000000
5	1	0.994318	1.000000	99.431818	100.000000	0.994318	1.000000	99.431818	100.000000	0.994318	1.000000	99.431818	100.000000
6	1	0.994318	0.977273	99.431818	100.000000	0.994318	0.977273	99.431818	100.000000	0.994318	0.977273	99.431818	100.000000
7	1	1.000000	1.000000	100.000000	100.000000	1.000000	1.000000	100.000000	100.000000	1.000000	1.000000	100.000000	100.000000
8	1	1.000000	0.977273	100.000000	97.727273	1.000000	0.977273	100.000000	97.727273	1.000000	0.977273	100.000000	97.727273
9	1	1.000000	0.977273	100.000000	97.727273	1.000000	0.977273	100.000000	97.727273	1.000000	0.977273	100.000000	97.727273
10	1	0.994318	1.000000	99.431818	100.000000	0.994318	1.000000	99.431818	100.000000	0.994318	1.000000	99.431818	100.000000
average		0.997727	0.990909	99.772727	99.318182	0.997727	0.990909	99.772727	99.318182	0.997727	0.990909	99.772727	99.318182

Figure 5.74: Example of 10 Runs Experiment for 1 gram by Eminem

In figure 5.75, a graph bar is produced based on the data from the table. By looking at the results, we will be able to calculate the average accuracy of each kernel that is run by BOW, IG, and CS, which will do in Excel.

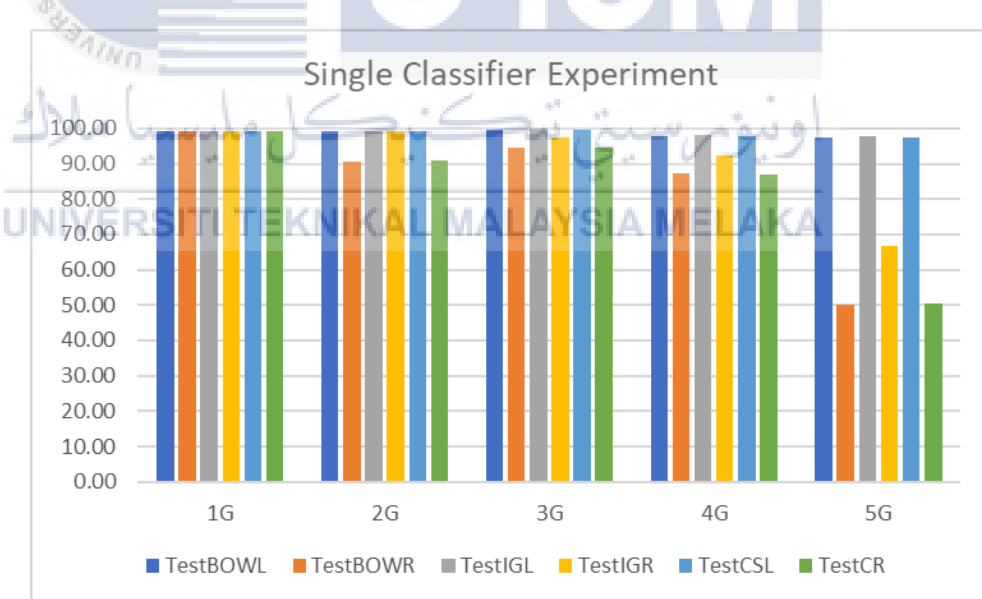


Figure 5.75: Graph of Experiment 1 Eminem

5.5.1.3 Accuracy Table

Following is a summary of the final result of EMINEM datasets for experiments single and multiple classifiers, categorized every model and ideal kernel based on the five gram of the experiments, shown in this subchapter. Table 5.34 contains the findings of experiment 1 EMINEM single classifier accuracy and figure 5.76 represents the findings of experiment 2 EMINEM multiple classifier.

Table 5.HH: Single Classifier of EMINEM Accuracy

EMINEM Experiment 1						
Gram	Bag Of Word (BOW)		Information Gain (IG)		Chi-Square (CS)	
	Linear(%)	RBF(%)	Linear(%)	RBF(%)	Linear(%)	RBF(%)
1	99.09	99.32	99.09	99.32	99.09	99.32
2	99.32	90.45	99.32	99.09	99.32	90.91
3	99.97	94.55	100.00	97.27	99.77	94.55
4	97.95	87.27	98.18	92.50	97.95	86.82
5	97.27	50.23	97.73	66.59	97.27	50.45

Run	Product	Mean	MV
1	100.00%	100.00%	100.00%
2	100.00%	100.00%	100.00%
3	100.00%	100.00%	100.00%
4	100.00%	100.00%	100.00%
5	100.00%	100.00%	100.00%
6	100.00%	100.00%	100.00%
7	100.00%	100.00%	100.00%
8	100.00%	100.00%	100.00%
9	97.73%	97.73%	100.00%
10	100.00%	100.00%	100.00%
Average	99.77%	99.77%	100.00%

Figure 5.76: Multiple Classifier of Eminem Accuracy

The results of the Ngram single classifier using three alternative models are as shown in Table 5.34. It displays the percentages of BOW, IG, and CS using the linear kernel and RBF. The linear 3 gram IG linear has a precision of 100.00% more than the other gram and kernel for each model. As a consequence of this finding, the experiment with a single classifier will use 3 gram of IG linear as the benchmark.

Meanwhile, the results for Ngram multiple classifiers using three alternative weights like product, mean, and majority voting (MV) as shown in figure 5.76. Experiment 2 has a precision of 100% in MV compare to the product and mean. It compares the percentage based on the ranking accuracy for Ngram, $n=1,2,3,4,5$.

5.5.2 Psy Dataset Result

The estimated number of instances of YouTube spam for the Psy dataset is 350, with 175 instances being spam and also, 175 instances being legitimate.

5.5.2.1 Attribute

It takes 175 instances to achieve the BOW's accuracy for every Ngram in both experiments. Thus, a total of 175 instances are used. Process feature selection was carried out using the specified attributes and ranks, with the degree of relevance for predictions being referred to as a benchmark. When it comes to the chosen attributes, the researcher previously explained in the previous chapter that the study had chosen 80 percent of the total set of attributes for this particular project before starting. Table

5.36 indicates the proportion of 80 percent characteristics for both the algorithm's single classifier and multiple classifier versions.

Table 5.II: 80% of Total Instances Attributes Psy

Gram	Total Attribute	80% of Total Attribute
1	404	324
2	1717	1374
3	5852	4682
4	10936	8749
5	15094	12076

5.5.2.2 10 Runs

Figure 5.77 signifies the ten run-time train and test results produced by BOW, IG, and CS for each kernel. TrainBOWL, TrainIGL and TrainCSL are the abbreviations for train BOW, IG and CS linear kernel, meanwhile TrainBOWR, TrainIGR and TrainCSR are abbreviations for train BOW, IG and CS RBF kernel. The same goes for the others. TestBOWL, TestIGL and TestCSL are abbreviations for linear test kernel, and TestBOWR, TestIGR and TestCSR are abbreviations for test RBF kernel.

Run	Gram	TrainBOWL	TestBOWL	TrainBOWR	TestBOWR	TrainIGL	TestIGL	TrainIGR	TestIGR	TrainCSL	TestCSL	TrainCSR	TestCSR
1	1	1.00	0.91	100.00	84.85	1.00	0.91	100.00	90.91	1.00	0.91	100.00	84.85
2	1	1.00	0.88	100.00	87.88	1.00	0.88	100.00	84.85	1.00	0.88	100.00	84.85
3	1	1.00	0.67	100.00	66.67	1.00	0.67	100.00	75.76	1.00	0.67	100.00	66.67
4	1	1.00	0.88	100.00	84.85	1.00	0.88	100.00	90.91	1.00	0.88	100.00	87.88
5	1	1.00	0.73	100.00	72.73	1.00	0.73	100.00	72.73	1.00	0.73	100.00	72.73
6	1	1.00	0.82	100.00	81.82	1.00	0.82	100.00	81.82	1.00	0.82	100.00	84.85
7	1	1.00	0.82	100.00	87.88	1.00	0.82	100.00	87.88	1.00	0.82	100.00	87.88
8	1	1.00	0.97	100.00	60.61	1.00	0.97	100.00	81.82	1.00	0.97	100.00	60.61
9	1	1.00	0.82	100.00	48.48	1.00	0.82	100.00	48.48	1.00	0.82	100.00	48.48
10	1	1.00	0.76	100.00	72.73	1.00	0.76	100.00	72.73	1.00	0.76	100.00	69.70
1G		1.00	0.82	100.00	74.85	1.00	0.82	100.00	78.79	1.00	0.82	100.00	74.85

Figure 5.77: Example of 10 Runs Experiment for 1 gram by Psy

In figure 5.78, a graph bar is produced based on the data from the table. By looking at the results, we will be able to calculate the average accuracy of each kernel that is run by BOW, IG, and CS, which will do in Excel.

