

CHI-SQUARE STATISTICAL AND SUPPORT VECTOR MACHINE FOR ANDROID
MALWARE DETECTION



FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2017

BORANG PENGESAHAN STATUS TESIS

JUDUL: Chi-Square Feature Selection and Support Vector Machine for Android Malware Detection

SESI PENGAJIAN: 2016 / 2017

Saya MUHAMMAD ZULHILMI BIN JOHARI

mengaku membenarkan tesis (PSM/Sarjana/Doktor Falsafah) ini disimpan di Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat Salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat Salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. ** Sila tandakan (/)

_____ SULIT
_____ TERHAD
_____ TIDAK TERHAD

(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

(TANDATANGAN PENULIS)

Alamat tetap:10, Lorong Alma Jaya 27,
Taman Alma Jaya, 14000 Bukit
Mertajam, Pulau Pinang.

Tarikh: _____

(TANDATANGAN PENYELIA)

DR. S.M WARUSIA BIN S.M.M
YASSIN

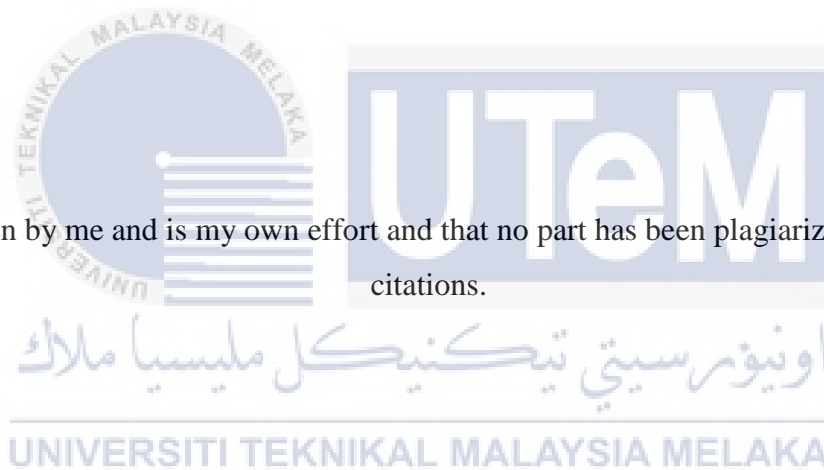
Tarikh: _____

CATATAN: * Tesis dimaksudkan sebagai Laporan Akhir Projek Sarjana Muda (PSM)
** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa.

DECLARATION

I hereby declare that this project report entitled
**CHI-SQUARE STATISTICAL AND SUPPORT VECTOR MACHINE FOR
ANDROID MALWARE DETECTION**

Is written by me and is my own effort and that no part has been plagiarized without citations.



STUDENT: _____ Date: _____

(MUHAMMAD ZULHILMI BIN JOHARI)

SUPERVISOR: _____ Date: _____

(DR. S.M WARUSIA MOHAMED)

DEDICATION

Dear Parent

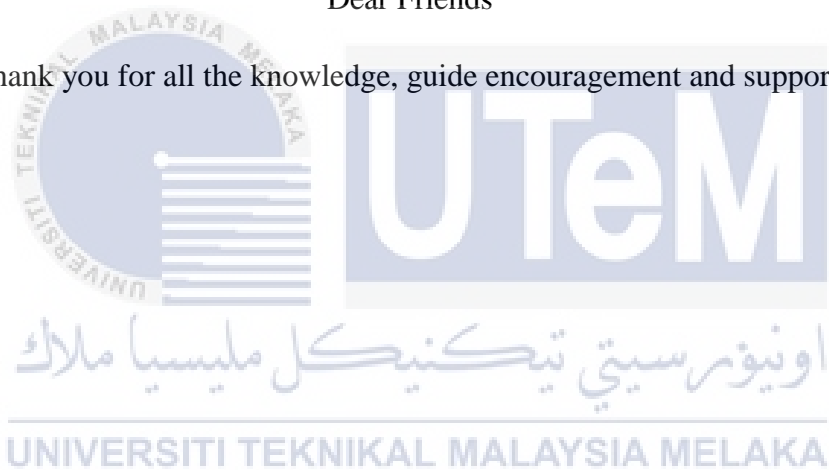
Thank you for your love and sacrifice.

Dear Supervisor and Evaluators

Thank you for all the knowledge and guidance for me to reach at this level.

Dear Friends

Thank you for all the knowledge, guide encouragement and support.



ACKNOWLEDGEMENT

First of all, I would like to express my very great appreciation to my supervisor, Dr. SM Warusia Mohamed for guide and give useful critiques and advise for this Feature Selection and Classification for Android Malware Detection project. I also would like to thanks to him and Sir Zaki Bin Mas'ud for teaching and assists me in keeping my progress on schedule. Furthermore, I would like to thanks to my friends and my BITZ course mate for helping me to complete my project. Lastly, I wish to thanks and give appreciation to my parents for their support and encouragement throughout my study.



ABSTRACT

This project focus on feature selection and classification approach for android malware. Feature selection step is reducing the available features to a set that is optimal or sub-optimal and capable of producing results which are equal or better to that of the original set. This technique is used for descriptors and allows survey response to be put in into meaningful categories in order to have the useful data. The classification of malware based on behaviour of different malware is done by using data mining techniques. Classifier is a supervised learning and requires training data accompanied by labelling the class and the new data is classified based on the training data set. So, this project is significantly to propose combinational method for better android malware detection. The problem statements for this project is there are too many irrelevant Features that make hard to detect android malware behaviour and also difficult to differentiate malware activity more precisely. So, the objective of this project is to obtain android malware activity more accurately using Support Vector Machine and propose Chi-square as Feature Selection method to identify relevant features accurately. In addition, the best possible accuracy and detection rate can be achieved by using feature selection method which is Chi-square. Throughout the experiment, accuracy and detection rate for android malware activity improved by applying Chi Square and Support Vector Machine.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

ABSTRAK

Projek ini fokus pada pendekatan pilihan rencana dan pengelasan untuk perisian perosak android. Langkah pilihan rencana mengurangkan ciri-ciri boleh didapati kepada satu set yang optimum atau sub optimum dan berkebolehan menghasilkan hasil yang mana sama jadi atau lebih baik dengan persekitaran asal. Teknik ini digunakan untuk pemerihal dan membenarkan sambutan tinjauan dimasukkan ke dalam kategori-kategori bermakna supaya mempunyai data yang bermanfaat. Pengelasan perisian perosak berasaskan kelakuan perisian perosak berbeza dibuat dengan menggunakan data teknik-teknik perlombongan. Pengelas ialah satu pembelajaran yang diawasi dan memerlukan data latihan diiringi oleh pelabelan kelas dan data baru diklasifikasikan berdasarkan set data latihan. Jadi, projek ini nyata sekali mengusulkan kaedah combinational untuk pengesanan perisian perosak android lebih baik. Penyataan masalah bagi projek ini ialah terdapat terlalu banyak Features tidak relevan yang membuat sukar untuk mengesan kelakuan perisian perosak android dan juga sukar membezakan kegiatan perisian perosak dengan lebih tepat. Jadi, objektif projek ini adalah untuk mendapatkan kegiatan perisian perosak android dengan lebih tepat menggunakan Support Vector Machine dan mencadangkan Chi-square sebagai kaedah Feature Selection mengenal pasti ciri-ciri relevan dengan tepat. Sebagai tambahan, kemungkinan ketepatan terbaik dan kadar pengesanan dapat tercapai dengan menggunakan kaedah pilihan rencana yang merupakan Chi-square. Di seluruh eksperimen, ketepatan dan kadar pengesanan untuk kegiatan perisian perosak android diperbaiki dengan menggunakan Chi Square and Support Vector Machine.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

TABLE OF CONTENTS

BORANG PENGESAHAN STATUS TESIS	ii
DECLARATION	iii
DEDICATION	iv
ACKNOWLEDGEMENT	v
ABSTRACT	vi
ABSTRAK	vii
LIST OF TABLES	xi
LIST OF FIGURES	xii
CHAPTER 1	14
INTRODUCTION	14
1.1 Introduction	14
1.2 Project Background	15
1.3 Problem Statement	16
1.4 Project Question	16
1.5 Project Objective	17
1.6 Scope	17
1.7 Expected Output	18
1.8 Thesis Organization	19
1.9 Summary	20
CHAPTER 2	21
LITERATURE REVIEW	21
2.1 Introduction	21
2.2 Malware	23
2.3 Detection Scheme	23
2.3.1 Intrusion Detection System (IDS)	24
2.3.2 Intrusion prevention System (IPS)	26
2.4 Types of malware	27
2.4.1 Computer malware	28
2.4.2 Android malware	31
2.5 Detection techniques	37
2.5.1 Signature detection	37
2.5.2 Anomaly Detection	38
2.6 Data Mining	39
	viii

2.6.1	Feature Selection	39
2.6.2	Classification	42
2.6.3	Clustering	44
2.7	Summary	49
CHAPTER 3		50
METHODOLOGY		50
3.1	Introduction	50
3.2	Methodology	50
3.2.1	Literature Review Phase	52
3.2.2	Data Collection Phase	52
3.2.3	Data Validation Phase	52
3.2.4	Feature Selection + Classification Phase	53
3.2.5	Evaluate Phase	54
3.2.6	Documentation phase	54
3.3	Research process	55
3.4	Software and Hardware Requirement	55
3.4.1	Software Requirement	55
3.4.2	Hardware Requirement	57
3.5	Project Milestone	58
3.6	Summary	60
CHAPTER 4		61
DESIGN		61
4.1	Introduction	61
4.2	Malware Analysis Progress	62
4.2.1	Data Collection	63
4.2.2	Feature Selection	66
4.2.3	Support Vector Machine	70
4.3	Method FS_SVM	73
4.3.1	Pseudo code FS_SVM	75
4.5	Summary	76
CHAPTER 5		77
IMPLEMENTATION		77
5.1	Introduction	77
5.2	Environment Set Up	77
5.3	Design Process	80

5.3.1	Add Data	81
5.3.2	Basic SVM Process	82
5.4	Connect to MYSQL	86
5.5	Conclusions	88
CHAPTER 6		89
RESULT AND ANALYSIS		89
6.1	Introduction	89
6.2	Results of Suggested Approaches	89
6.2.1	Basic SVM Process	90
6.2.2	Chi Square + SVM Process	91
6.3	Calculation Result	95
6.4	Conclusions	96
CHAPTER 7		97
CONCLUSIONS		97
7.1	Introduction	97
7.2	Project Summarization	97
7.3	Project Limitation	99
7.4	Future Works	99
7.5	Conclusions	100
REFERENCES		101



LIST OF TABLES

TABLE	TITLE	PAGE
1.1	Problem Statement	3
1.2	Project Question	3
1.3	Project Objective	4
2.1	Advantages of Types of IDS	12
2.2	Categories of Adverts	21
2.3	Advantages and Disadvantages of Signature-Based	25
2.4	Collection of Journal	32
3.1	Project Milestones	45
6.1	Weight of Chi Square Statistics	81
6.2	Result of Training Different Algorithm	82
6.3	Result of Testing Table	82



LIST OF FIGURES

FIGURE	TITLE	PAGE
2.1	Chapter Outline Diagram	9
2.2	IDS Functionality	12
2.3	Features of IPS	14
2.4	Feature Selection Process	28
2.5	System Diagram for Malware Classifiers	29
3.1	Project Methodology	38
3.2	Architecture for Android Malware Detection	42
4.1	Android Malware Analysis Progress	49
4.2	Data Collection Procedure	50
4.3	Data Collection	51
4.4	Flow Process of Collecting Data	52
4.5	A General Framework of Feature Selection of Classification	54
4.6	Flow Process of Feature Selection	55
4.7	Support Vector Machine Process Method	59
4.8	Flow Process of Support Vector Machine	59
4.9	Flow Process of FS_SVM	61
5.1	Detailed Framework of Implementation	65
5.2	Dataset Used	66
5.3	Chart of Dataset	67
5.4	Step of Add Data	68
5.5	Interface of Rapid Miner Studio	68
5.6	Basic Design Process	69
5.7	Design Process of Training and Testing Phase	70
5.8	Parameters of Cross Validation	71
5.9	Parameters of Write Database	72
5.10	Apply Model and Performance Operator	72
5.11	Interface of MySQL Workbench	73
5.12	SQL Query Calculation of Results	74
6.1	Results of testing SVM Basic Process	77
6.2	Charts of training SVM	78

6.3	Results of testing Chi Square +SVM	79
6.4	Charts of SVM with Chi Square	80

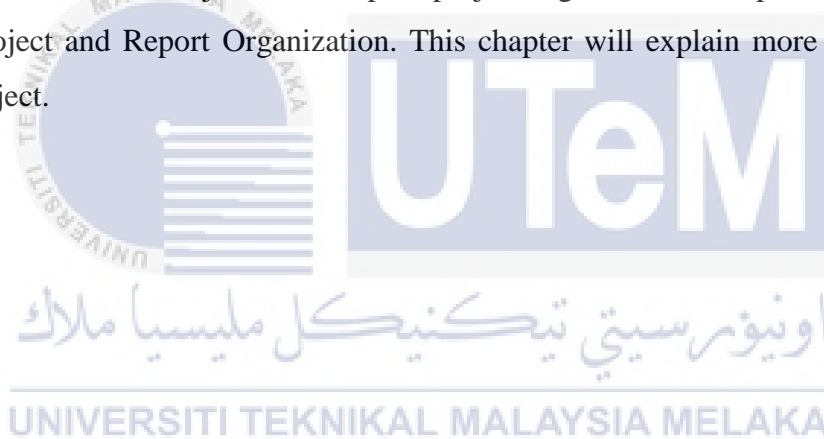


CHAPTER 1

INTRODUCTION

1.1 Introduction

The first chapter of this project will discuss about the project background, problem statement, objectives, scopes, project significances, expected output from this project and Report Organization. This chapter will explain more detailed about the project.



1.2 Project Background

This project focuses on feature selection and classification for android malware. Feature selection step is reducing the available features to a set that is optimal or sub-optimal and capable of producing results which are equal or better to that of the original set. This technique is used for descriptors and allows survey response to be put in into meaningful categories in order to have the useful data. Classification such as SVM is applied to make the system gain to accommodate the function of the new malicious software sample, so as to detect new malicious software and existing malwares. In this work, we use two classes which are malware and clean. All type of data can be categorized easily because of using this technique.

This project is significant because everybody can put their own application to allow other user to download, and then it will cause the limited control on malware application has been distributed. There are too many tool for repacking existing application with malicious code inside that, and because of this, it is too difficult to detect malware application using know day tool. Feature Selection can improves the accuracy of the final result and makes the data mining results easier to understand and more applicable. Beside, this project is significant because it help and guide other researcher on several issues such as technique to choose for android malware data and the ways to perform data collection.

This project focus on determine the malicious behaviour of android malware using Feature Selection method which is Chi-square. Since Support Vector Machine method combines the characteristics of the malicious software categories and operating environment to record the behavior of the malicious software, so, this project can obtain android malware activity more accurately.

1.3 Problem Statement

The problem statement part will be discussed about the current problem need to solve in this project. There are several classifier techniques through machine learning, but there has not been a proof which technique will produce the best result and performance for android malware behaviour detection.

Table 1.1: Problem Statement

No	Problem Statement
PS 1	It is hard to detect android malware behaviour because there are many irrelevant features.
PS 2	There is no comparative study on malware behaviour detection on android platform using classification method. It will be very difficult to differentiate malware activity more precisely.