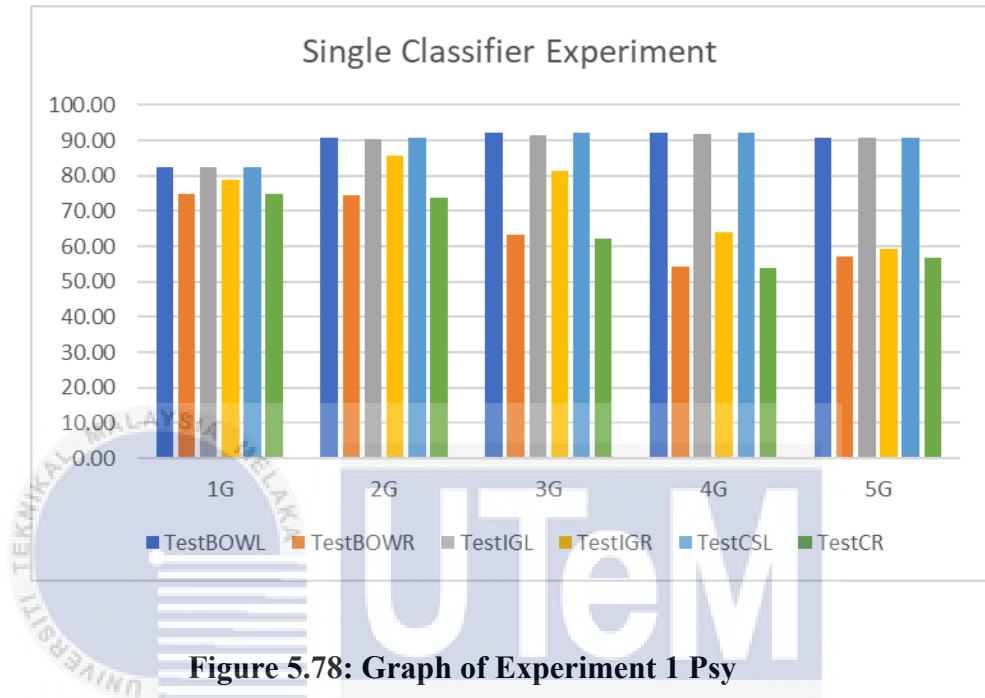


Figure 5.78: Graph of Experiment 1 Psy

5.5.2.3 Accuracy Table

Following is a summary of the final result of Psy datasets for experiments single and multiple classifiers, categorized every model and ideal kernel based on the five gram of the experiments, shown in this subchapter. Table 5.37 contains the findings of experiment 1 Psy single classifier accuracy and figure 5.79 represents the findings of experiment 2 Psy multiple classifier.

Table 5.JJ: Single Classifier of Psy Accuracy

Psy Experiment 1						
Gram	Bag Of Word (BOW)		Information Gain (IG)		Chi-Square (CS)	
	Linear(%)	RBF(%)	Linear(%)	RBF(%)	Linear(%)	RBF(%)

1	82.42	74.85	82.42	78.79	82.42	74.85
2	90.61	74.55	90.30	85.76	90.61	73.94
3	92.12	63.33	91.52	81.21	92.12	62.12
4	92.12	54.24	91.82	63.94	92.12	53.54
5	90.91	57.27	90.91	59.39	90.91	56.67

Run	Product	Mean	MV
1	93.94%	93.94%	96.97%
2	96.97%	93.94%	93.94%
3	87.88%	90.91%	87.88%
4	90.91%	90.91%	90.91%
5	93.94%	96.97%	96.97%
6	93.94%	93.94%	93.94%
7	96.97%	96.97%	96.97%
8	96.97%	96.97%	93.94%
9	90.91%	87.88%	84.85%
10	87.88%	84.85%	90.91%
Average	93.03%	92.73%	92.73%

Figure 5.79: Multiple Classifier of Psy Accuracy

The results of the Ngram single classifier using three alternative models are as shown in Table 5.37. It displays the percentages of BOW, IG, and CS using the linear kernel and RBF. The linear 3 gram and 4 gram BOW and CS linear have a precision of 92.12% compare to the other gram and kernel for each model. As a consequence of this finding, the experiment with a single classifier will use 3 gram and 4 gram of BOW and CS kernel linear as the benchmark.

Meanwhile, the results for Ngram multiple classifiers using three alternative weights like product, mean, and majority voting (MV) as shown in figure 5.79. Experiment 2 has a precision of 93.03% in product compare to the mean and MV. It compares the percentage based on the ranking accuracy for Ngram, n=1,2,3,4,5.

5.5.3 Shakira Dataset Result

The estimated number of instances of YouTube spam for the Shakira dataset is 370, with 174 instances being spam and 196 instances being legitimate.

5.5.3.1 Attribute

It takes 170 instances to achieve the BOW's accuracy for every Ngram in both experiments. Thus, a total of 170 instances are used. Process feature selection was carried out using the specified attributes and ranks, with the degree of relevance for predictions being referred to as a benchmark. When it comes to the chosen attributes, the researcher previously explained in the previous chapter that the study had chosen 80 percent of the total set of attributes for this particular project before starting. Table 5.39 indicates the proportion of 80 percent characteristics for both the algorithm's single classifier and multiple classifier versions.

Table 5.KK: 80% of Total Instances Attributes Shakira

Gram	Total Attribute	80% of Total Attribute
1	375	300
2	1333	1067
3	4633	3707
4	9656	7725
5	13932	11146

5.5.3.2 10 Runs

Figure 5.80 signifies the ten run-time train and test results produced by BOW, IG, and CS for each kernel. TrainBOWL, TrainIGL and TrainCSL are the abbreviations for train BOW, IG and CS linear kernel, meanwhile TrainBOWR, TrainIGR and TrainCSR are abbreviations for train BOW, IG and CS RBF kernel. The same goes for the others. TestBOWL, TestIGL and TestCSL are abbreviations for linear test kernel, and TestBOWR, TestIGR and TestCSR are abbreviations for test RBF kernel.

Run	Gram	TrainBOWL	TestBOWL	TrainBOWR	TestBOWR	TrainIGL	TestIGL	TrainIGR	TestIGR	TrainCSL	TestCSL	TrainCSR	TestCR
1	1	1.00	0.91	100.00	52.94	1.00	0.91	100.00	82.35	1.00	0.91	100.00	52.94
2	1	1.00	0.85	100.00	85.29	1.00	0.85	100.00	85.29	1.00	0.85	100.00	85.29
3	1	1.00	0.85	100.00	88.24	1.00	0.85	100.00	85.29	1.00	0.85	100.00	88.24
4	1	1.00	0.88	100.00	94.12	1.00	0.88	100.00	91.18	1.00	0.88	100.00	94.12
5	1	1.00	0.76	99.26	73.53	1.00	0.76	99.26	73.53	1.00	0.76	99.26	70.59
6	1	1.00	0.91	100.00	94.12	1.00	0.91	100.00	94.12	1.00	0.91	100.00	94.12
7	1	1.00	0.94	100.00	94.12	1.00	0.94	100.00	94.12	1.00	0.94	100.00	94.12
8	1	1.00	0.85	100.00	50.00	1.00	0.85	100.00	50.00	1.00	0.85	100.00	50.00
9	1	1.00	0.79	100.00	85.29	1.00	0.79	100.00	85.29	1.00	0.79	100.00	88.24
10	1	1.00	0.88	100.00	88.24	1.00	0.88	100.00	88.24	1.00	0.88	100.00	88.24
1G		100.00	86.47	99.93	80.59	1.00	86.47	99.93	82.94	1.00	86.47	99.93	80.59

Figure 5.80: Example of 10 Runs Experiment for 1 gram by Shakira

In figure 5.81, a graph bar is produced based on the data from the table. By looking at the results, we will be able to calculate the average accuracy of each kernel that is run by BOW, IG, and CS, which will do in Excel.

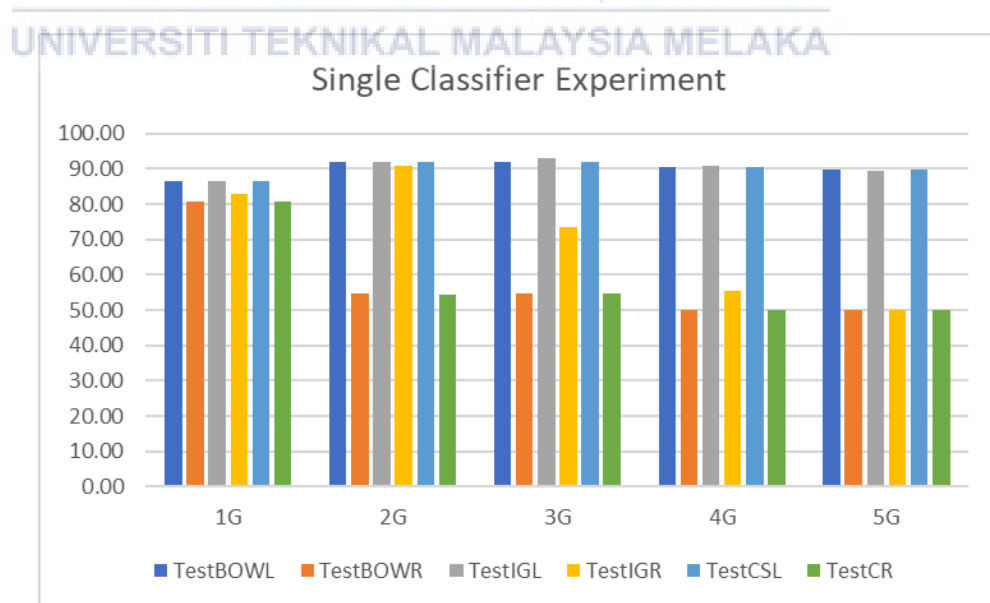


Figure 5.81: Graph of Experiment 1 Shakira

5.5.3.3 Accuracy Table

Following is a summary of the final result of Shakira datasets for experiments single and multiple classifiers, categorized every model and ideal kernel based on the five gram of the experiments, shown in this subchapter. Table 5.40 contains the findings of experiment 1 Shakira single classifier accuracy and figure 5.82 represents the findings of experiment 2 Shakira multiple classifier.

Table 5.LL: Single Classifier of Shakira Accuracy

Shakira Experiment 1						
Gram	Bag Of Word (BOW)		Information Gain (IG)		Chi-Square (CS)	
	Linear(%)	RBF(%)	Linear(%)	RBF(%)	Linear(%)	RBF(%)
1	86.47	80.59	86.47	82.94	86.47	80.59
2	92.06	54.17	92.06	90.88	92.06	54.41
3	92.06	54.17	92.94	73.53	92.06	54.17
4	90.59	50.00	90.88	55.29	90.59	50.00
5	89.71	50.00	89.41	50.00	89.71	50.00

Run	Product	Mean	MV
1	91.18%	88.24%	88.24%
2	91.18%	91.18%	94.12%
3	97.06%	97.06%	94.12%
4	97.06%	97.06%	94.12%
5	88.24%	91.18%	91.18%
6	94.12%	94.12%	94.12%
7	94.12%	94.12%	94.12%
8	79.41%	79.41%	82.35%
9	91.18%	91.18%	91.18%
10	100.00%	100.00%	100.00%
Average	92.35%	92.35%	92.35%

Figure 5.82: Multiple Classifier of Shakira Accuracy

The results of the Ngram single classifier using three alternative models are as shown in Table 5.40. It displays the percentages of BOW, IG, and CS using the linear kernel and RBF. The linear 3 gram IG linear have a precision of 92.94% compare to the other gram and kernel for each model. As a consequence of this finding, the experiment with a single classifier will use 3 gram of IG kernel linear as the benchmark.

Meanwhile, the results for Ngram multiple classifiers using three alternative weights like product, mean, and majority voting (MV) as shown in figure 5.82. Experiment 2 has same precision of 92.35% for all product, mean and MV. It compares the percentage based on the ranking accuracy for Ngram, $n=1,2,3,4,5$.

5.5.4 LMFAO Dataset Result

The estimated number of instances of YouTube spam for the LMFAO dataset is 438, with 236 instances being spam and 202 instances being legitimate.

5.5.4.1 Attribute

It takes 200 instances to achieve the BOW's accuracy for every Ngram in both experiments. Thus, a total of 200 instances are used. Process feature selection was carried out using the specified attributes and ranks, with the degree of relevance for predictions being referred to as a benchmark. When it comes to the chosen attributes, the researcher previously explained in the previous chapter that the study had chosen

80 percent of the total set of attributes for this particular project before starting. Table 5.41 indicates the proportion of 80 percent characteristics for both the algorithm's single classifier and multiple classifier versions.

Table 5.MM: 80% of Total Instances Attributes LMFAO

Gram	Total Attribute	80% of Total Attribute
1	470	376
2	1351	1081
3	3852	3082
4	7027	5622
5	9494	7596

5.5.4.2 10 Runs

Figure 5.83 signifies the ten run-time train and test results produced by BOW, IG, and CS for each kernel. TrainBOWL, TrainIGL and TrainCSL are the abbreviations for train BOW, IG and CS linear kernel, meanwhile TrainBOWR, TrainIGR and TrainCSR are abbreviations for train BOW, IG and CS RBF kernel. The same goes for the others. TestBOWL, TestIGL and TestCSL are abbreviations for linear test kernel, and TestBOWR, TestIGR and TestCSR are abbreviations for test RBF kernel.

Run	Gram	TrainBOWL	TestBOWL	TrainBOWR	TestBOWR	TrainIGL	TestIGL	TrainIGR	TestIGR	TrainCSL	TestCSL	TrainCSR	TestCR
1	1	1.00	0.78	100.00	77.50	1.00	0.78	100.00	80.00	1.00	0.78	100.00	80.00
2	1	1.00	0.88	100.00	90.00	1.00	0.88	100.00	90.00	1.00	0.88	100.00	87.50
3	1	1.00	0.90	100.00	85.00	1.00	0.88	100.00	92.50	1.00	0.90	100.00	82.50
4	1	1.00	0.90	98.13	87.50	1.00	0.88	100.00	87.50	1.00	0.90	98.13	87.50
5	1	0.96	0.90	100.00	90.00	0.96	0.93	100.00	92.50	0.96	0.90	100.00	90.00
6	1	1.00	0.90	97.50	90.00	1.00	0.90	96.88	87.50	1.00	0.90	97.50	87.50
7	1	1.00	0.80	100.00	77.50	1.00	0.80	100.00	77.50	1.00	0.80	100.00	77.50
8	1	0.93	0.83	100.00	85.00	1.00	0.85	100.00	85.00	0.93	0.83	100.00	85.00
9	1	1.00	0.85	100.00	85.00	1.00	0.85	100.00	85.00	1.00	0.85	100.00	85.00
10	1	1.00	0.90	100.00	90.00	1.00	0.90	100.00	92.50	1.00	0.90	100.00	90.00
1G		0.99	86.25	99.56	85.75	1.00	86.25	99.69	87.00	0.99	86.25	99.56	85.25

Figure 5.83: Example of 10 Runs Experiment for 1 gram by LMFAO

In figure 5.84, a graph bar is produced based on the data from the table. By looking at the results, we will be able to calculate the average accuracy of each kernel that is run by BOW, IG, and CS, which will do in Excel.