1.4 Project Question

Table 1.2: Project Question

No	Project Question
PQ 1	How to collect data for classifier on android malware platform?
PQ 2	How to select appropriate feature from dataset?
PQ 3	Which classifier will produce the accurate result to detect malware?

1.5 Project Objective

Based on the problem statement, there are problems that need to solve. So, the project objective is built to overcome the problem statement. This project is based on behaviour android malware and we need to classify the method that most suitable for classification android malware.

Table 1.3: Project Objective

No	Project Objective
PO 1	To propose Chi-square as Feature Selection method to identify relevant features accurately.
PO 2	To obtain android malware activity more accurately using Support Vector Machine.

1.6 Scope

For the project scope, it will divide into three parts which is type of android malware, software and hardware requirement in this project.

1. Case study

To perform the data collection on this project, it will require a malware with different variant. The dataset were collected a total of 558 APK. . The collection is consisted of 279 applications with low privileges of the free access dataset and a random selection of 279 malwares of the MalGenome. The sample of malware are divided into Type 1 or 0 which means 1 is malware and 0 is non-malware or clean. Then, the data was analysed by Feature Selection method which is Chi-square and classify whether it is either attack or non-attack.

2. Software

For the software requirement in this project, we use Microsoft Office Word, Microsoft Project, Microsoft Visio, Microsoft Excel, Wireshark, Rapid miner and SQL.

3. Hardware

The hardware requirement that needed in this project are computer and workstation.

1.7 Expected Output

This propose project has its significance and also has the advantages:

1. Towards body of knowledge.
 - i. Performance Indicator.

This project may be produce the performance indicator to use to evaluate the performance of classification technique for android malware data.

- ii. Malware Behaviour

This project may be provide list of android malware behaviour that created by application malware.

- iii. Classify malware behaviour using Feature Selection and Support Vector Machine. After all phase, this project can provide the accurate algorithm to use for other researcher to classify android data.

2. Data Collection

This project may provide a formal procedure to perform data collection for other researcher.

1.8 Thesis Organization

CHAPTER 1: Introduction

This chapter explained about the definition, background, problem statement, objective, scope and expected output that related to the android malware detection. The overview and the introduction about the project is also discussed.

CHAPTER 2: Literature Review

This chapter review about the behaviour of malware, detection technique and the best classifier for android malware from the past researcher. It will help to understand about the malware detection through machine learning. In addition, this chapter also discuss about the previous work that related to this project.

CHAPTER 3: Methodology

In chapter 3, project methodology for the whole project will be discussed. The methodology involved are data collection, data validation, classification, evaluation and documentation. Besides, Gantt chart and milestone also need to be design for this project. Software and hardware requirement also discussed in this chapter.

CHAPTER 4: Design

This chapter provides techniques used in this project. For analysis phase it will require data of malware behavior. On this phase it consists several parts which is step, procedure and application to use. This chapter also cover the flow process of data collection and technique used. The steps needs to follow to get a positive data to use for training data on machine learning.

CHAPTER 5: Implementation

This chapter describe the activity involved in the implementation phase and the expected output. Besides, this chapter include environment setup, parameters, variables and assumptions used in this project.

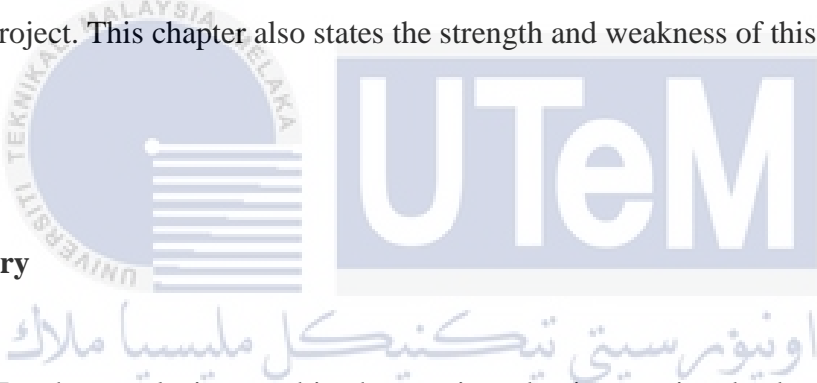
CHAPTER 6: Testing and Analysis

This chapter is about the activity in the implementation phase in the analysis of the project and include graphical results from critical analysis using the collected data.

CHAPTER 7: Project Conclusion

In this chapter summarization of the project and report all implementation and testing phase highlighted. Finally, it will conclude the significant results that will gain in this project. This chapter also states the strength and weakness of this project.

1.9 Summary



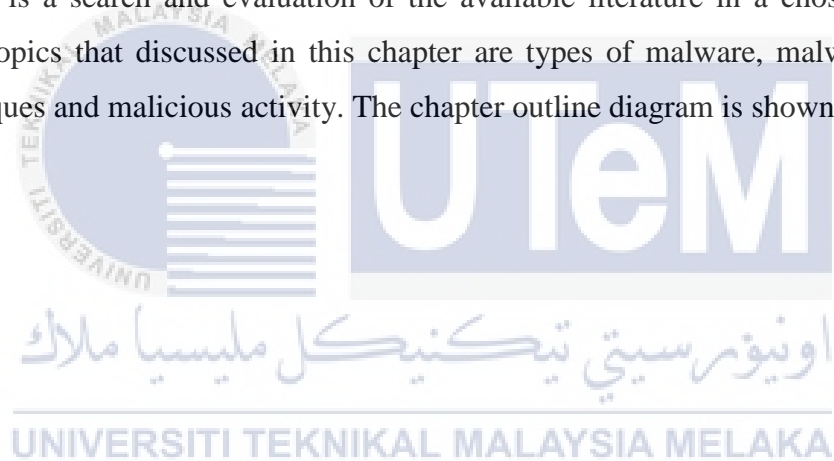
For the conclusion on this chapter, introduction, project background, detailed of the project, problem statement and also objective was already explained. Besides, it also explained the purpose and scope of the project. The next chapter of this project report is Literature Review. This chapter is very important in this paper project because it is a guideline for this project to decide what need to choose for the best classifier used to determine malware through machine learning by using a different technique.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, literature review of the project will be explained. Literature review is a search and evaluation of the available literature in a chosen topic. The main topics that discussed in this chapter are types of malware, malware detection techniques and malicious activity. The chapter outline diagram is shown in Figure 2.1.



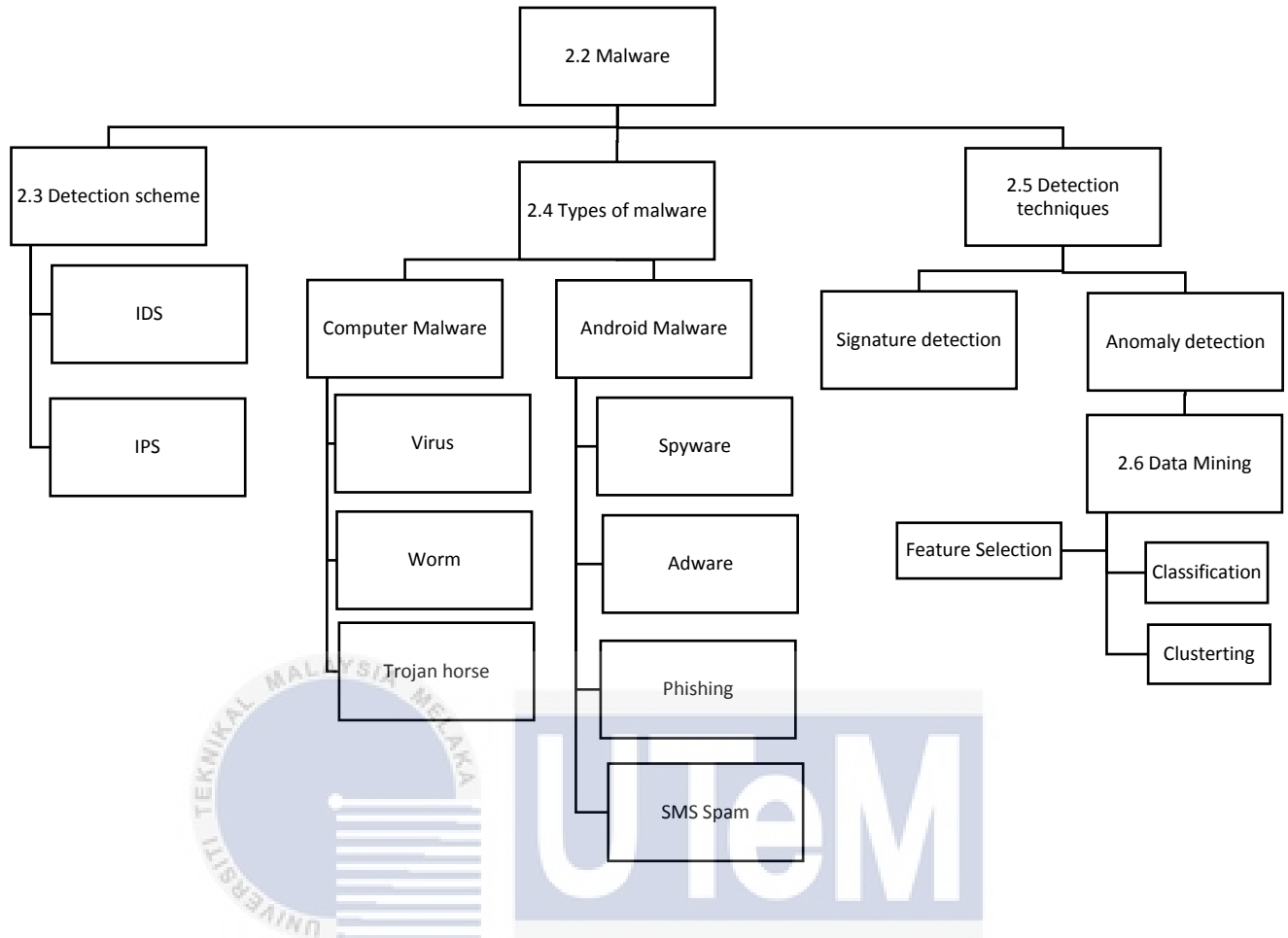


Figure 2.1: Chapter Outline Diagram

Based on figure 2.1, malware outline diagram was created. In the first section, it is discussed about detection scheme which cover about the Intrusion Detection System and Intrusion Prevention System. Then, types of malware which is computer and android malware also discussed. The examples of bot malware are Virus, Worm, and Trojan horse while mobile malware are spyware, Adware, Phishing apps and SMS Spam. The third section is the detection techniques which is discussed about the detection techniques that are used for malware detection. The techniques discussed are signature detection and anomaly detection. In anomaly detection, there is data mining. Three types of data mining which is classification, clustering and feature selection.

2.2 Malware

Malware or is a set of instructions that run on your computer or software and make your system do what attacker wants it to do. Malware is considered to be a major computer security problem as it can attach to other computer programs and infect them (F.Cohen, 2015). According in (Wikipedia, 2017), Malware is a term used to refer a variety of forms of unsympathetic or intrusive software including computer viruses, worms and other malicious programs. It can take form of executable code and script content and other software. Smart devices like smartphones are becoming more popular in current trends by keeping the people to be connected always. The popularity of smartphone usage attracts the malware authors to thefts the private information of users and to affect the device in several ways.

2.3 Detection Scheme

Notwithstanding, mobile computing has been faced with many challenges since its inception. These include high electromagnetic interference, limited bandwidth and low security. Fault detection concerns itself with monitoring a system, identifying the occurrence of a fault and identifying the type of fault as well as its location. Faults are unavoidable in large and complex communication networks. However, quick troubleshooting can significantly improve network reliability (Puhan Zhang and Yufang Sun, 2001). Fault detection is utilized to determine whether or not a problem has occurred in a certain channel or area of operation. In other words, the software application may recognize that the system is operating successfully, but performing sub- optimally to the pre-determined target. The software application identifies the reason for the sub-optimal performance so that the organization can troubleshoot it. Fault detection can also be automated for greater flexibility and efficiency.

Advances in wireless networking have prompted a new concept of computing, called mobile computing; in which users carrying portable devices have access to a shared infrastructure, independent of their physical location. Technology has evolved so rapidly in the recent years that computing can be available everywhere over the mobile technology in a distributed manner (Michael F. Goodchild and Douglas M. Johnston, 2004). Generally the detection scheme can be grouped into two major group such as IDSs and IPSs (Yingbing, Y. 2012). However the major concern in this project are IDSs.

2.3.1 Intrusion Detection System (IDS)

An Intrusion Detection System is an application used for monitoring the network and protecting it from the intruder. With the rapid progress in the internet based technology new application areas for computer network have emerged (PeymanKabiri and Ali A.Ghorbani, 2005). For example, the fields like business, financial, industry, security and healthcare sectors the LAN and WAN applications have progressed. All of these application areas made the network an attractive target for the abuse and a big vulnerability for the community. Malicious users or hackers use the organization's internal systems to collect information's and cause vulnerabilities like Software bugs, Lapse in administration, leaving systems to default configuration. As the internet emerging into the society, new stuffs like viruses and worms are imported. The users use different techniques like cracking of password, detecting unencrypted text are used to cause vulnerabilities to the system. Hence, security is needed for the users to secure their system from the intruders. IDS are used in network related activities, medical applications, credit card frauds, Insurance agency (Christopher Low, 2005).

IDS have 2 different types which is host based and network based. Table 2.4 below shows the advantages of the types of IDS:

Table 2.1: Advantages of Types of IDS

Host based	Network based
Verifies success or failure of an attack	Lower Cost of Ownership
Monitors System Activities	Easier to deploy
Detects attacks that a network based IDS fail to detect	Detect network based attacks
Does not require additional hardware	Real Time detection and quick response.

According to the journal by (Dr. S.Vijayarani¹ and Ms. Maria Sylviaa.S, 2015), The IDS consist of four key functions namely, data collection, feature selection, analysis and action as in Figure 2.2.

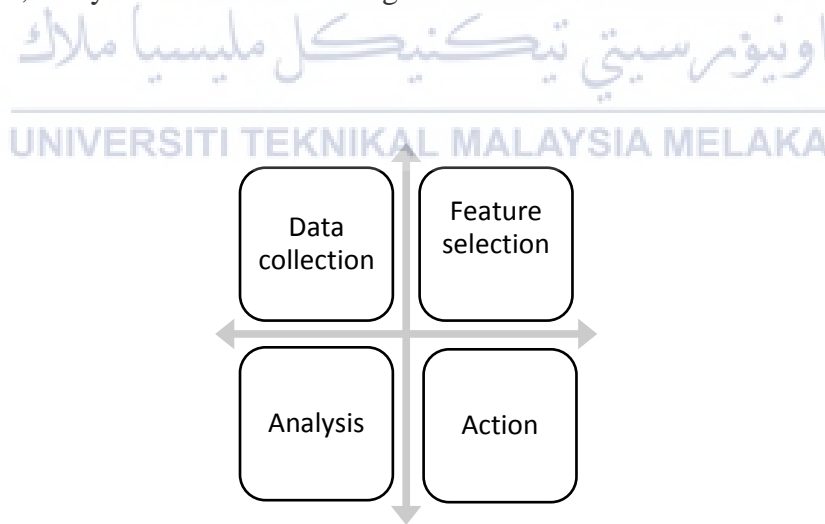


Figure 2.2: IDS Functionality.