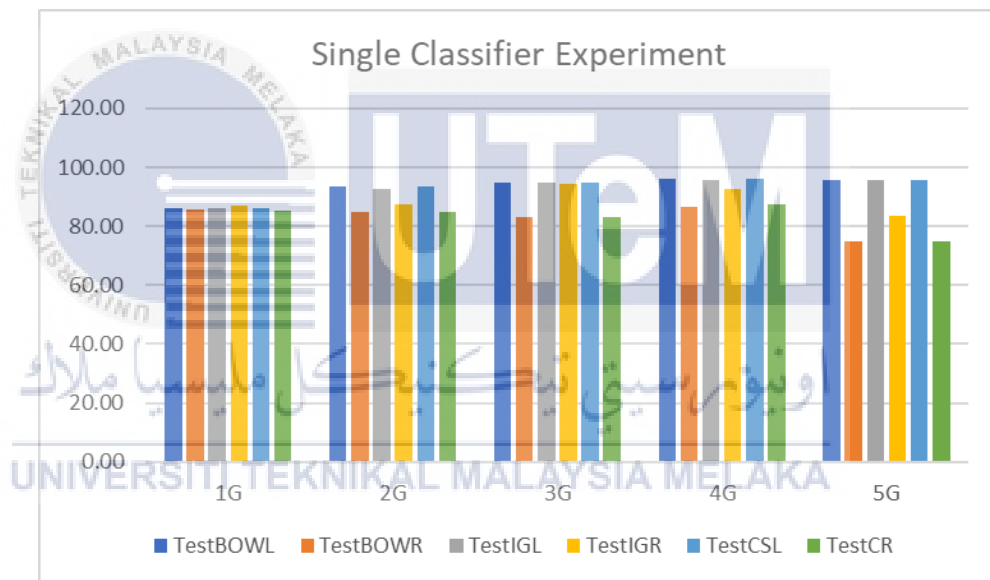


Figure 5.84: Graph of Experiment 1 LMFAO

5.5.4.3 Accuracy Table

Following is a summary of the final result of LMFAO datasets for experiments single and multiple classifiers, categorized every model and ideal kernel based on the five gram of the experiments, shown in this subchapter. Table 5.42 contains the findings of experiment 1 LMFAO single classifier accuracy and figure 5.85 represents the findings of experiment 2 LMFAO multiple classifier.

Table 5.NN: Single Classifier of LMFAO Accuracy

LMFAO Experiment 1						
Gram	Bag Of Word (BOW)		Information Gain (IG)		Chi-Square (CS)	
	Linear(%)	RBF(%)	Linear(%)	RBF(%)	Linear(%)	RBF(%)
1	86.25	85.75	86.25	87.00	86.25	85.25
2	93.50	84.75	92.75	87.25	93.50	84.75
3	95.00	83.25	95.00	94.25	95.00	83.00
4	96.00	86.50	95.75	92.75	96.00	87.25
5	95.75	75.00	95.75	83.50	95.75	74.75

Run	Product	Mean	MV
1	92.50%	92.50%	92.50%
2	100.00%	100.00%	97.50%
3	95.00%	95.00%	95.00%
4	92.50%	92.50%	92.50%
5	95.00%	95.00%	90.00%
6	97.50%	100.00%	97.50%
7	95.00%	95.00%	97.50%
8	100.00%	100.00%	97.50%
9	92.50%	92.50%	92.50%
10	100.00%	100.00%	97.50%
Average	96.00%	96.25%	95.00%

Figure 5.85: Multiple Classifier of LMFAO Accuracy

The results of the Ngram single classifier using three alternative models are as shown in Table 5.42. It displays the percentages of BOW, IG, and CS using the linear kernel and RBF. The linear 4 gram BOW and CS linear have a precision of 96.00% compare to the other gram and kernel for each model. As a consequence of this finding, the experiment with a single classifier will use 4 gram of BOW and CS kernel linear as the benchmark.

Meanwhile, the results for Ngram multiple classifiers using three alternative weights like product, mean, and majority voting (MV) as shown in figure 5.85. Experiment 2 has same precision of 96.25% in mean compare to product and MV. It compares the percentage based on the ranking accuracy for Ngram, $n=1,2,3,4,5$.

5.5.5 Katy Perry Dataset Result

The estimated number of instances of YouTube spam for the Katy Perry dataset is 350, with 175 instances being spam and also, 175 instances being legitimate.

5.5.5.1 Attribute

It takes 175 instances to achieve the BOW's accuracy for every Ngram in both experiments. Thus, a total of 175 instances are used. Process feature selection was carried out using the specified attributes and ranks, with the degree of relevance for predictions being referred to as a benchmark. When it comes to the chosen attributes, the researcher previously explained in the previous chapter that the study had chosen 80 percent of the total set of attributes for this particular project before starting. Table 5.43 indicates the proportion of 80 percent characteristics for both the algorithm's single classifier and multiple classifier versions.

Table 5.00: 80% of Total Instances Attributes Katy Perry

Gram	Total Attribute	80% of Total Attribute
1	402	322
2	2078	1663
3	7451	5961
4	13802	11042
5	19122	15298

5.5.5.2 10 Runs

Figure 5.86 signifies the ten run-time train and test results produced by BOW, IG, and CS for each kernel. TrainBOWL, TrainIGL and TrainCSL are the abbreviations for train BOW, IG and CS linear kernel, meanwhile TrainBOWR, TrainIGR and TrainCSR are abbreviations for train BOW, IG and CS RBF kernel. The same goes for the others. TestBOWL, TestIGL and TestCSL are abbreviations for linear test kernel, and TestBOWR, TestIGR and TestCSR are abbreviations for test RBF kernel.

Run	Gram	TrainBOWL	TestBOWL	TrainBOWR	TestBOWR	TrainIGL	TestIGL	TrainIGR	TestIGR	TrainCSL	TestCSL	TrainCSR	TestCR
1	1	1.00	0.89	100.00	88.57	1.00	0.89	100.00	88.57	1.00	0.89	100.00	88.57
2	1	1.00	0.86	98.57	82.86	1.00	0.86	98.57	82.86	1.00	0.86	99.29	82.86
3	1	1.00	0.91	100.00	91.43	1.00	0.91	100.00	91.43	1.00	0.91	100.00	91.43
4	1	1.00	0.86	100.00	85.71	1.00	0.86	100.00	85.71	1.00	0.86	100.00	85.71
5	1	1.00	0.83	100.00	82.86	1.00	0.83	100.00	82.86	1.00	0.83	100.00	82.86
6	1	0.93	0.89	100.00	85.71	0.93	0.89	100.00	85.71	0.93	0.89	100.00	88.57
7	1	1.00	0.86	100.00	88.57	1.00	0.86	100.00	82.86	1.00	0.86	100.00	88.57
8	1	1.00	0.71	100.00	74.29	1.00	0.71	100.00	68.57	1.00	0.71	100.00	74.29
9	1	1.00	0.77	100.00	77.14	1.00	0.77	100.00	77.14	1.00	0.77	100.00	77.14
10	1	1.00	0.86	100.00	91.43	1.00	0.86	100.00	88.57	1.00	0.86	100.00	91.43
1G		0.99	84.29	99.86	84.86	0.99	84.29	99.86	83.43	0.99	84.29	99.93	85.14

Figure 5.86: Example of 10 Runs Experiment for 1 gram by Katy Perry

In figure 5.87, a graph bar is produced based on the data from the table. By looking at the results, we will be able to calculate the average accuracy of each kernel that is run by BOW, IG, and CS, which will do in Excel.

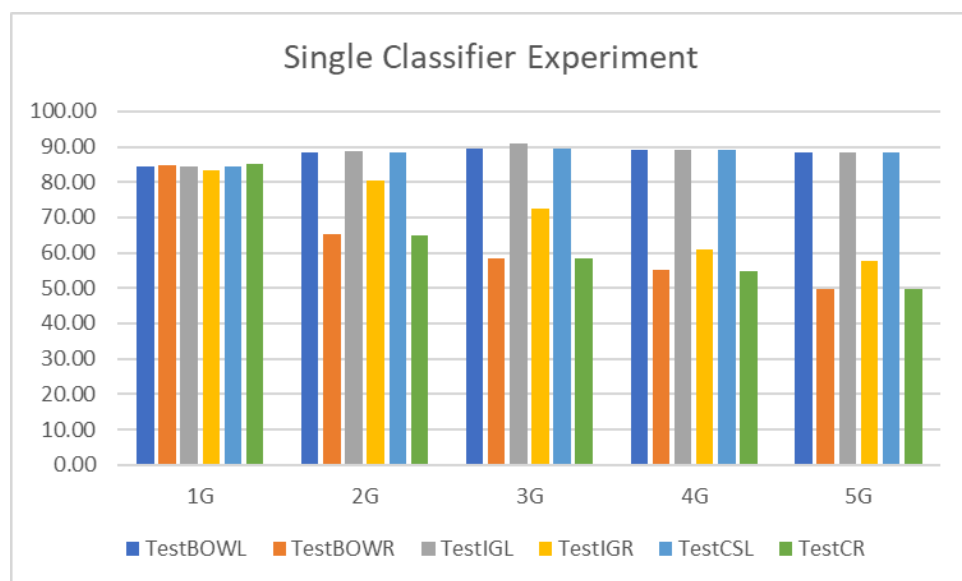


Figure 5.87: Graph of Experiment 1 Katy Perry**5.5.5.3 Accuracy Table**

Following is a summary of the final result of Katy Perry datasets for experiments single and multiple classifiers, categorized every model and ideal kernel based on the five gram of the experiments, shown in this subchapter. Table 5.44 contains the findings of experiment 1 Katy Perry single classifier accuracy and figure 5.88 represents the findings of experiment 2 Katy Perry multiple classifier.

Table 5.PP: Single Classifier of Katy Perry Accuracy

LMFAO Experiment 1						
Gram	Bag Of Word (BOW)		Information Gain (IG)		Chi-Square (CS)	
	Linear(%)	RBF(%)	Linear(%)	RBF(%)	Linear(%)	RBF(%)
1	84.29	84.86	84.29	83.43	84.29	85.14
2	88.57	65.14	88.86	80.57	88.57	64.86
3	89.43	58.29	90.86	72.57	89.43	68.57
4	89.14	55.14	89.14	60.86	89.14	54.86
5	88.57	49.71	88.57	57.71	88.57	49.71

Run	Product	Mean	MV
1	94.29%	91.43%	94.29%
2	91.43%	91.43%	91.43%
3	88.57%	85.71%	85.71%
4	91.43%	97.14%	97.14%
5	85.71%	88.57%	85.71%
6	85.71%	85.71%	88.57%
7	94.29%	97.14%	94.29%
8	85.71%	82.86%	82.86%
9	91.43%	97.14%	97.14%
10	88.57%	91.43%	91.43%
Average	89.71%	90.86%	90.86%

Figure 5.88: Multiple Classifier of Katy Perry Accuracy

The results of the Ngram single classifier using three alternative models are as shown in Table 5.44. It displays the percentages of BOW, IG, and CS using the linear kernel and RBF. The linear 3 gram IG linear have a precision of 90.86% compare to the other gram and kernel for each model. As a consequence of this finding, the experiment with a single classifier will use 3 gram of IG kernel linear as the benchmark.

Meanwhile, the results for Ngram multiple classifiers using three alternative weights like product, mean, and majority voting (MV) as shown in figure 5.88. Experiment 2 has same precision of 90.86% in mean and mv compare to product. It compares the percentage based on the ranking accuracy for Ngram, n=1,2,3,4,5.

5.5.6 Summary Dataset Result

The summarization for the best technique for each dataset is pointed in Table 5.43. Not all classification methods work well for all datasets. In addition, specific classification methods are not appropriate for the attributes. However, a methodology for spam detection is needed to enhance future spam detection methods.

Table 5.QQ: Dataset Summarization

Dataset	Experiment	Gram	Model/ Weight	Accuracy(%)

Eminem	Single and Multiple Classifier	3	IG Linear and MV	100
Psy	Multiple Classifier	3 & 4	Product	93.03
Shakira	Multiple Classifier	3	Product, Mean and MV	92.35
LMFAO	Multiple Classifier	4	Mean	96.25
Katy Perry	Single and Multiple Classifier	3	IG Linear, Mean and MV	90.86

5.6 Conclusion

By referring to the result analysis, we can observe that a more significant percentage increases the multiple classifiers experiment's accuracies. The kernel most often employed kernel is a linear kernel. The next chapter examines the findings produced for this study in more profound, and we will also evaluate the research's goal. In conclusion, this chapter should contain a structure for planning development and execution to maximize the effectiveness of this study.

CHAPTER 6: DISCUSSION

6.1 Introduction

We reviewed and clearly defined all of the elements that this research must implement to create a method for the project in the previous chapter. The point that has been brought up is the preparation of the environment and the creation of software, the module of a process that includes data collecting, preprocessing data, training and testing data, and the implementation of the experiment's execution. The results of tests are documented and seen in graph bar form while an experiment is taking place.

Hence, this chapter will discuss how the researcher carried out the project from the start to completion and the outcomes. Furthermore, it will explain and suggest a technique to create the result and evaluate the outcome to determine if it will beat or not in this section. Also, we will analyze the results of the experiment and compare the suggested technique with the reference method.

6.2 Discussion of Project

One of the most serious issues is the growing online platform, website cybersecurity, which exposes users to digital hazards. This study employs four methods for detecting YouTube spam. After data preprocessing, the procedure produces Ngrams, BOWs, and Chi-Square, and Information Gain features. UCI Machine Learning Repository data is taken.

A technique may remove special characters such as non-ASCII characters, convert content to lower case, replace no author and no date to 0, implement stemming

single classifier and multiple classifiers, ensemble method. Script running generates Ngram feature extraction, feature vector, and train test data. Then, preprocess data for BOW, a vector space model. The dataset specifies the feature's frequency. The IG and CS algorithms will finalize the processes. The prediction file is then used to identify legitimate and spam comments using the SVM classifier. Finally, each process accuracy will be evaluated using a prediction file, and we will calculate the results.

6.3 Discussion on The Newly Proposed Method

As indicated in table 6.1, the preceding chapter provides a base for comparison.

Table 6.A: The Benchmark

Dataset	Experiment	Gram	Model	Accuracy(%)
Eminem	With stemming	2	IG RBF Kernel	93.4
Psy	Without stemming	3	IG Linear Kernel	81.8
Shakira	With stemming	2	IG RBF Kernel	91.9
LMFAO	With stemming	3	BOW, IG and CS Linear Kernel	94.8
Katy Perry	Without stemming	5	BOW and CS Linear Kernel	76.0

As demonstrated in Table 6.2, the newly suggested method beat the benchmark. In the Eminem dataset, there is a 6.6% difference. There is a 0.45% difference in Shakira's dataset, and there is a 1.45% difference in the LMFAO dataset. Until the Psy data set, there is a moderate inequality between 11.23% and 14.85%, respectively. The preprocessed data's author, date, and content are the most

critical factors that significantly impact the results. Thus, the removal of non-ASCII characters had a considerable effect on noise reduction.

Table 6.B: Final Result

Dataset	Experiment	Gram	Model/ Weight	Accuracy(%)
Eminem	Single Classifier	3	IG Linear	100
Psy	Multiple Classifier	3 & 4	Product	93.03
Shakira	Multiple Classifier	3	Product, Mean and MV	92.35
LMFAO	Multiple Classifier	4	Mean	96.25
Katy Perry	Single Classifier	3	IG Linear	90.86

6.4 Conclusion

Finally, this chapter elaborates on the project's conception, implementation, and execution. The execution approach turns the design and analysis process into a youtube spam detection. This chapter is indeed essential for defining the project's parameters and methods.

CHAPTER 7: PROJECT CONCLUSION

7.1 Introduction

This chapter focuses on summarising all of the project's outcomes and results. This last phase is necessary for ensuring that the goals are met and discussed effectively. This chapter will describe how each of the goals is completed and what may be done to enhance the process in the future. Constraints are also addressed and evaluated since they are the critical factor determining the outcome of this project.

7.2 Project Summary

A spam comment detection model is needed since it may benefit in preventing attacks and differentiating between spam and legitimate comments. Unfortunately, the spam comments have resulted in individuals becoming spam victims. To address the problems, it is necessary to research internet security attacks in spam attacks to comprehend how they work in social media, like YouTube. Also, it is essential to understand classification and machine learning techniques to differentiate spam from legitimate comments through machine learning implementation.