2.3.2 Intrusion prevention System (IPS)

Intrusion Detection System was developed to identify and report the attack in the late 1990s, as hacker's attacks and network worms began to affect the internet, it detected hostile traffic and sent alerts but did nothing to stop the attacks (Y. Weinsberg and S. Tzur-David, 2006). It has been a long road for Intrusion Detection System (IDS), almost two decades since it has become a major issue. In other words, Intrusion Detection is passive. It is not able to detect all malicious programmes and activities most of the time and incompatible to integrate with control restriction to stop traffic inbound-outbound from attacking; which means it was only capable to detect attack actions, without prevention action.

Intrusion Prevention System (IPS) is primarily a network-based defence system, with increasing global network connectivity and combines the technique firewall with that of the IDS properly with proactive technique. This system is a proactive technique which prevents attacks before entering the network by examining various data record and detects demeanour pattern recognition sensor. When an attack is identified, intrusion prevention blocks and logs the offending data. Currently, requirement for a system to provide early detection / warning from intrusion security violation with knowledge based has become a necessity. Therefore, the system must be active and smart in classifying and distinguishing packet data, if curious or mischievous data are detected, alert is triggered and event response is executed. This mechanism is activated to terminate or allow packet data to process associated with the event. It prevents attack before entering the network by examining various data records and prevents demeanour of pattern recognition (D. Stiawan, 2010).

IPS functions as radar to monitor stream network traffic; detecting, identifying, and recognising any signal that could be considered a security violation. With respect from proposal work by (A. Piskozub, and N. Tymoshyk, 2007) they present real-time intrusion prevention and anomaly system. In (2011), L.Hu and W.Wang declared IPS has correlation between intrusion detection and firewall, also design and implementation of trusted communication protocol based on XML is provided, and then (E.E. Schultz and E. Ray, 2007) had predicted the future of IPS technology, such as:

1. Better underlying intrusion detection,
 2. Advancement in application-level analysis,
 3. More sophisticated response capabilities, and
 4. Integration of intrusion prevention into other security devices.
- Moreover, the prediction concerns on intrusion prevention technology which are very positive in market.

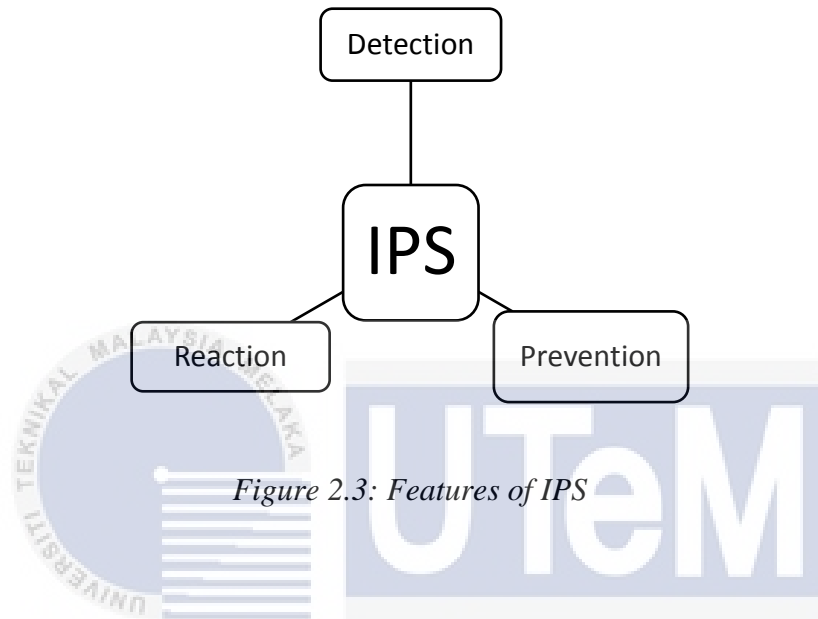


Figure 2.3: Features of IPS

2.4 Types of malware

Nowadays, there are many types of malware as the number of malware increased every day. Types of malware can be divide into two which is computer malware and mobile malware. The types of malware that will be discussed are virus, worm, Trojan horse, spyware, adware, and phishing and bot process due to their most common types of malware these days.

2.4.1 Computer malware

According to McAfee Labs Threats Report (August, 2015), computer malware is a global problem that causes significant economic losses and loss of productivity. Incidences of malware have been increasing consistently at a rate of 50% per year, and the number of unique malware samples has reached 433 million as of 2015. In addition, malware typically result in theft of personal and sensitive data, spam and denial of service attacks, as well as ransom ware. Malware development and exploitation is an industry with suppliers, markets, and service providers. In many cases, new computing equipment is sold with pirated software that contain malware. Also, malware may infect computing equipment from pirated software downloaded through the Internet or bought from vendors. The amount of computers will still grow in the next years. Not only will the number of desktop computers and large computer systems increase but the number of embedded systems as well (T. Ungerer, 2008). The microcomputer will significantly influence everyday life. They will be used for human health monitoring, in the traffic management or in aerial/space technologies. The dependency of single persons and the society on these systems will definitely grow. Today almost all digital attacks are directed against personal computer and mainframes/server. The embedded systems in the sense of microcontrollers have not yet been discovered as an attack target.

Today IT security research is focused on desktop and desktop-like computers. However, such technologies are only applicable with certain computer families. The need for uniform methods and structures to improve security of several computer systems will definitely grow in the future. This work describes a general hardware architecture that will improve the security.

2.4.1.1 Virus

Virus is a program with harmful intent that injects a malicious code into existing programs to replicate itself. A virus might corrupt or delete data on your computer, use your email program to spread itself to other components, or even erase everything on your hard disk. Viruses are most easily spread by attachments in email messages or instant messaging messages. With the development of computer technology and communication technology, the number of viruses has increased dramatically, which have brought about huge financial losses (L.A Gordon, M.P Loeb, W. Lucyshyn, R. Richardson, 2005). Virus may be spread to another computers through network or corrupted media such as USB drives and floppy disk (Vinod, P *et al.*, 2009). Most of viruses are pretty harmless.

The user might not notice the virus. Sometimes viruses might causes random damage to data files and over a long period they might destroy files and disks. According to the Essam Al Daoud et.al (2008), virus can be detected by a few methods which is string scanning method, skeleton method and by using bookmarks method.

2.4.1.2 Worms

Worms are generally self-propagating, though the line between worms and viruses blurs when discussing mass-mailing viruses, since they self-propagate but usually rely on the receiving user to activate them (N. Weaver and V. Paxson, 2003). The behaviour or characteristics of worms is it infects a victim host and then the host begins to infect other victims (Kim and Karp, 2004). Worm spreads using network to send copies of itself without user's authorization. Worm may cause harm to the network by consuming extraordinary amount of bandwidth (Al Amro and Alkhalifah, 2015). The worms starts with exploiting to control a machine. The exploit is normally a buffer overflow attack caused by sending packets that contains more data than the allocated buffer size.

In order to detect worm, their characteristics such as destination ports and payload length obtained from scanning the worm's activity is used (Chalurkar and Meshram, 2012). An example of the ports used by the Sasser worm to attack port 445. Besides, the hash value of the content is an efficient identifier that allows the detection scheme to identify known worm in real time and with low memory. If both hash value and the static destination port for a worm are used as the parameter for detection, it can be more efficient (James Joshi, 2008).

2.4.1.3 Trojan horse

Trojan horse masquerade as benign program but perform malicious activities (Kolter and Maloof, 2004). According to Al Amro and Alkhalifah (2015), Trojan horse imitate the behaviour of program such as login shell and hijacks user password to gain control of system remotely. Trojan horse does not replicate itself and it is usually spread when the host downloads or open attachments. According to Vinod, P *et al.* (2009), Trojan horse can damage the system resources such as files and disk data. Trojan can cause Denial of Services (DoS) attack where a single internet connection will conduct attacks against a targeted web address by sending multiple requests that will overwhelm the target and eventually leads to denial of service. In order to detect Trojan, it is important to figure out the objects of Trojan horse operations such as the operations on registry, file, port, process, system service and other I/O interfaces such as keyboard. Trojan horse consists of program codes that calls for different API functions in order to run on the target computer. Thus, API hooking is a way to detect a Trojan (Vamshi Krisna Gudipati, 2015).

2.4.2 Android malware

In recent years, as smartphones become a dominant device of choice for performing most of our daily activities, with Android claiming 82% market share (Ramon Llamas and Ryan Reith, 2015), a rapidly increasing number of Android applications are being developed to make user experience more convenient. Unfortunately, more convenient usually also means less secure, which is particularly true for Android platform. Android is open source with low barriers to enter; while this may help to grow a huge collection of third-party apps quickly that makes up a major component of the Android experience, it also makes the Android platform an easy target for malware intrusion. According to the mobile threat report released by F-Secure in 2014, over 95% of malicious apps were distributed on the Android platform (Yajin Zhou and Xuxian Jiang, 2012).

Recently, there have been some research efforts on detecting Android malware by using various machine learning algorithms. APIs and IP address as features and uses the Support Vector Machine (SVM) algorithm to learn a classifier from the existing ground truth datasets, which can then be used to detect unknown malware (Daniel Arp and Michael Spreitzenbarth, 2014). While (Dong-Jie Wu and Ching-Hao Mao, 2012), alternatively applies KNN (K-Nearest Neighbor) algorithm to permissions and intents to identify malware. In addition, (Yousra Aafer, Wenliang Du, 2013), focuses on providing several lightweight classifiers based on the API level features. However, Android malicious apps are becoming increasingly difficult for traditional malware detection programs to identify, with the ability to perform code obfuscation.

2.4.2.1 Spyware

Spyware represents one of the main internet threats. It represents one of the most significant security threats for individual users and organizational networks. Spyware is a software program which can compromise the covert and the overt of the confidential and non-confidential information from the Internet users' PCs and/or networks (Dinev and Hu, 2007). These programs have the ability to hide in the background on victim's systems or move through the Internet or local networks. Other types of spyware are remotely controlled and directed by its creator commands (Thomson et al., 2006). Spyware can efficiently attack Internet user's privacy by data collecting, system controlling, and/or actions reporting of victims PCs and networks (Cordes, 2005). Internet users are targeted by a number of threats that intent to get an unauthorized access to their private and critical data or trying to compromise their system privacy where their systems are often not well protected.

Awareness is defined as the extent to which a target population is conscious of an innovation and formulates a general perception of what it entails (Schmidt et al, 2008). Moreover, the number of users of this technology had been increased, which give the above research result an important part for Internet security resolving and improvement. In the investigation of Internet consumers' awareness about spyware, Zhang (2005) found that the general comprehension and understanding about security, confidentiality, and Spyware are lacking. Moreover, the privacy attacks are not sufficiently noticed by the users, with respondents having limited knowledge about using Spyware removal tools.

Kucera et al. (2005) reported about the presence of spyware in many of the popular freeware and shareware that may download from Internet websites. Once it downloaded, it has been found that this software has the ability to gather various forms of victims' personal information. Meanwhile, Lee and Kozar (2005) identified 3 types of factors that had significant impacts upon Internet users' adoption of anti-spyware protection. Firstly, they determined two user attitude factors; namely users' compatible morals and relative advantages of anti-spyware usages. Secondly, they determined two social influence factors; namely the visibility and image of Spyware

threats. Finally, they stated two behavioural control factors; computing capacity and the ability to test/try the protection product.

2.4.2.2 Adware

Adware is software which generally makes pop-up, banner etc. advertisements to appear on the user's computer (D. Evett, 2006). According to (MacAfee) the intention of the creator is usually to create revenue. Adverts may be downloaded or sometimes contained in free programs. For instance, Skype and Yahoo messenger have adverts. Even though some software offers the selection not to set up the additional adverts, some appear to sneak it in without the user's consent. Because of this, they are usually referred to as irritating people (D. Evett, 2006). They are difficult to remove once installed on a PC. The adware might be in the user interface of the software. In addition to that, it will be presented to people on the monitor while the software is being installed. The adware may be planned to analyse what kind of web sites users use in order to display relevant adverts to the sorts of things or services featured on the screen. The use of adware publicly started in 1987, which the Usenet newsgroup comp.sys.mac. used it on the internet for entertaining purpose, the post refers to a Macintosh program instead of a Windows program (E. Chien, 2005). It is hard to categorize all the possible kinds of adware (J. Aycock, 2009). The categories of adverts are:

Table 2.2: Categories of Adverts

Banner adverts	Most common type of adware. It usually seems a small strip at the top of the web site.
Pop-under adverts	These adverts are precisely the same as a pop-up advertisement. But this advertisement opens another window at the back of the current web page.
Floating adverts	Floating advertisement is designed inside the current web page. They probably prevent the user from seeing the windows appropriately.
Tear-back adverts	The tear-back is a variation on the floating advert. When users click on the advert, they will see a teaser. It looks like a dog-eared book page and it will “tear back” to reveal the advertisement.
In-text adverts	This advert is not similar to the other kinds of adverts that we have looked until now. This is because content is altered. Keywords contain links in the content. If the mouse goes over them the adverts become visible.
Transition adverts	This advert is inserted in among two web pages of content. For instance, when the user clicks on the content link, the advert may become visible, followed ultimately by the next page of content
Video adverts	This advert content permits advertisement methods application from television with the probability of the user interaction. This advert has two different types which are linear and non-linear Linear advert.

2.4.2.3 Phishing

Mobile phishing is an emerging threat in today's connected world. In a mobile phishing attack, an attacker usually sends an SMS message containing links to phishing web pages or applications which, if visited, ask for credential information. Phishing attacks is an effort for accessing people important information like username, password and credit cards information using social engineering techniques (Dadkhah M, Jaz, 2014). Attacks can also be initiated via email messages loaded in the browser of mobile devices. A report finds that the number of mobile phishing attacks has been increasing over the last few years for various mobile device platforms (Ashford, W. 2014). Compared with traditional desktop software users, mobile application users are more vulnerable to phishing H. Shahriar et al. 207 attacks at least three times (Kessem, L. 2012). Experts agree on some of the common, well-known reasons for this vulnerability:

- 1) Within a small device, it is rather difficult for a user to check whether a page is legitimate, as is confirming the actual pointed hyperlinks, as URLs are not often displayed within mobile browsers.
- 2) Mobile users are less aware of security options to stop or prevent phishing attacks.
- 3) Most legitimate mobile applications require users to enter their credentials with very simple user interfaces, making the job of an attacker rather easy to come up with fake apps or plain websites mimicking legitimate user interfaces.
- 4) Surveys find that 40% of mobile application users enter passwords into their phones at least once.

Based on the Symantec Internet Security Threat Report 2014, Vol. 19, the scope of phishing attacks is vast, and the consequences can be severe. On the other hand, there is limited knowledge among users on how to avoid phishing attacks. Symantec's Norton Report shows that 44% of adults are unaware that security solutions exist for mobile devices. This clearly shows not only the lack of awareness, but also the danger posed by mobile application phishing attacks. A phishing or fraudulent mobile application potentially can grab victim's account information and data stored on mobile devices

2.4.2.4 SMS spam

For the past two decades, the Short Messaging Service (SMS) has gained tremendous popularity throughout the world. Reports estimate billions of text messages handled daily by cellular providers' messaging infrastructures (V. Shannon, 2007), generating millions of dollars of yearly revenue (M. Ablot, 2011). Being unquestionably successful, text messaging is steadily becoming an annoyance due to the surge of SMS fraudulent activities such as spam and the spreading of malware two of the main examples (N. Perlroth, 2012). Spam is the widely adopted name to refer to unwanted messages that are massively sent to a large number of recipients. This kind of messaging abuse is a well-known and tackled problem in the context of electronic mail (e-mail). Numerous applications detect and block spam e-mails daily resulting in a small amount of spam reaching customer's inboxes and it is common nowadays to have anti-spam engines integrated into e-mail services. Based on (2009 Annual Security Report), these anti e-mail spam services are very effective, especially given the estimates indicating that 90% of the daily electronic mail traversing the Internet is spam.

The defense against message abuse often relies on SIM shutdowns and subsequent account cancelations. However, as our results support, this does not stop most spammers, though, who purchase multiple cards and swap them to limit the daily per-SIM volume (A. Bobotek, 2010). Message abusers also rapidly replace cancelled SIM cards to continue their spam campaigns. Millions of illegitimate text messages are transmitted via cellular networks daily (N. Perlroth, 2012). These messages consume network resources that could be allocated to legitimate services otherwise. SMS spam results also in a major inconvenience for cellular customers because, without an unlimited plan, the end user is paying at a per received message basis. Therefore, SMS spam potentially generates unwanted bill charges for some users leading to negative messaging experience and customer dissatisfaction. Spam also exposes smartphone users to attacks. Often multiple fraudulent messaging activities such as phishing, identity theft and fraud are related to SMS spam (N. Zablotskaya, 2008). SMS is also known as an entry vector for malware propagation (I. Murynets and R. Piqueras Jover, 2012).

Beyond the results presented in this paper expands our dive into SMS spam with the following contributions:

- Analyze certain behaviour and strategy change of spammers over six months.
- Analyze the content of the spam messages and identify certain common spamming campaigns.
- Identify the device reuse of spamming tools after account cancellation.

2.5 Detection techniques

2.5.1 Signature detection

Signature-based detection is an anti-malware approach that identifies the presence of a malware infection or instance by matching at least one byte code pattern of the software in question with the database of signatures of known malicious programs, also known as blacklists. This detection scheme is based on the assumption that malware can be described through patterns that also called signatures (Hwang K and Cai M, 2007). Signature-based detection is the most commonly used technique for anti-malware systems. Since the signature-based anti-malware systems are constructed on the basis of known malware, they are unable to detect unknown malware, or even variants of known malware. Thus, without accurate signatures, they cannot effectively detect polymorphic malware (J.P Anderson, 2012). Therefore, signature-based detection does not provide zero-day protection. Moreover, since a signature-based detector uses a separate signature for each malware variant, the database of signatures grows at an exponential rate. According to Kumar.V and Sangwon.O.P (2012), an example of Signature based Intrusion Detection System is SNORT. The advantages and disadvantages of signature-based detection is also discussed in Table 2.3

Table 2.3: Advantages and Disadvantages of Signature-Based Detection.

Advantages	Disadvantages
Misuse detection system begins protecting your network immediately upon installation.	How to keep up with large volume of incoming traffic when each packet needs to be compared with every signature in the database
Signature-Based Detection is easy to use.	Misuse detection system must have a signature defined for all of the possible attacks that an attacker may launch against your network.
There are low false positives as long as attacks are clearly defined in advance.	Misuse detection has a well-known problem of raising alerts regardless of the outcome.

2.5.2 Anomaly Detection

Anomaly detection aims to detect anomalous observations from a system. In the machine learning version of this problem we cannot directly model the normal behaviour of the system since it is either unknown or too complex. Instead, we have some sample observations from which the normal behaviour is to be learned. This anomaly detection learning problem has many important (Yeung and Chow, 2002; Fan et al., 2001). Based on journal Neda Noori (2012), anomalies can be classified into following categories:

1. Point anomalies: If an individual data instance can be considered as anomalous with respect to the rest of data, then the instance is termed as a point anomaly. This is the simplest type of anomaly and is the focus of majority of research on anomaly detection.
2. Contextual Anomalies: If a data instance is anomalous in a specific context and concept, then it is termed as a contextual anomaly (Jermaine, C., and Ranka, S. 2007). Each data instance is defined using following two sets of attributes:

- a. Contextual attributes: The contextual attributes are used to determine the context or neighbourhood for that instance.
- b. Behavioural attributes: The behavioural attributes define the non-contextual characteristics of an instance.

Detecting anomalies in data has been studied in the statistics community as early as the 19th century. Over time, a variety of anomaly detection techniques have been developed in several research communities. Many of these techniques have been specifically developed for certain application domains, while others are more generic. The previous survey tries to provide a structured and comprehensive overview of the research on anomaly detection. In 2004, Hodge and Austin provide an extensive survey of anomaly detection techniques developed in machine learning and statistical domains. In 2006, a broad review of anomaly detection techniques for numerous data is presented by Agyemang. In 2003, an extensive review of anomaly detection techniques using neural networks and statistical approaches has been presented. In 2007, Patch and Park presented a survey of anomaly detection techniques used specifically for cyber-intrusion detection.

2.6 Data Mining

2.6.1 Feature Selection

One of the detection methods is by combining automatic dynamic behaviour malware analysis with data machine learning classification techniques. However, collecting large number of extracted features that is irrelevant to the machine learning classification can lead to some classifier drawback such as misleading the learning algorithm, over-fitting, reducing generality and increasing model run-time (Shabtai, Asaf, Uri Kanonov, 2012). Feature selection or variable selection consists of reducing the available features to a set that is optimal or sub-optimal and capable of producing results which are equal or better to that of the original set. Reducing the feature set scales down the dimensionality of the data which in turn reduces the training time of

the induction algorithm selected and computational cost, improves the accuracy of the final result, and makes the data mining results easier to understand and more applicable (Guyon & Elisseeff, 2003; Kohavi & John, 1997).

While reducing the feature set may improve the performance of most classification algorithms, especially for K-NN algorithm, it may also lower the accuracy of decision trees (Li, Zhang, & Ogihara, 2004). Since decision trees have the capability of reducing the original feature set in the tree building process, beginning the process with fewer features may affect final performance. Dash and Liu (1997) broke down the feature selection process into four steps; generation, evaluation, stopping criterion, and validation. Feature selection has three types which is Filter, Wrapper and Embedded.

The filter method of feature selection reduces the number of features using properties of the data itself independently to what learning algorithm is eventually used (John, Kohavi, and Pfleger, 1994). One advantage of applying a filter algorithm to a feature set is that the number of features used in the final induction algorithm will be reduced. Therefore not only the performance of classification algorithms will be improved, but also amount of the computer processing time will be reduced. Unlike wrapper methods, filter methods do not incorporate the final learning algorithm in their process. This independency has been reported as another benefit of using filter methods (Ladha & Deepa, 2011). Wrapper algorithms use a preselected induction algorithm as part of the feature selection process. As features are added or subtracted the final results are ranked as to effectiveness of the selection. Since the induction algorithm itself is used in the evaluation phase of the selection process wrapper methods tend to score better results than filter methods. Kohavi and John (1997) compared the wrappers for feature subset selection against filter methods. They concluded that relevancy of attributes contribute greatly to the performance of the learning algorithms when the algorithm is taken into consideration. However, there are some limitations to these methods. The computational cost of running the evaluation is far greater than that of filter methods and increases as the number of attributes increases. Another disadvantage of the wrapper method is the likelihood of over-fitting the data.

Meanwhile, Support Vector Machine (SVM) is a supervised learning model with associated learning algorithms that analyze data used for classification and regression analysis. According to Drebin (2014), he uses the Support Vector Machine (SVM) algorithm to learn a classifier from the existing ground truth datasets, which can then be used to detect unknown malware. D. Fradkin (2006) said that the Support Vector Machine (SVM) is one of the classification methods that has an efficient training process and can be optimised in all field issues. This is corroborated by the opinion of P. Wang (2014) stating that the SVM classification algorithms are more accurate than those using other machine learning approaches involving non-optimised search methods, such as artificial neural networks, least squares, knearest neighbour, bayesian probability, and classification and regression trees particularly when defence systems collect only limited training data.

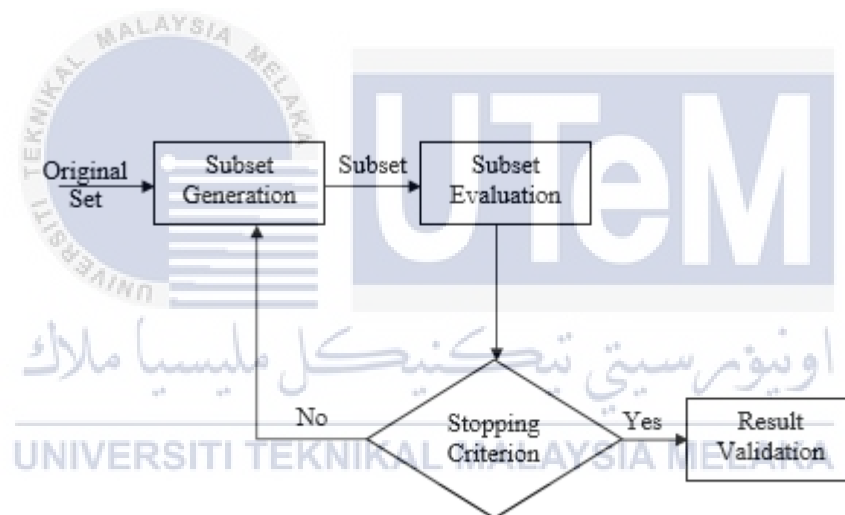


Figure 2.4: Feature Selection Process

2.6.2 Classification

According to Microsoft Research One Microsoft Way (2013), malware classification systems are often based on sparse, binary feature sets. In the research, they also employ sparse binary features based on file strings, application programming interface (API) tri-grams, and API call plus parameter value. To achieve good classification accuracy, they use over 179 thousand sparse, binary features generated from feature selection. Logistic regression on all of the features demonstrates reasonable accuracies at large-scale. However, the error rates are still not small enough for fully automated malware classification. In addition, the high-dimensionality of the input space prevents more complicated algorithms from being utilized. They also investigate using principal component analysis (PCA) to reduce the dimensionality of the input vector (Ping LI and Kenneth W.Church, 2006). This paper makes the following contributions:

- A large-scale system to classify unknown files with random projections and neural networks is proposed and implemented, and the results are presented.
- Random projections are used to reduce the dimensionality of the input space by a factor of 45 allowing a neural network to be trained on the high-dimensional input data.
- Investigate the number of random projections, hidden layers, and hidden units required to achieve good performance and compare the performance of logistic regression and neural networks for the task of multi-class malware classification

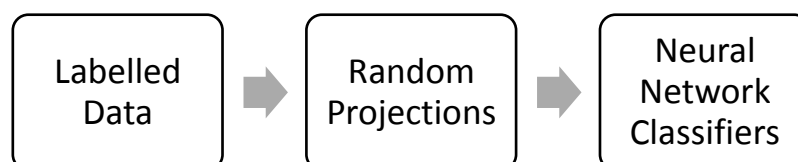


Figure 2.5: System Diagram for Malware Classifiers.

The classification of malware based on behaviour of different malware is done by using machine learning techniques and data mining techniques. At present there are few researches on malware visualization technique. Kyoung Soo Han. *et al* (2015), proposed an algorithm that a new malware family classification by converting binaries into 2 dimensional gray scale images and computing entropy of individual families and similarities also calculated and stored in feature vector in database and entropy graph effectively distinguish a malware families. Kong. *et.al* (2013), proposed a malware samples as malware classification is done on distance learning based for identification of malware structural information.

One of the feature selection techniques in educational data mining is chi-square that used to reduce dimension of data and improve high accuracy (S. A. Kavitha, R. Kavitha, J. Viji Gripsy, n.d.). Chi-square is used for assessing two kind of comparing which are tests of independence and tests of goodness of fit. In feature selection, test of independence is assessed by chi-square and estimate whether the class label is independent of a feature (N. M. Mustafa, L. Chowdhury, M. S. Kamal, 2012). According to S. L. Y. Lam (1999), the Chi-square statistics formula is related to information theoretic feature selection functions which try to capture the intuition that the best terms for the class (c) are the ones distributed most differently in the sets of positive and negative examples of class (c).

2.6.3 Clustering

Clustering can be defined as a division of data into group of similar objects. Each group, or cluster, consists of objects that are similar to one another and dissimilar to objects in other groups (Berkhin P, 2006). Clustering algorithms are able to detect intrusions without prior knowledge. There are various methods to perform clustering that can be applied for the anomaly detection (Shikha and Jitendra, 2015). For example, k-Means clustering and x-means clustering.

K-Means clustering is a cluster analysis method where we define k disjoint clusters on the basis of the feature value of the objects to be grouped. Here, k is the user defined parameter (T.Pang-Ning,M, Steinbach, V.Kumar,2006). There has been a Network Data Mining (NDM) approach which deploys the K-mean clustering algorithm in order to separate time intervals with normal and anomalous traffic in the training dataset. The resulting cluster centroids are then used for fast anomaly detection in monitoring of new data (Munz, G., LI S., Carle G., 2006). The k-means is particularly used to identify outliers because when there is a value that is far away from the majority of the data, the mean value of the cluster will be significantly distorted. This study will use k-means clustering as a method of outlier detection. In this outlier detection model, it is assumed that normal behaviour pattern are far more frequent than the outliers or anomalous behaviours (Solane, Mohd Nizam, 2015).

X-Means clustering is a variation of k-means clustering that refines cluster assignments by repeatedly attempting subdivision, and keeping the best resulting splits, until some criterion is reached (D.Pelleg and A.Moore, 2000).

Table 2.4: Collection of Journals

No	Author & Years	Aim/ Objective	Technique	Parameters	Results
1.	Krzysztof Grabczewski and Norbert Jankowski (2014)	To augment the efficiency using feature selection.	Features of decision tree for separating objects representing different classes.	n - vectors count f - features count	Feature selection method based on the SSV criterion and experimentally confirmed that they are a very good alternative to the most popular methods by 91.2% accuracy.
2.	Osiris Villacampa (2015)	To compare and contrast different feature selection methods.	Correlation-based feature selection and Information gain.	Confusion Matrix; TP, FN, FP, TP.	Increase 92% accuracy results using SVM model with wrapper dataset.
3.	Nooritawati Md Tahir, Aini Hussain et.al (2006)	To select the best features vectors for the PR task.	Use Decision tree as classifier and perform classification using top-down approach	Use Classification and Regression Tree (CART) to predict values of continuous variable.	KG-rule and Scree test can be used as a guide in the optimal feature selection process as the decision tree classifier result is 89.5%.
4.	Ugur PEHLIVAN and Nuray BALTACI (2014)	To select the best features and classification algorithms in permission based.	Random Forest and J48 Decision Tree classification algorithms.	<ul style="list-style-type: none"> • Overall Accuracy (ACC) • True Positive (TP) Rate • False Positive (FP) Rate • Precision 	Cfs Subset Evaluator feature selection method gave a good performance when it was used with 25 features.
5.	Jeevanandam Jotheeswaran and Dr.	To test feature selection for	feature selection for Opinion	TF-IDF which is term frequency and inverse	Feature selection is needed for successful

	Y. S. Kumaraswamy (2013)	Opinion mining using decision tree based feature selection.	mining using decision tree is proposed	document frequency.	data mining applications, as they lower data dimensionality by 15%.
6.	Mohd Zaki Mas'ud, Shahrin Sahib et.al (2014)	To analyse features selection and machine learning classifier.	Feature selection methods using Information Gain and Chi-square.	<ul style="list-style-type: none"> • Overall Accuracy (ACC) • True Positive (TP) Rate • False Positive (FP) Rate • Precision 	Best performance was achieved by the Multilayer perceptron (MLP) classifier using the features selection set derives from the features selection method in 84.5%.
7.	Dharm Singh, Naveen Choudhary and Jully Samota (2013)	To analyse data mining classification with decision tree technique	ID3 classification and cascaded model of RBF network	Adaptive step forward/decision tree (ASF/DT)	Results ID3 classifier with CRBF accuracy is higher than ID3 classifier which is 94.1%.