To perform the experiment, the following steps will be taken, determining the accurate model of the machine-learning algorithm, evaluating the behaviour N-gram and feature selection of Information Gain (IG) and Chi-Square (CS), and validating the model between various datasets as Eminem, Katy Perry, Psy, and Shakira. This procedure decides whether or not the finished result can exceed the benchmark in terms of results.

7.3 Project Constraint

This research does encounter some constraints while performing several experiments, including the fact that they are time-consuming and draining since they need workstations with advanced memory and CPU. Apart from that, this technique is restricted to Linear kernels and RBF.

7.4 Project Contribution

The project's contribution would be to identify the optimal pattern of accuracy performance on YouTube datasets based on the Bag of Words' behaviour and feature selection. Besides, the project is verifying itself effectively by using various YouTube datasets.

7.5 Project Limitation

This research project is entirely devoted to content processing. This project makes use of YouTube spam data collected by many researchers from the UCI Machine Learning Repository. Therefore, this research comprises to use machine learning techniques solely.

7.6 Future Work

In the future, the researcher may enhance this project by processing all features, not only content features. For example, a spam detection technique should be introduced and evaluated against a variety of classifiers to detect spam comments rather than focusing only on one classifier. In addition, the researcher may improve accuracy by using improved stop word lists, parameters and data preparation. However, since the training process is dependent on specified devices, the project may include modern technology.

7.7 Conclusion

To summarise, the study accomplished all three objectives. Yet, the researcher may make some future enhancements to maximise their effectiveness. Any proposed modification should always guarantee a high degree of accuracy in detecting spam comments. In addition, it will help to ensure that users are not vulnerable to spam attacks if the spam comments have been seen before.

REFERENCES

- Abdulhamid, S. M., Abd Latiff, M. S., Chiroma, H., Osho, O., Abdul-Salaam, G., Abubakar, A. I., & Herawan, T. (2017). A Review on Mobile SMS Spam Filtering Techniques. *IEEE Access*, 5, 15650–15666. <https://doi.org/10.1109/ACCESS.2017.2666785>
- Agarwal, V. (2015). Research on Data Preprocessing and Categorization Technique for Smartphone Review Analysis. *International Journal of Computer Applications*, 131(4), 30–36. <https://doi.org/10.5120/ijca2015907309>
- Ahmad, K., Vivekananda, S., & Pradesh, U. (2015). Classification of Internet Security Attacks Classification of Internet Security Attacks. *Computing For Nation Development, October*, 1–4.
- Ahmad, N., & Habib, M. K. (2010). Analysis of Network Security Threats and Vulnerabilities by Development & Implementation of a Security Network Monitoring Solution. *School of Engineering Department of Telecommunication Blekinge Institute of Technology SE - 371 79 Karlskrona Sweden*.
- Aiyar, S., & Shetty, N. P. (2018a). N-Gram Assisted Youtube Spam Comment Detection. *Procedia Computer Science*, 132(Iccids), 174–182. <https://doi.org/10.1016/j.procs.2018.05.181>
- Aiyar, S., & Shetty, N. P. (2018b). N-Gram Assisted Youtube Spam Comment Detection. *Procedia Computer Science*, 132(Iccids), 174–182. <https://doi.org/10.1016/j.procs.2018.05.181>
- Alberto, T. C., Lochter, J. V., & Almeida, T. A. (2016). TubeSpam: Comment spam filtering on YouTube. *Proceedings - 2015 IEEE 14th International Conference on Machine Learning and Applications, ICMLA 2015, 2013*, 138–143. <https://doi.org/10.1109/ICMLA.2015.37>
- Ali, A., & Amin, M. Z. (2016). *An Approach for Spam Detection in YouTube Comments Based on Supervised Learning. December 2019*, 1–10.
- Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). *A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques*.
- Amir Sjarif, N. N., Mohd Azmi, N. F., Chuprat, S., Sarkan, H. M., Yahya, Y., & Sam, S. M. (2019). SMS spam message detection using term frequency-inverse

- document frequency and random forest algorithm. *Procedia Computer Science*, 161, 509–515. <https://doi.org/10.1016/j.procs.2019.11.150>
- Annadatha, A., & Stamp, M. (2018). Image spam analysis and detection. *Journal of Computer Virology and Hacking Techniques*, 14(1), 39–52. <https://doi.org/10.1007/s11416-016-0287-x>
- Attar, A., Rad, R. M., & Atani, R. E. (2013). A survey of image spamming and filtering techniques. *Artificial Intelligence Review*, 40(1), 71–105. <https://doi.org/10.1007/s10462-011-9280-4>
- Aziz, A., Foozy, C. F. M., Shamala, P., & Suradi, Z. (2018). Youtube spam comment detection using support vector machine and K-nearest neighbor. *Indonesian Journal of Electrical Engineering and Computer Science*, 12(2), 607–611. <https://doi.org/10.11591/ijeecs.v12.i2.pp607-611>
- Banu, M. N., & Banu, S. M. (2013). A Comprehensive Study of Phishing Attacks. *International Journal of Computer Science and Information Technologies (IJCSIT)*, 4(6), 783–786.
- Berk, R. A. (2020). *Support Vector Machines*. 339–359. https://doi.org/10.1007/978-3-030-40189-4_7
- Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, 300, 70–79. <https://doi.org/10.1016/j.neucom.2017.11.077>
- Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers and Electrical Engineering*, 40(1), 16–28. <https://doi.org/10.1016/j.compeleceng.2013.11.024>
- Chaudhari, M., & Govilkar, S. (2015). A Survey of Machine Learning Techniques for Sentiment Classification. *International Journal on Computational Science & Applications*, 5(3), 13–23. <https://doi.org/10.5121/ijcsa.2015.5302>
- Chen, S. H., & Pollino, C. A. (2012). Good practice in Bayesian network modelling. *Environmental Modelling and Software*, 37, 134–145. <https://doi.org/10.1016/j.envsoft.2012.03.012>
- Chowdury, R., Monsur Adnan, M. N., Mahmud, G. A. N., & Rahman, R. M. (2013). A data mining based spam detection system for YouTube. *8th International Conference on Digital Information Management, ICDIM 2013*, 373–378. <https://doi.org/10.1109/ICDIM.2013.6694038>
- Crawford, M., Khoshgoftaar, T. M., Prusa, J. D., Richter, A. N., & Al Najada, H.

- (2015). Survey of review spam detection using machine learning techniques. *Journal of Big Data*, 2(1). <https://doi.org/10.1186/s40537-015-0029-9>
- Damodaram, R. (2016). Study on Phishing Attacks and Antiphishing Tools. *International Research Journal of Engineering and Technology*, 3(1), 700–705.
- Das, M., & Prasad, V. (2014). Analysis of an Image Spam in Email Based on Content Analysis. *International Journal on Natural Language Computing*, 3(3), 129–140. <https://doi.org/10.5121/ijnlc.2014.3313>
- Ellis-monaghan, J. (2006). *A gentle introduction to the Potts Model*. 1–21.
- Floderus, S., & Rosenholm, L. (2019). *An educational experiment in discovering spear phishing attacks*. February.
- Fund, M., Resource, H., & Estate, R. (2021). *Definition of 'Cryptography'*. 23–26. <https://economictimes.indiatimes.com/definition/cryptography>
- Gharatkar, S., Ingle, A., Naik, T., & Save, A. (2018). Review preprocessing using data cleaning and stemming technique. *Proceedings of 2017 International Conference on Innovations in Information, Embedded and Communication Systems, ICIECS 2017, 2018-Janua*, 1–4. <https://doi.org/10.1109/ICIECS.2017.8276011>
- Gupta, A., & Sharma, L. (2020). *Mitigation of DoS and Port Scan Attacks Using Snort*. *International Journal of Computer Sciences and Engineering Open Access Mitigation of DoS and Port Scan Attacks Using Snort*. January. <https://doi.org/10.26438/ijcse/v7i4.248258>
- Hayati, P., Potdar, V., Talevski, A., Firoozeh, N., Sarenche, S., & Yeganeh, E. A. (2010). Definition of Spam 2.0: New spamming boom. *4th IEEE International Conference on Digital Ecosystems and Technologies - Conference Proceedings of IEEE-DEST 2010, DEST 2010, May*, 580–584. <https://doi.org/10.1109/DEST.2010.5610590>
- Hulu, S., & Sihombing, P. (2020). Analysis of Performance Cross Validation Method and K-Nearest Neighbor in Classification Data. *International Journal of Researcher and Review*, 7(4), 69–73.
- Juneja, P. G., & Pateriya, R. K. (2014). A Survey on Email Spam Types and Spam Filtering Techniques. *International Journal of Engineering Research*, 3(3), 2309–2314.
- Kamoru, B. A., Jaafar, A. Bin, Murad, M. A. A., Ernest, E. O., & Jabar, M. B. A. (2017). Spam Detection Approaches and Strategies: A Phenomenons. *International Journal of Applied Information Systems*, 12(9), 13–18.