8.	Chunling Cong and Chris Tsokos (2015)	To precisely study about applications of Decision Tree with statistical software.	Package rpart and tree can be used to construct classification, regression and survival trees in implementation of WEKA.	$(X, Y) = (X_1, X_2, X_3 \dots X_k, Y)$ and the target variable Y can be classified or predicted as necessary.	Decision Tree performs well with large data in a short time which is in below 10 minutes.
9.	M. Dash and H. Liu (2010)	This survey identifies the future research areas in feature selection, introduces newcomers to this field, and paves the way for practitioners	Heuristic generation procedures are subdivided into ‘forward selection’, ‘backward selection’, ‘combined forward/backward’, and ‘instance-based’ categories.	Feature X is preferred to feature Y if the information gain from feature X is greater than that from feature Y	Reducing the 12% number of irrelevant/redundant features drastically reduces the running time of a learning algorithms.

10.	Ajit Kumar, and Pramod Sagar (2017)	To detect malwares through code analysis.	Static analysis and dynamic analysis.	classify the given Android application as benign or malware with the help of GIST feature extracted from each image	Proposed machine learning based light weight Android malware classification technique which uses image features as the code is highly detected in 88.9%.
-----	-------------------------------------	---	---------------------------------------	---	--



2.7 Summary

In conclusion, the types of malware, malware detection techniques and malicious activity have been done and discussed. The Chi Square Statistical and SVM classifier are chosen as it easy to calculate and interpret. The next chapter will discuss about the methodology that will be used in this project. All of the activities involved in the project will be described.

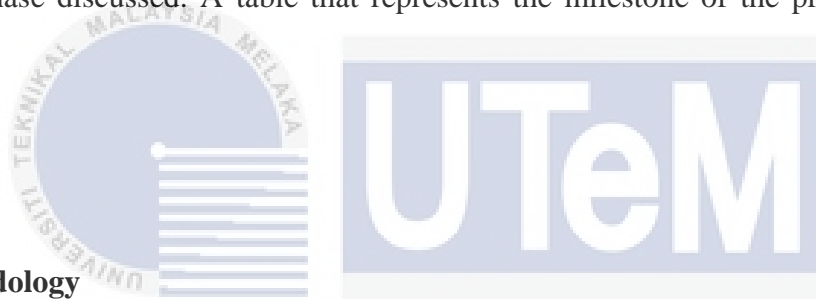


CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, the project methodology will be described and the activities for each phase discussed. A table that represents the milestone of the project has been created.



3.2 Methodology

Methodology is the method or steps used to carry out a project. It will be explained about whole process of the project and the main is the data collection phase and classification algorithm on machine learning. This project consists of seven phases namely literature review phase, data collection phase, data validation phase, classification phase, evaluation phase, and documentation phase as depicted in Figure 3.1.

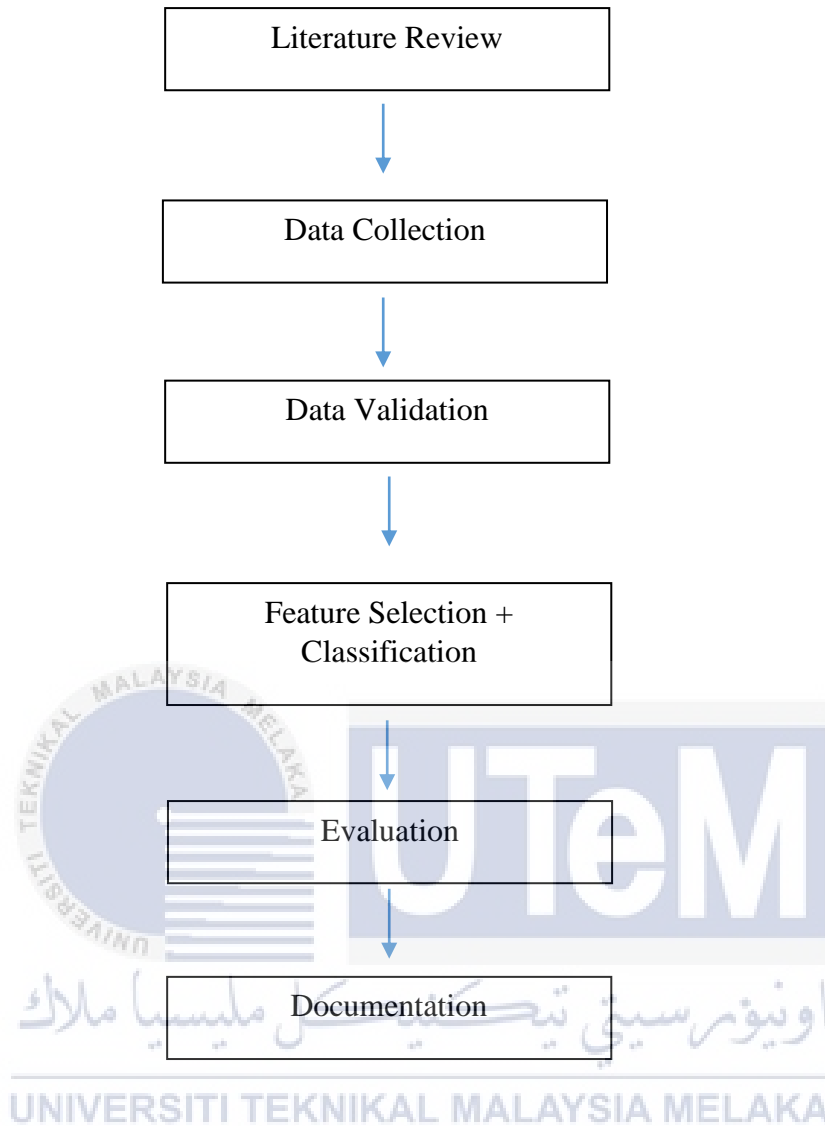


Figure 3.1: Project methodology

Figure 3.1 shows the whole project methodology that consists literature review, data collection, data validation, classification, evaluation and documentation. Each process will be explained latter for better understanding to this project.

3.2.1 Literature Review Phase

In the literature review phase, the activity that is carried out is conducting the related research based on types of malware and detection techniques. It also can help to this project for choose the best algorithm to use in feature selection and classification technique on machine learning. The process start with capturing network traffic of malicious as well as normal apps. The collected journals are used as a reference to help this project significantly.

3.2.2 Data Collection Phase

Data collection is process to gather the data of process from client site to use for machine learning. There were various methods of data collection method or procedure to do that. For this project, the processes that involve on this phrase are data collection, then for data validation this phrase is very important because 160 type of application needed to collect. Then 80 applications that already embedded malware and 80 for clean application. The data was collected from malware a total of 558 APK and extracted to .xml file. The features generator and new dataset containing granted permissions for each application was created.

3.2.3 Data Validation Phase

Data validation process is to determine whether the data is good or not to be used. This process is carried out while collecting data. Validation process has been divided into two parts which is valid or invalid data. After the data has been execute to AndroidManifest.xml file, we developed some tools in Python for each permissions to create the binary vectors that correspond to the granted permissions. The step continues with the decision phase. If the data obtained, the process will take a further step. If the data is not obtained, the process will be start again at installing the .apk files. The last process is training and testing. In this step, two partitions was created

which is 71% for training data while 29% is for testing phase, respectively. Lastly, the machine learning algorithm was trained and verified.

3.2.4 Feature Selection + Classification Phase

Feature selection phase is done during the data pre-processing and empowered the machine learning classifier to identify the malware android application. So, the feature selection should be done while preserving the accuracy. Basically, feature selection is mainly affects the training phase of classification. So, feature selection will perform first to select subset of features and then continue with process the data with the selected features to the learning algorithm. With the selected features, a classifier is induced for the prediction phase. Classification has two stages; the first is the learning procedure where the preparation informational collections are investigated by grouping calculation. The educated model or classifier is displayed as grouping guidelines or examples. The second stage is the utilization of model for characterization, and test informational collections are utilized to appraise the precision of grouping standards. Classification phase is the phase when the classifier algorithm technique are ready to test by using RAPID MINER STUDIO. This software will produce the output for the project. The last result to determine which technique will produce the best performance that depends on output from this process. Classification is a supervised learning guidelines. Forecast and order in information mining are two types of information examination assignment that is utilized to concentrate models portraying information classes or to anticipate future information patterns. Classification process has two stages; the first is the learning procedure where the preparation informational indexes are investigated by characterization calculation. The educated model or classifier is displayed as grouping standards or examples. The second process is the utilization of model for characterization, and test informational indexes are utilized to gauge the exactness of arrangement tenets. For the most part, malware is have a place with a similar family dependably have comparable noxious practices. This prompts their modalities are comparative.

3.2.5 Evaluate Phase

Evaluate phase is the process which is the output will produce the result to achieve the objective of the project. To evaluate the performance of classifier we have several parameters or attribute as references such as time taken, percentage of result and others. Traffic network has been used as the dynamic feature for Android malware detection. Based on the testing, the result will be analysed and discussed. The details of the discussion will be presented in Chapter VI.

3.2.6 Documentation phase

The last phase of this project is the documentation phase. The activities that will be done in this phase are producing final report of the project and presenting the project outcome in the final presentation. The document will conclude the result gained from this project and state the weakness and strength, project contribution, project limitation and suggestions on future works in details.

3.3 Research process

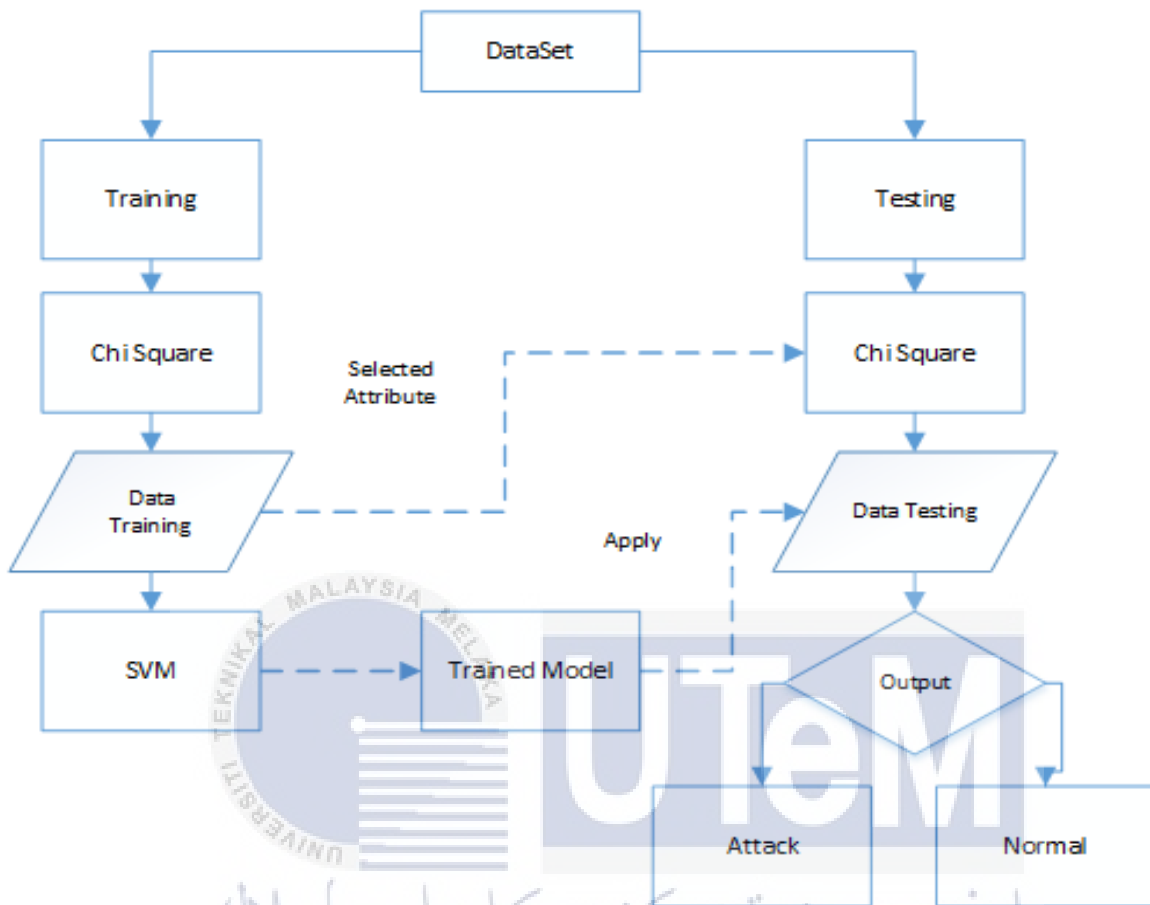


Figure 3.2: Architecture for Android malware Detection

Figure 3.2 shows the whole architecture of the project android malware detection. As we seen this figure consists of components has divided into a few elements that followed by their own function on this project. The diagram also shows the relationship between each component, it also shows the reason they need to communicate each other.

3.4 Software and Hardware Requirement

3.4.1 Software Requirement

There are some of project tools and project requirements are involved in this project. The required software tools that has been used are:

- i. Microsoft Office Word 2013
This software has been used for documentation of the project. For example are proposal and report.
- ii. Microsoft Project 2007
It has been used for project scheduling and management on the project. For example to create a Gantt chart.
- iii. Microsoft Visio
This software has been used to create a better flow chart to make it easier to understand.
- iv. Microsoft Office Excel 2013
Excel has been used to sort the name of the process. It also used to manage data that will be used for classifier as data training set.
- v. Rapid Miner Studio
Software for rapidly building predictive analytic workflows. This all-in-one tool features hundreds of data preparation and machine learning algorithms to support all dataset.
- vi. My SQL
My SQL is an uninhibitedly can be open source Relational Database Management System (RDBMS) that ultimate Structure Query Language (SQL). It is most noted for its fast handling, demonstrated spreadsheet, and straightforwardness and adaptable of utilization.

3.4.2 Hardware Requirement

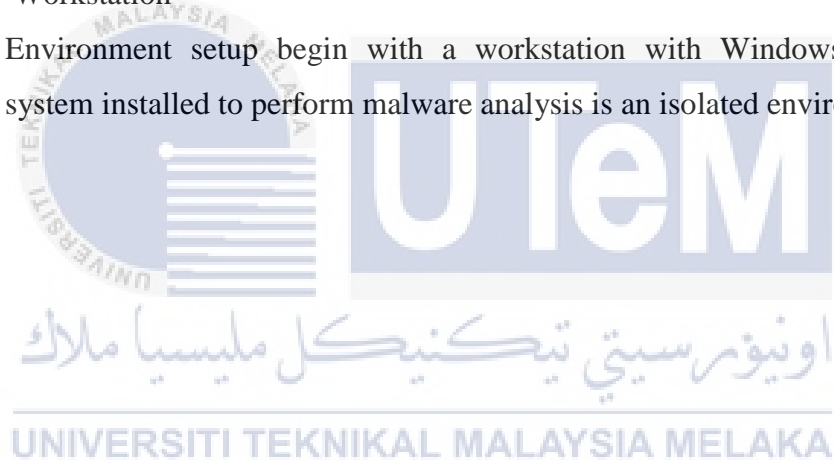
The hardware requirement that needed will be discussed in this section. The required hardware tools that has been used are:

i. Computer

Computers are used in the experiment and detection algorithm development. The computers that are used in this experiment are the workstations where the malware attack is launched. A different computer will be used to develop the algorithm.

ii. Workstation

Environment setup begin with a workstation with Windows 7 operating system installed to perform malware analysis is an isolated environment.



3.5 Project Milestone

Table 3.1: Project milestones

Week	Activity	Note
1 13 – 17 February 2017	Proposal Submission	Topic research.
2 20 – 24 February 2017	Proposal Enhancement	Topic research.
3 27 Feb – 3 Mar 2017	Chapter 1	Topic research.
4 6 – 10 Mar 2017	Chapter 2	Topic research.
5 13 – 17 Mar 2017	Chapter 3	Network environment set up.
6 20 – 24 Mar 2017	Chapter 3	Installation of operating system, virtual machine and android emulator.
7 27 – 31 Mar 2017	Chapter 4	Collecting normal behaviour of the application.
8 1 – 9 April 2017	Mid Semester Break	Research
9 10 – 14 April 2017	Chapter 4	Collecting normal behaviour of the application
10 17 – 21 April 2017	Chapter 4	Collecting normal behaviour of the application
11 24 – 28 April 2017	Chapter 4	Collecting normal behaviour of the application
12 1 – 5 May 2017	Chapter 5	Collecting information of the application after the infection of the malware
13 8 – 12 May 2017	Chapter 5	Collecting information of the application after the

		infection of the malware
14 15 – 19 May 2017	Chapter 5	Collecting information of the application after the infection of the malware
15 22 – 26 May 2017	Chapter 5	Collecting information of the application after the infection of the malware
16 29 May – 2 June 2017	Chapter 5	Collecting information of the application after the infection of the malware
3 – 11 June 2017	Semester Break	Research
12 – 16 June 2017	Chapter 6	Finding the malware signature command and function in the application source code
19 – 23 June 2017	Chapter 6	Finding the malware signature command and function in the application source code
26 – 30 June 2017	Chapter 6	Finding the malware signature command and function in the application source code
3 – 7 July 2017	Chapter 7	Finding the malware signature command and function in the application source code
10 – 14 July 2017	Chapter 7	Finding the malware signature command and function in the application source code
17 – 21 July 2017	Chapter 7	Finding the malware signature command and

		function in the application source code
24 – 28 July 2017	Documenting Result	Documenting the project findings
31 July – 4 August 2017	Documenting Result	Documenting the project findings
7 – 11 August 2017	Documenting Result	Documenting the project findings
14 – 18 August 2017	Final Presentation	Presenting the project result to the supervisor and evaluator



3.6 Summary

In conclusion, this chapter highlighted the methodology that are used in order to complete the project. Each phase involved in the methodology is described and the milestones of the project is presented. The next chapter will discuss the design of the experiment and analysis to be carried out in this project.

CHAPTER 4

DESIGN

4.1 Introduction

In this chapter, design and implementation of this project discussed. The requirement needed in this chapter is the design part whereby the flowchart of the sequence of the data flow process in which the behavioural data is captured from the start process to the end process using appropriate tools. Lastly, the implementation of this research project which display a few samples of expected outputs will be received and discussed in this chapter as well.

4.2 Malware Analysis Progress

In this section, design part of the project discussed which covered up both experimental and analysis design.

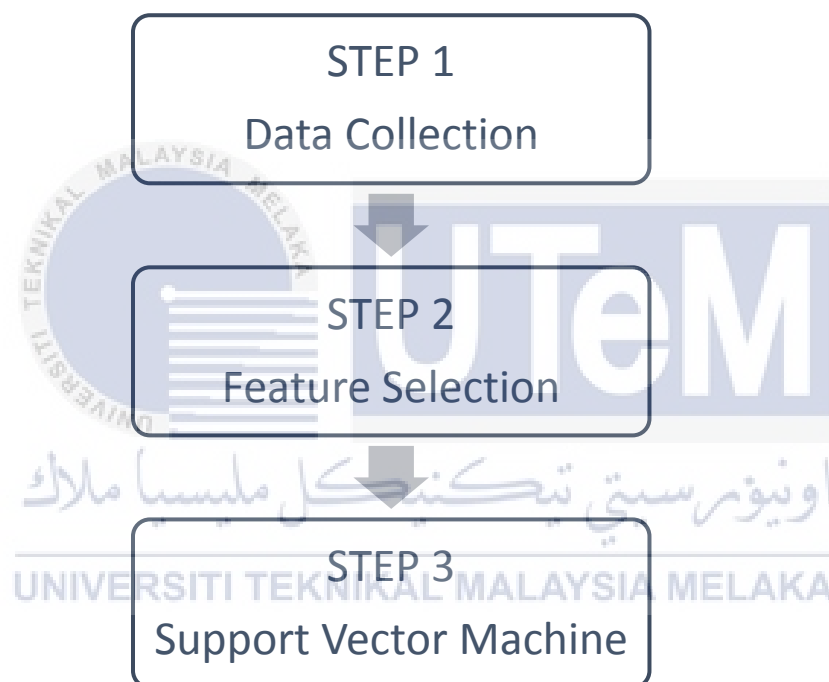


Figure 4.1: Android Malware Analysis Progress.

Figure 4.1 shows the progress of android malware analysis. The flow starts off with data collection. The collected data will be analysed thoroughly. Next, it followed by feature selection and Support Vector Machine.

4.2.1 Data Collection

Data collection is the process of gathering and measuring information on variables of interest, in an established systematic fashion that enables one to answer research questions, test hypotheses, and evaluate output. The data collection component of research is common to all fields of study. While methods vary by discipline, the emphasis on ensuring accurate and honest collection remains the same. In others process it can be done by personal interviewing, mail and internet. For this project data collection is process to gather the data of process from the client site to use for machine learning. There were various methods of data collection method or procedure to do that.

For this project, the processes that involve on this phrase are data collection, then for data validation this phrase is very important because 160 type of application needed to collect. Then 80 applications that already embedded malware and 80 for clean application. There are several reason and method to determine either the data is acceptable or not to use. The example of data cannot be used for machine learning because during data collection there are steps that cannot be done because of the problem with application or device.

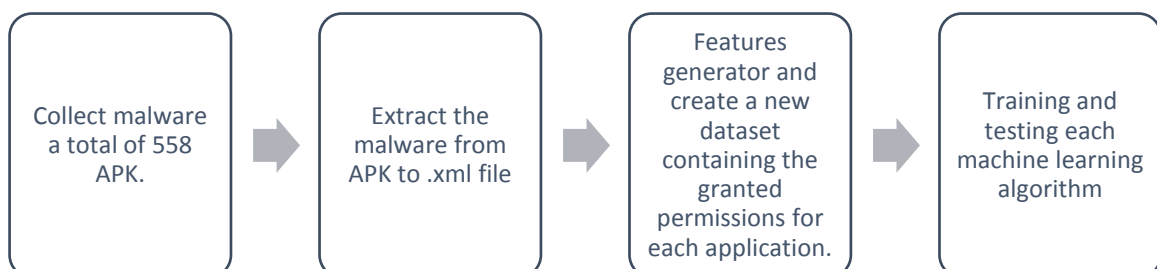


Figure 4.2: Data Collection Procedure

A	B	C	D	E	F	G	H	I	J	K	L
android	android.a	android.ir	android.o	android.p	android.p	android.p	android.p	android.p	android.p	android.p	android.p
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.3: Data Collection



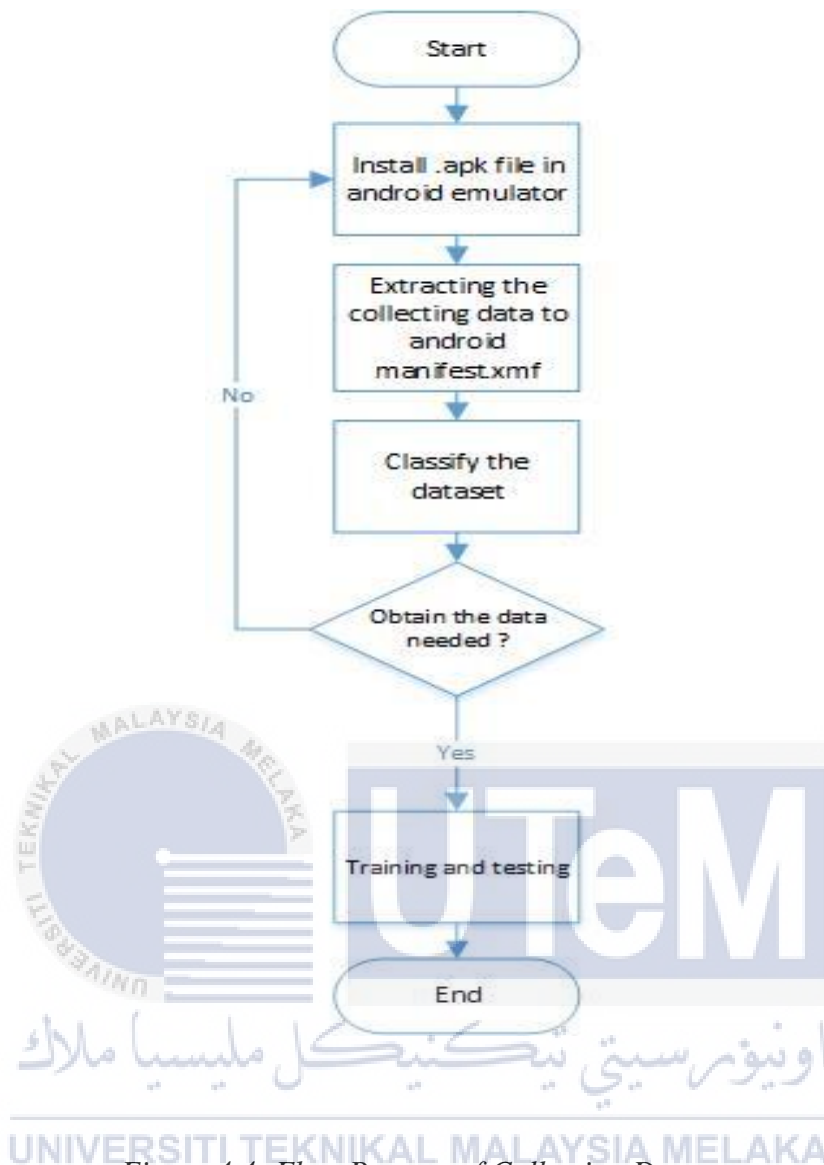


Figure 4.4: Flow Process of Collecting Data.

Based on the figure 4.6, the flowchart explain about the algorithm to make better understand on data collection phase. The first step of the process is installing the .apk file that contains malware in the android emulator. The data needed to be collected was mentioned earlier which are a total of 558 APK. The collection is consisted of 279 applications with low privileges of the free access dataset and a random selection of 279 malwares of the MalGenome. Next, we extract the .apk files to AndroidManifest.xml by using the reverse engineering tools which is ApkTool 2.0.3 in Kali Linux. After we get the AndroidManifest.xml file, we developed some tools in Python for each permissions to create the binary vectors that correspond to the granted permissions. The step continues with the decision phase. If the data obtained,

the process will take a further step. If the data is not obtained, the process will be start again at installing the .apk files. The last process is training and testing. In this step, two partitions was created which is 71% for training data while 29% is for testing phase, respectively. Lastly, the machine learning algorithm was trained and verified.

4.2.2 Feature Selection

There are many ways or solution regarding the detection methods. One of them is by combining automatic dynamic behaviour malware analysis with data machine learning classification technique. Feature selection phase is done during the data pre-processing and empowered the machine learning classifier to identify the malware android application become more productive. So, the feature selection should be done while preserving the accuracy. Feature selection can be categorized into embedded models, wrapper models and filter models. Basically, feature selection methods has four basic steps which is subset generation, subset evaluation, stopping criterion and result validation. The first step is chose a candidate feature subset based on a given search strategy which is sent. Next, the sent will be evaluated according to certain evaluation criteria. The subset that will be chosen from all the candidates are the best that fits the evaluation criteria after the stopping criteria are met. Lastly, the chosen subset will be validated using validation set or domain knowledge.

As for classification problems, it is difficult to get good classifiers before removing the unwanted features due to the huge size of data. If we reduce the number of redundant features, the running time of the learning algorithms can drastically reduce. This lead to getting a better insight into basic concept of classification problem. Usually, feature selection for classification android malware select minimally sized subset of features based on the following criteria:

- Classification accuracy does not significantly decrease.
- Given only the values for the selected features as close as possible to the original distribution resulting the class distribution

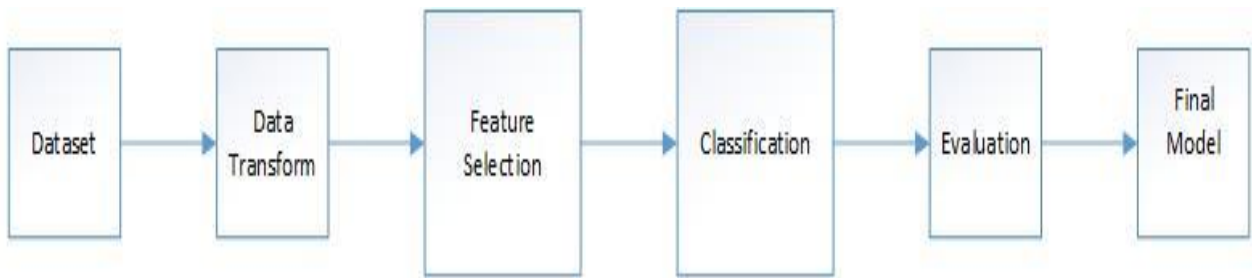
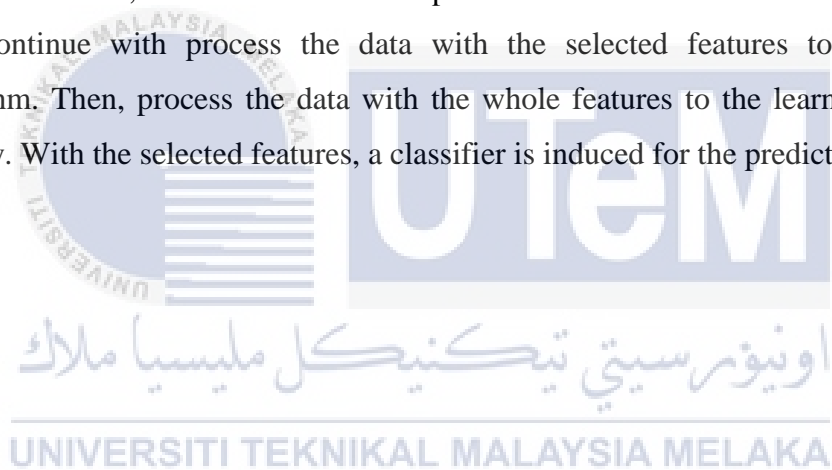


Figure 4.5: A General Framework of Feature Selection of Classification

The figure 4.5 above shows a general of feature selection for classification framework. Basically, feature selection is mainly affects the training phase of classification. So, feature selection will perform first to select subset of features and then continue with process the data with the selected features to the learning algorithm. Then, process the data with the whole features to the learning algorithm directly. With the selected features, a classifier is induced for the prediction phase.