<https://doi.org/10.5120/ijais2017451728>

- Kang, M., Ahn, J., & Lee, K. (2017). Opinion mining using ensemble text hidden Markov models for text classification. *Expert Systems with Applications*, 94, 218–227. <https://doi.org/10.1016/j.eswa.2017.07.019>
- Karim, A., Azam, S., Shanmugam, B., Kannoorpatti, K., & Alazab, M. (2019). A comprehensive survey for intelligent spam email detection. *IEEE Access*, 7, 168261–168295. <https://doi.org/10.1109/ACCESS.2019.2954791>
- Kavitha, K. M., Shetty, A., Abreo, B., D’Souza, A., & Kondana, A. (2020). Analysis and classification of user comments on YouTube videos. *Procedia Computer Science*, 177(2018), 593–598. <https://doi.org/10.1016/j.procs.2020.10.084>
- Khalid, S., Khalil, T., & Nasreen, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. *Proceedings of 2014 Science and Information Conference, SAI 2014*, 372–378. <https://doi.org/10.1109/SAI.2014.6918213>
- Krishnamurthy, V. (2015). *Internet spam threats and email exploitation – A scuffle with inbox attack. January 2014*. <https://doi.org/10.6088/ijaser.030400015>
- Kumar, M., Babaeizadeh, M., Erhan, D., Finn, C., Levine, S., & Dinh, L. (2015). *VideoFlow: A Flow-Based Generative Model for Video*.
- Matteo, R., & Giorgio, V. (2001). Ensemble methods: a review. In *Data Mining and Machine Learning for Astronomical Applications* (Issue January 2012).
- McCord, M., & Chuah, M. (2011). Spam detection on twitter using traditional classifiers. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6906 LNCS, 175–186. https://doi.org/10.1007/978-3-642-23496-5_13
- Mehmood, A., On, B. W., Lee, I., Ashraf, I., & Sang Choi, G. (2018). Spam comments prediction using stacking with ensemble learning. *Journal of Physics: Conference Series*, 933(1). <https://doi.org/10.1088/1742-6596/933/1/012012>
- Method, H. O., & Method, R. S. (2021). *Hold Out Method & Random Sub-Sampling Method Hold-Out Method Random Sub-sampling Method*. 1–5.
- Miao, J., & Niu, L. (2016). A Survey on Feature Selection. *Procedia Computer Science*, 91(Itqm), 919–926. <https://doi.org/10.1016/j.procs.2016.07.111>
- Mokoena, T., & Zuva, T. (2018). Malware analysis and detection in enterprise systems. *Proceedings - 15th IEEE International Symposium on Parallel and Distributed Processing with Applications and 16th IEEE International*

- Conference on Ubiquitous Computing and Communications, ISPA/IUCC 2017*, 1304–1310. <https://doi.org/10.1109/ISPA/IUCC.2017.00199>
- Moritz, M., & Steding, D. (2019). Lexical and semantic features for cross-lingual text reuse classification: An experiment in English and Latin paraphrases. *LREC 2018 - 11th International Conference on Language Resources and Evaluation, 2008*, 1976–1980.
- Najadat, H., & Hmeidi, I. (2008). Web Spam Detection Using Machine Learning in Specific Domain Features. *Work*, 3(September), 220–228.
- Nath, A., & Dasgupta, A. (2016). Classification of Machine Learning Algorithms. *International Journal of Innovatice Research in Advanced Engineering*, 3(March), 6–11.
- Nath, R. K., Thapliyal, H., Caban-Holt, A., & Mohanty, S. P. (2020). Machine Learning Based Solutions for Real-Time Stress Monitoring. *IEEE Consumer Electronics Magazine*, 9(5), 34–41. <https://doi.org/10.1109/MCE.2020.2993427>
- Noble, W. S. (2006). What is a support vector machine? *Nature Biotechnology*, 24(12), 1565–1567. <https://doi.org/10.1038/nbt1206-1565>
- Olatunji, S. O. (2019). Improved email spam detection model based on support vector machines. *Neural Computing and Applications*, 31(3), 691–699. <https://doi.org/10.1007/s00521-017-3100-y>
- Oskuie, M. D., & Razavi, S. N. (2014). A Survey of Web Spam Detection Techniques. *International Journal of Computer Applications Technology and Research*, 3(3), 180–185. <https://doi.org/10.7753/ijcatr0303.1010>
- Paluszek, M., & Thomas, S. (2019). An Overview of Machine Learning. *MATLAB Machine Learning Recipes, January*, 1–18. https://doi.org/10.1007/978-1-4842-3916-2_1
- Pandey, H. (2020). *Credit Card Fraud Detection With Bagging Ensemble Learning*. 1–9.
- Perumal, S., & Velmurugan, T. (2018). Preprocessing by Contrast Enhancement Techniques for Medical Images. *International Journal of Pure and Applied Mathematics*, 118(18), 3681–3688.
- Pietrzykowski, M., & Sałabun, W. (2014). Applications of Hidden Markov Model: state-of-the-art. *International Journal Computer Technology & Applications*, 5(4), 1384–1391.
- Reddy, K. S., & Reddy, E. S. (2019). Integrated approach to detect spam in social

- media networks using hybrid features. *International Journal of Electrical and Computer Engineering (IJECE)*, 9(1), 562. <https://doi.org/10.11591/ijece.v9i1.pp562-569>
- Renear, A. H., Sacchi, S., & Wickett, K. M. (2010). Definitions of dataset in the scientific and technical literature. *Proceedings of the ASIST Annual Meeting*, 47, 3–6. <https://doi.org/10.1002/meet.14504701240>
- S.Mangrulkar, N., R. Bhagat Patil, A., & S. Pande, A. (2014). Network Attacks and Their Detection Mechanisms: A Review. *International Journal of Computer Applications*, 90(9), 37–39. <https://doi.org/10.5120/15606-3154>
- Samsudin, N. M., Mohd Foozy, C. F. B., Alias, N., Shamala, P., Othman, N. F., & Wan Din, W. I. S. (2019). Youtube spam detection framework using naïve bayes and logistic regression. *Indonesian Journal of Electrical Engineering and Computer Science*, 14(3), 1508–1517. <https://doi.org/10.11591/ijeecs.v14.i3.pp1508-1517>
- Sharma, M. (2018). *A Survey of Email Spam Filtering Methods*. 7, 14–21.
- Singh, Y., Bhatia, P. K., & Sangwan, O. (1999). A review of studies on machine learning techniques. *International Journal of Computer Science and Security*, 1(1), 70–84.
- Tahir, R. (2018). A Study on Malware and Malware Detection Techniques. *International Journal of Education and Management Engineering*, 8(2), 20–30. <https://doi.org/10.5815/ijeme.2018.02.03>
- Tran, H. (2019). *A SURVEY OF MACHINE LEARNING AND DATA MINING TECHNIQUES USED IN MULTIMEDIA SYSTEM*. 113, 13–21. <https://doi.org/10.13140/RG.2.2.20395.49446/1>
- Uysal, A. K. (2018). Feature Selection for Comment Spam Filtering on YouTube. *DATA SCIENCE AND APPLICATIONS*, 1(1).
- Vaidya, P. P. T. S. (2017). The Impact of Computer Virus Attacks and its Detection and Preventive Mechanism among Personal Computer PC Users. *International Journal of Science and Research (IJSR)*, 6(6), 2245–2247. <http://ir.lib.seu.ac.lk/handle/123456789/382%0Ahttps://www.ijsr.net/archive/v6i6/ART20174636.pdf>
- Vapnik, V. (1995). Support Vector Machine Section I : Theory. *Image Processing*.
- Yahaya, A., Mohd, M., & Abdullah, I. (2017). A Review and Proof of Concept for Phishing Scam Detection and Response using Apoptosis. *International Journal*