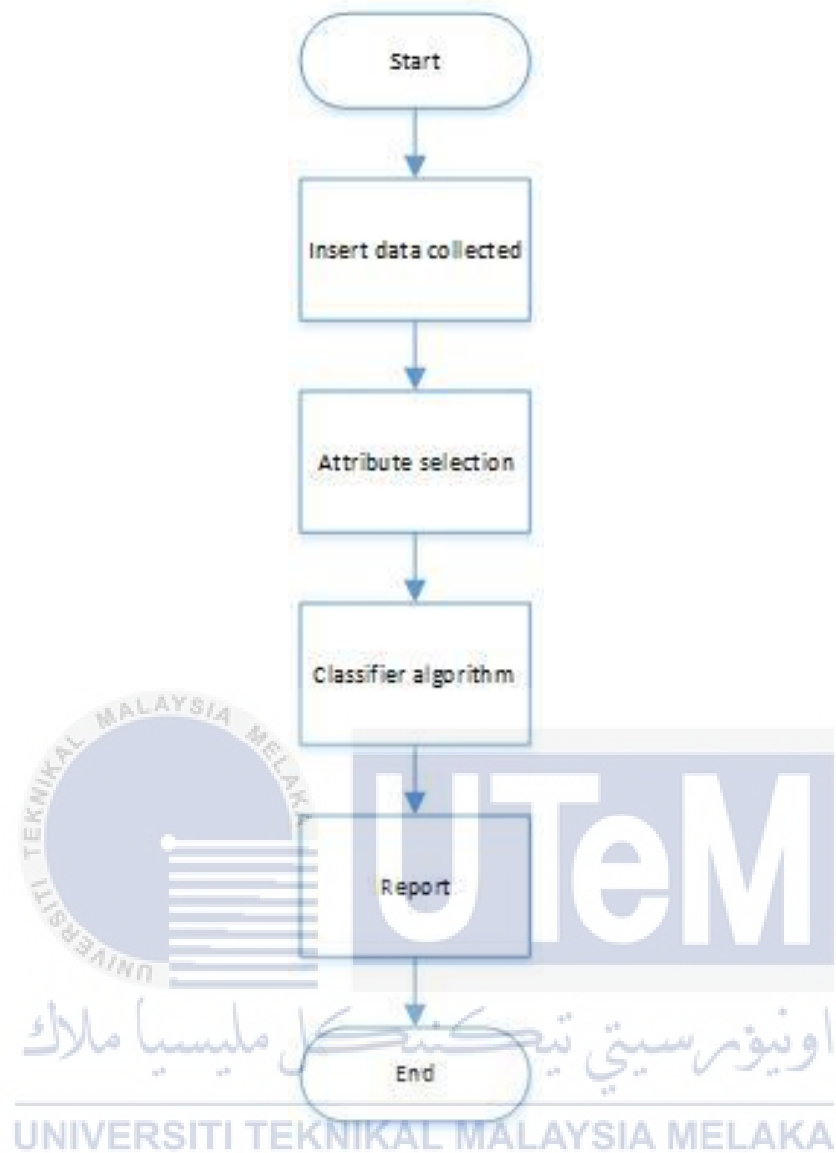


Figure 4.6: Flow Process of Feature Selection.

Based on figure 4.6, the flowchart above shows the steps or flow for the feature selection phase. The process start with insert the previous collected data. Next steps is the process of selecting the features to be used in machine learning classifier. Below are the performance metrics.

$R_1=1$ which is the analyzer detect a granted permission

$R_1=0$ which is any other case

$C_{-1}=1$ which is if the application is malware


$C_{-1}=0$ which is in any other case

$$TPR = \frac{TP}{TP + FP}$$

$$FPR = \frac{FP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Equation 1: Performance of Feature Selection



Where,

- TP is true positive items correctly classified as positive.
- FP is negative items classified as positive.
- FN is positives items classified as negative.
- TN is negative items correctly identified as negatives.

The performance that we test is based on a few aspects which are TPR, FPR, and accuracy.

4.2.2.1 Chi-Square

Feature selection has three methods which is Filter Methods, Wrapper Methods and Embedded Methods. In this project, we apply Filter Methods which is the Chi-Square test. This model is faster than the other approach and the results are also better generalization because of the act is independently. Chi-Square test is done by measuring their chi-squared statistic with classes and the X^2 method evaluates the features individually. In the researches (Imam, I. F., Michalski,R), firstly the moving averages are calculated based on the frequency of event occurrence in the audit. Secondly these researches indicate to detect attacks calculating the X^2 values as the expected values. For this project, the formula can be stated as:

$$(X)^2 = \sum_{i=1}^n \sum_{j=1}^q \frac{(n_{ij} - E_{ij})^2}{E_{ij}}$$

Equation 2: Formula of Chi-Square

Where:

- n is the number of feature while q is the number of classes.
- E_{ij} represents an expected number of instances of j^{th} class and i^{th} value.
- n_{ij} represents the found number of instances of j^{th} class and i^{th} value.

4.2.3 Support Vector Machine

Support Vector Machine or SVM are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. SVM is one of the machine learning classifiers receiving the most attention currently, and its various applications are being introduced because of its high performance. The SVM could also solve the problem of classifying nonlinear data. Of the input features, unnecessary ones are removed by the SVM machine learning classifier itself and the modelling is carried out, so there is some overhead in the aspect of time. However, it could be expected to perform better than other machine learning classifiers in the aspect of complexity or accuracy in analysis.

The process method are shown below:

1. Sampling of data from sampling technique
2. Split data into two parts training and testing part
3. Read directory
4. A = Read syscall from line;
5. Syscall_array [A]++;
6. }
7. Out = output file;
8. While (syscall_array as syscall_freq){
9. Syscall_freq > out;
10. }
11. Print syscall_freq > train.csv;
12. }

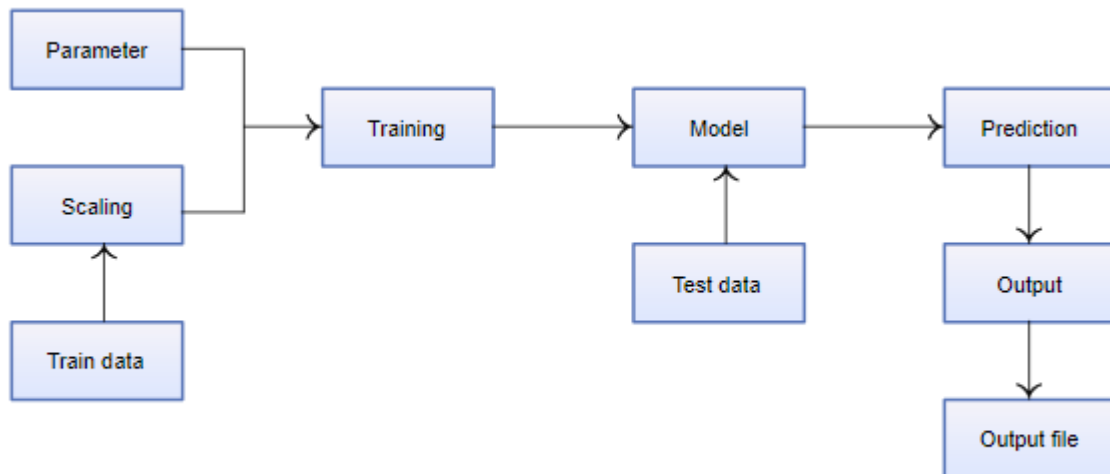


Figure 4.7: SVM Process Method.

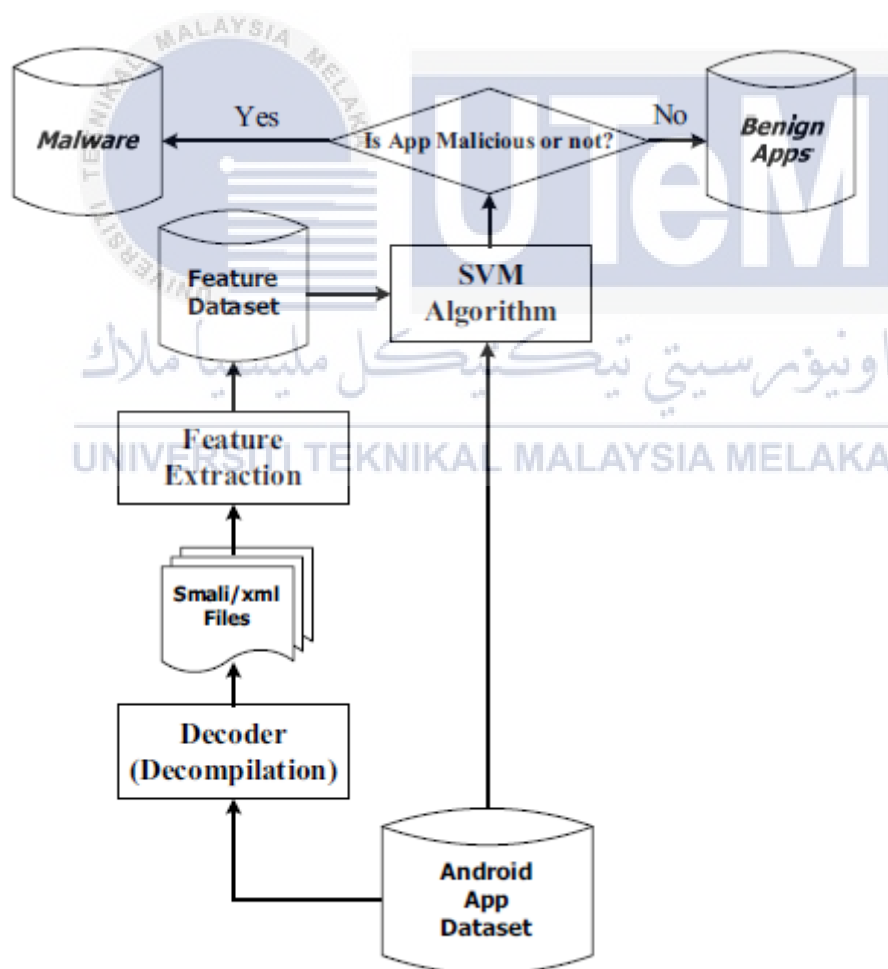


Figure 4.8: Flow Process of SVM

As shown in figure 4.8, this is the flow process of SVM learning algorithm. There are three major components in the malware detection scheme, namely decoder (recompilation), feature extraction, and classifier. In the recompilation component, each Android app is unpacked and decoded into a readable small file. Some key features, such as risky permissions, suspicious API calls, and URLs are then extracted in feature extraction components according to several important and widely accepted measures, such as TF-IDF and cosine similarity. Finally, we use machine learning algorithm to build a classification model and evaluate them on the Android app dataset by classifying them into malware or benign apps.

4.3 Method FS_SVM

Method FS_SVM is a combination from a method Feature Selection and Support Vector Machine. SVM is a supervised learning model with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

Theory SVM has the main parts which is:

- First, we propose a feature extraction method based on keywords correlation distance which is different from the traditional method based on binary program.
- Second, we use feature vector to describe malicious software feature including not only permission, APIs, but also the common parameters and common package etc.
- Third, we give a malware detection method through SVM based on the feature vector set, which can detect new malwares and malicious software variants.

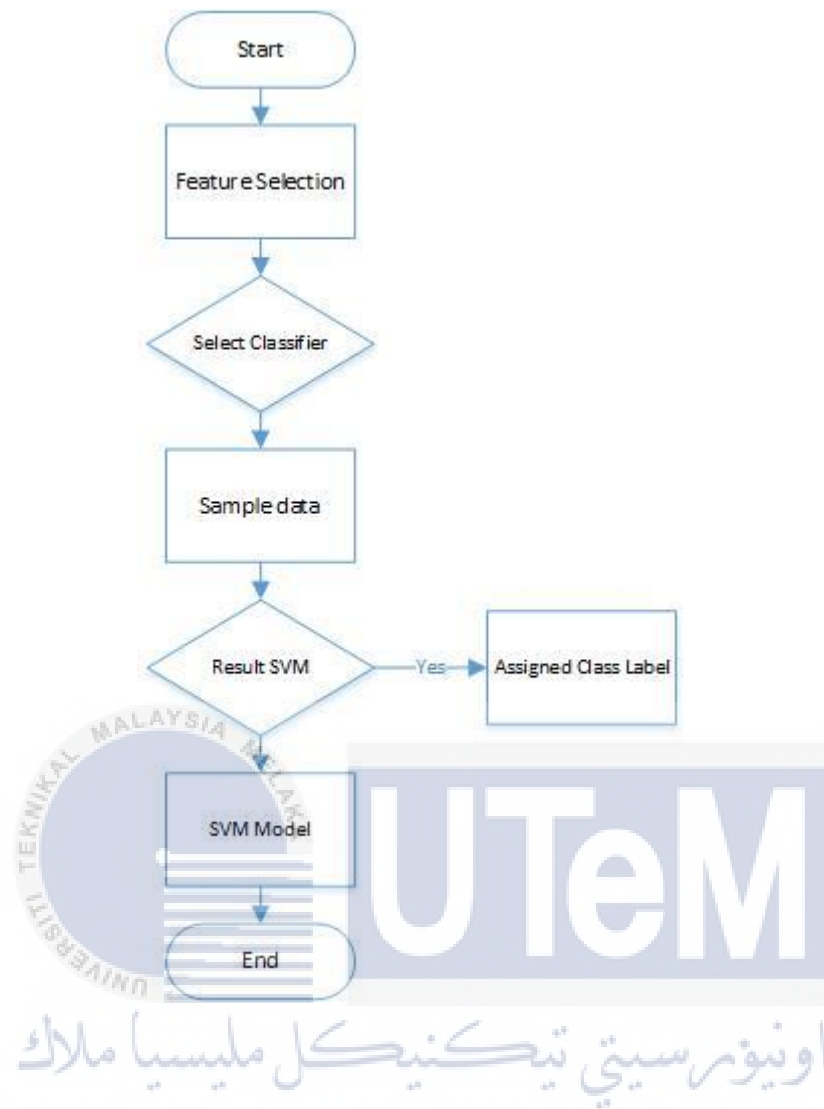


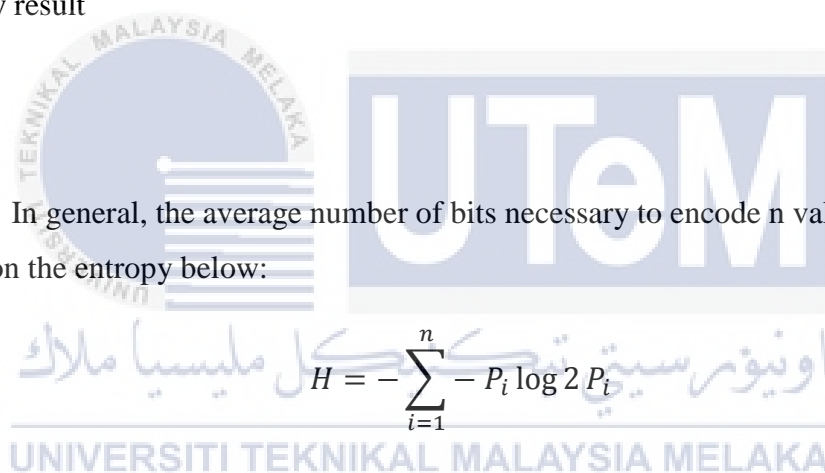
Figure 4.9: Flow Process of FS_SVM

Based on figure 4.9 above, there are total of 558 APK number of data which 279 malware are random selection while the other 279 malware are MalGenome. Inductive machine learning algorithm are used to learn the decision tree function stored in data in form of $(X, Y) = (X_1, X_2, X_3 \dots X_k, Y)$. The target about the variable Y can be classified or predicted.