- of Advanced Computer Science and Applications*, 8(6), 284–289.
<https://doi.org/10.14569/ijacsa.2017.080637>
- Yang, P., Hwa Yang, Y., B. Zhou, B., & Y. Zomaya, A. (2010). A Review of Ensemble Methods in Bioinformatics. *Current Bioinformatics*, 5(4), 296–308.
<https://doi.org/10.2174/157489310794072508>
- Yu, K. A. I., Ji, L., & Zhang, X. (2002). *Kernel Nearest-Neighbor Algorithm*. 147–156.
- Zhang, Y., Zhang, H., Cai, J., & Yang, B. (2014). A weighted voting classifier based on differential evolution. *Abstract and Applied Analysis*, 2014.
<https://doi.org/10.1155/2014/376950>
- Zhou, Z.-H. (2009). Ensemble Learning. *Encyclopedia of Biometrics*, 270–273.
https://doi.org/10.1007/978-0-387-73003-5_293



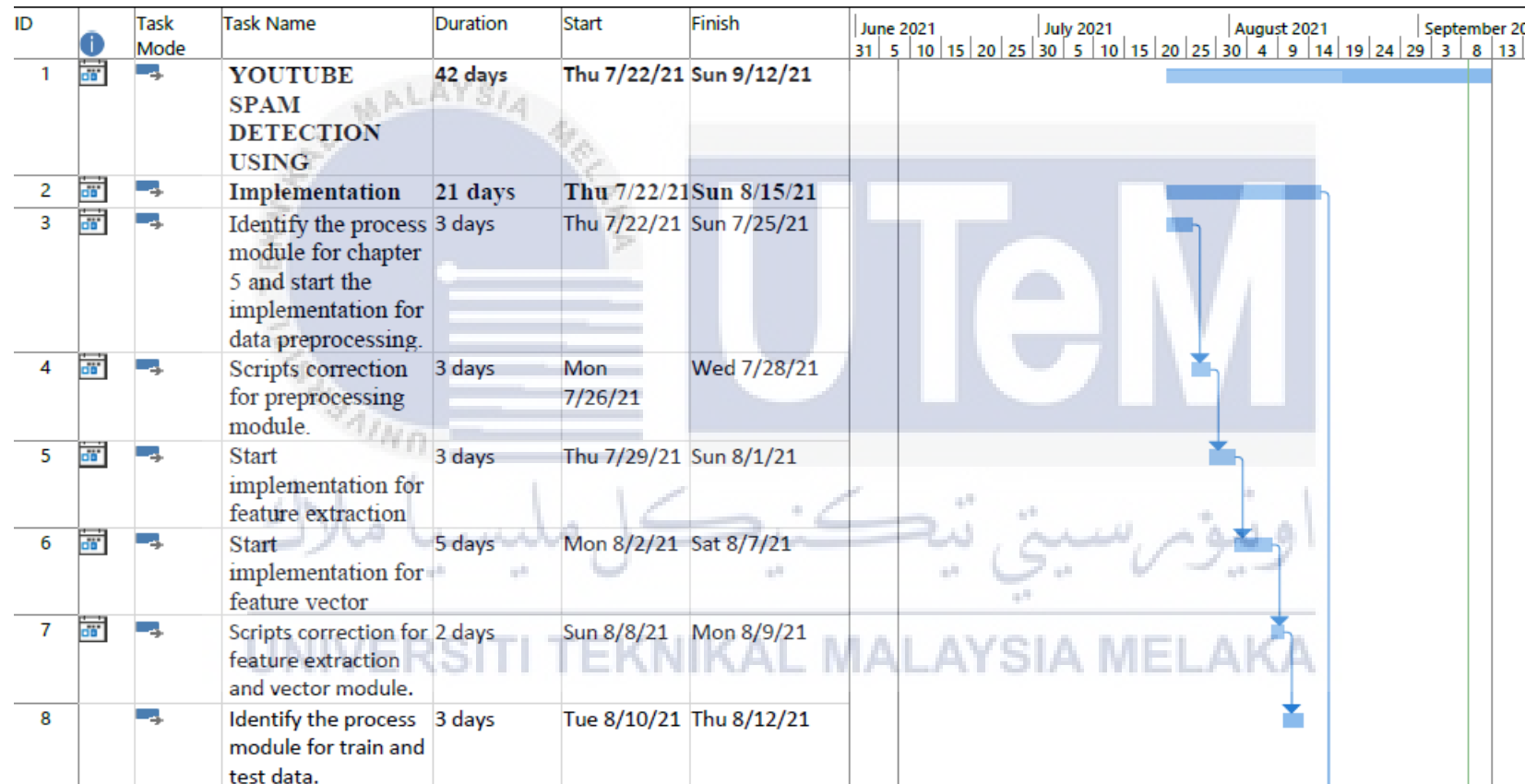
APPENDIX

Appendix A- Gantt Chart PSM 1

ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Qtr 2, 2021				Qtr 3, 2021	
							Mar	Apr	May	Jun	Jul	Aug
1		YOUTUBE SPAM DETECTION USING	80 days	Mon 15/3/21	Sun 4/7/21							
2		Planning	20 days	Mon 15/3/21	Sun 11/4/21							
3		Proposal discussion with Supervisor. Identify title, problem statement	5 days	Mon 15/3/21	Sun 21/3/21							
4		Study and research the literature review. Write and submit project proposal to Supervisor and	5 days	Mon 22/3/21	Sun 28/3/21	3						
5		Proposal accepted. Identify title, problem statement, objective and scope	5 days	Mon 29/3/21	Fri 2/4/21	4						
6		Chapter 1 is done and submit to supervisor for evaluation.	5 days	Mon 5/4/21	Fri 9/4/21	5						
7		Analysis	25 days	Tue 8/6/21	Mon 12/7/21							

ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Qtr 2, 2021				Qtr 3, 2021	
							Mar	Apr	May	Jun	Jul	Aug
8		Studies on related work and previous research and finding of spam classification.	10 days	Mon 12/4/21	Sun 25/4/21	2						
9		Study methodology on previous research.	10 days	Mon 26/4/21	Sun 9/5/21	8						
10		MID SEMESTER BREAK	5 days	Mon 10/5/21	Sun 16/5/21	9						
11		Design	35 days	Mon 17/5/21	Sun 4/7/21							
12		Information collection and analysis. Design the project and choose the tools for implement.	5 days	Tue 13/7/21	Mon 19/7/21	7						
13		Design the environment for implementation.	10 days	Tue 20/7/21	Mon 2/8/21	12						
14		Project demonstration and report.	10 days	Tue 3/8/21	Mon 16/8/21	13						
15		Final project demonstration.	5 days	Tue 17/8/21	Mon 23/8/21	14						
16		Correction and submission PSM1 Report	5 days	Tue 24/8/21	Mon 30/8/21	15						

Appendix B- Gantt Chart PSM 2



ID	Task Mode	Task Name	Duration	Start	Finish	June 2021							July 2021							August 2021				September 2021			
						31	5	10	15	20	25	30	5	10	15	20	25	30	4	9	14	19	24	29	3	8	13
9	→	Start implementation for train and test	2 days	Fri 8/13/21	Sat 8/14/21																						
10	→	Scripts correction for train and test data module and finalize results from experiments and report writing.	1 day	Sat 8/14/21	Sun 8/15/21																						
11	→	Discussion	6 days	Mon 8/16/21	Sun 8/22/21																						
12	→	Chapter 6 is done and submit to supervisor for	6 days	Mon 8/16/21	Sun 8/22/21																						
13	→	Conclusion	15 days	Mon 8/23/21	Fri 9/10/21																						
14	→	Chapter 7 is done and submit to supervisor for	2 days	Mon 8/23/21	Tue 8/24/21																						
15	→	Write and finalize project report	11 days	Wed 8/25/21	Wed 9/8/21																						
16	→	Submit project report to evaluator and supervisor.	1 day	Thu 9/9/21	Thu 9/9/21																						
17	→	Final project presentation	1 day	Fri 9/10/21	Fri 9/10/21																						