4.3.1 Pseudo code FS_SVM

- 1.0 Start
- 2.0 Insert data collection that has been extract
- 3.0 Use Chi-square feature selection to produce results.
 - 3.1 Measuring chi-squared statistic with classes
 - 3.2 Evaluates the X^2 method features individually
- 4.0 Read directory
 - 4.1 while (file in directory){
 - 4.2 Read file;
 - 4.3 Vector = Concatenate (vector, line, “ “)
 - 4.4 Print Vector
- 5.0 Display result
- 6.0 End

In general, the average number of bits necessary to encode n values is the based on the entropy below:


$$H = - \sum_{i=1}^n - P_i \log_2 P_i$$

Equation 3: Formula to Encode n Values

- P_i = probability of occurrence of value i:
 - High entropy: All the classes are nearly equally like
 - Low entropy: A few classes are likely most of the classes are rarely observed.

4.5 Summary

This chapter has provided a through explanation on few topics which are the framework, flow chart and the method for feature selection and decision tree. Besides, it also show data collection that has been implementing during this project. This chapter also show overview about the process and the procedure need to follow for data collection phase. In the next chapter, implementation task will be covered which is testing and result analysis.



CHAPTER 5

IMPLEMENTATION

5.1 Introduction

In this chapter, implementation of the project discussed. In the previous chapter, the Design had been discussed and Implementation had been done based on the previous chapter.

5.2 Environment Set Up

Based on the network design that had mentioned in Chapter3, the actual environment will be set up in Figure 5.1.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

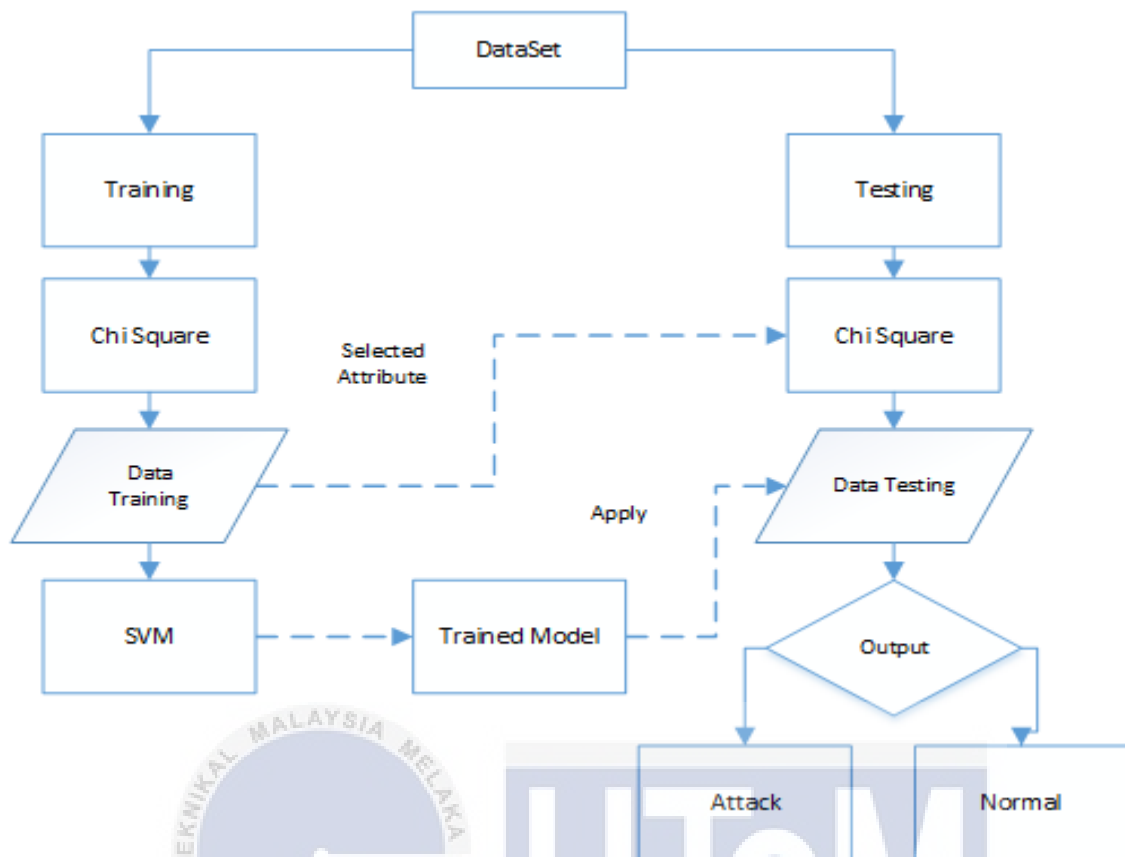


Figure 5.1: Detailed Framework of Implementation

The main idea of operation of this analysis can be divided into two main steps as illustrated in figure 5.1. Training phase creates normal profile behaviours packet data and calculate packet score while testing phase test normal packet with testing dataset. We started with insert dataset which has one special attribute and 330 regular attribute into Rapid Miner Studio. Based on the dataset used, there are 160 attributes, 80 are malware and 80 are clean.

ExampleSet (398 examples, 1 special attribute, 330 regular attributes) Filter (398 / 398 examples): all

Row No.	type	android	android.app...	android.inten...	android.os.ct...	android.per...	android.per...	android.per...	androi
1	1	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0	0
9	1	0	0	0	0	0	0	0	0
10	1	0	0	0	0	0	0	0	0
11	1	0	0	0	0	0	0	0	0
12	1	0	0	0	0	0	0	0	0
13	1	0	0	0	0	0	0	0	0
14	1	0	0	0	0	0	0	0	0
15	1	0	0	0	0	0	0	0	0
16	1	0	0	0	0	0	0	0	0
17	1	0	0	0	0	0	0	0	0
18	1	0	0	0	0	0	0	0	0

Figure 5.2: Dataset used

Based on Figure 5.2, the dataset used has 160 fields such as android, android.app.cts.permission.TEST_GRANTED, android.permission.ACCESS_ALL_DOWNLOADS, android.permission.ACCESS_ALL_EXTERNAL_STORAGE, android.permission.ACCESS_BLUETOOTH_SHARE, android.permission.ACCESS_CACHE_FILESYSTEM, android.permission.ACCESS_DOWNLOAD_MANAGER, and many more.

These dataset are in CSV file format then manipulated through Rapid Miner which also works as database storage as well to generate the Train and Test Dataset.

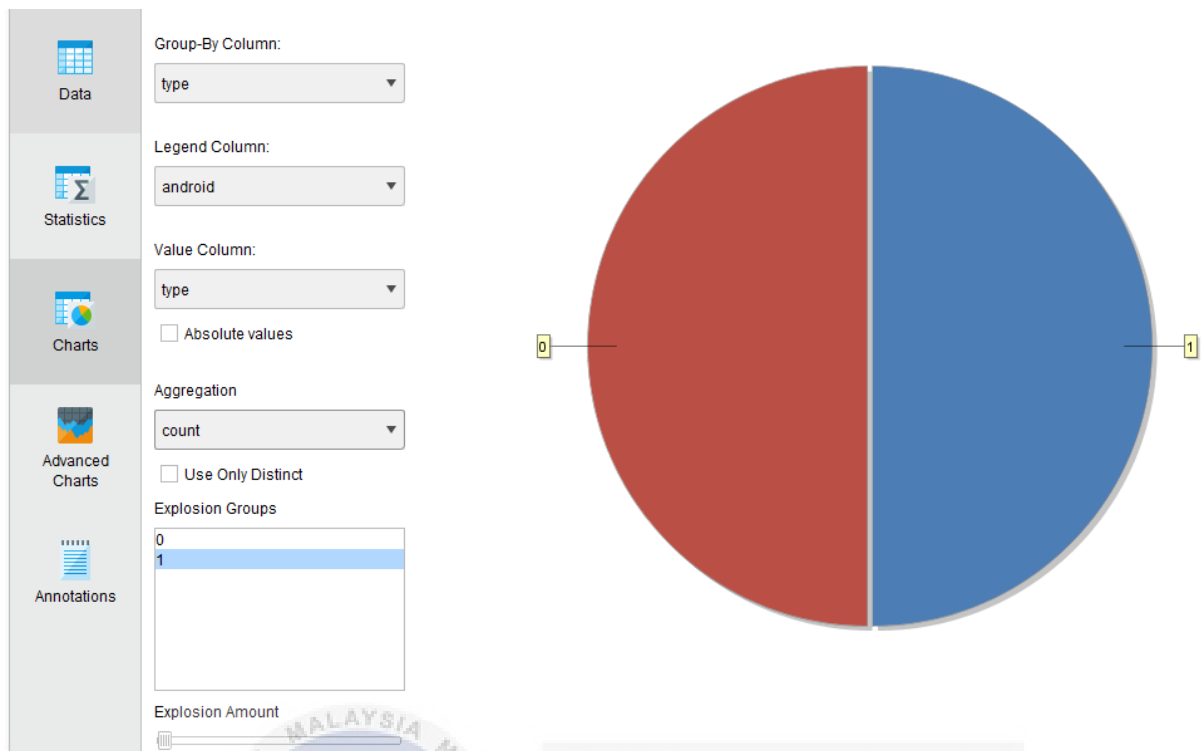


Figure 5.3: Chart of Dataset.

Based on figure 5.3, the Pie Chart can be created based on the dataset used. We can set the parameters to produce the type of charts that we want. The Pie is built based on the type of android which has value 1 and 0.

5.3 Design Process

In this section, the step of design will be explained. The process of design start with a database created before we start the design. The training and testing dataset which already has unique features are used to interpret dataset which involves in training and testing phase.

5.3.1 Add Data

The step of add data will be explained in this section.

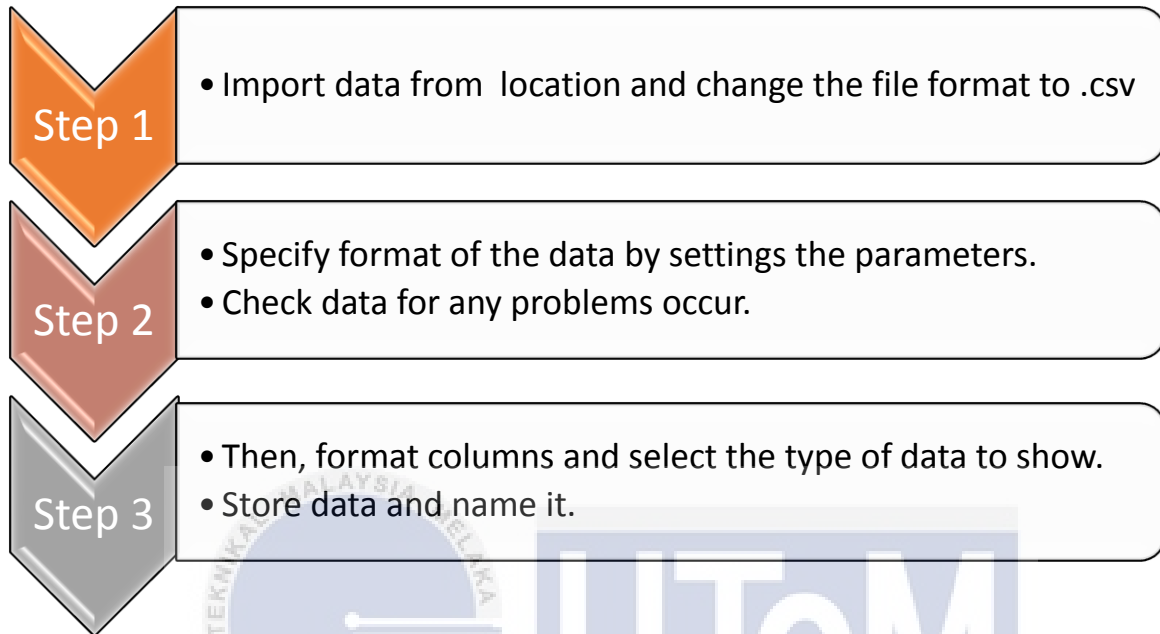


Figure 5.4: Step of Add Data

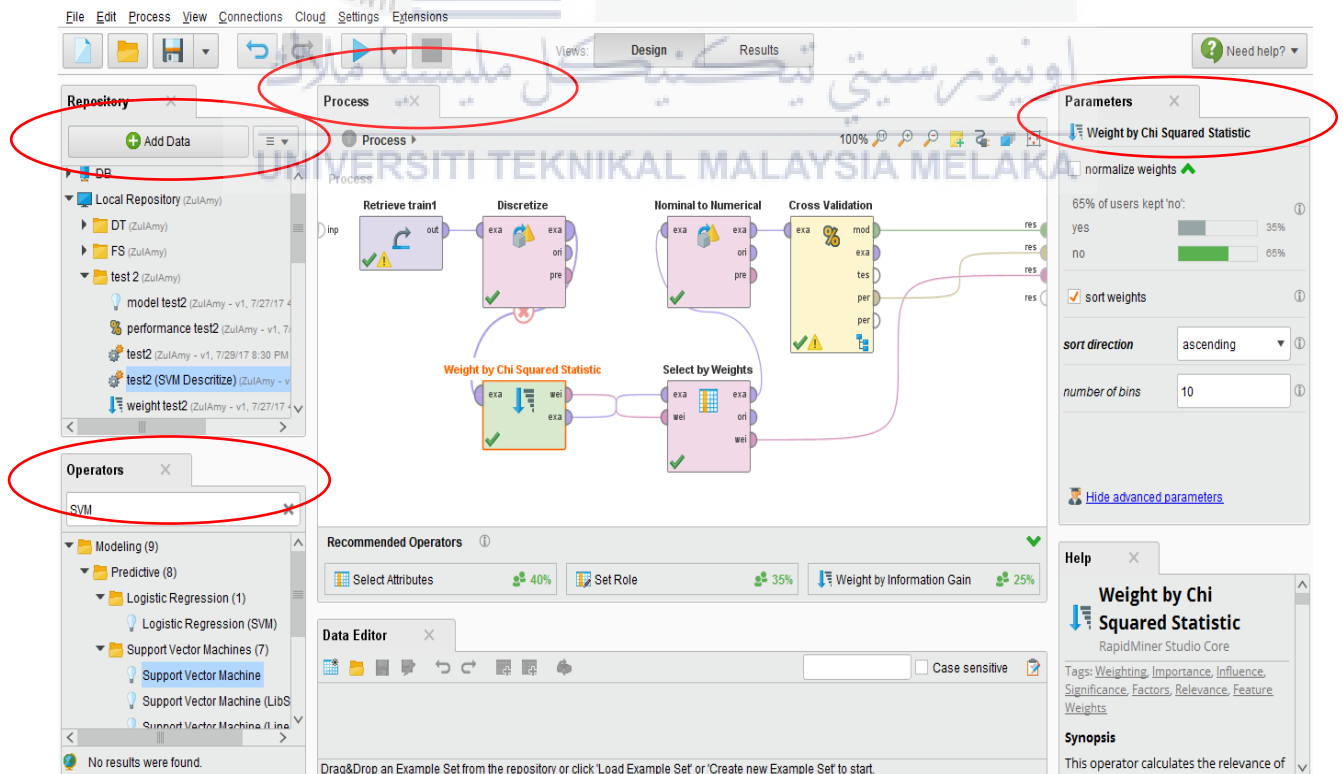


Figure 5.5: Interface of Rapid Miner Studio.

Based on Figure 5.5, it shows the interface of Rapid Miner Studio. As we can see, on the top left corner is the Add Data section. From there, we can add dataset or any other file to store in Rapid Miner. Below of that are operators section where can search for the operators that want to use. The centre of interface is process part. This part is for design the process. It can just simply drag the operator to destination. On the right corner are the parameters of the operator. Once click the operator and can set parameters.

5.3.2 Basic SVM Process

The step of designing the basic process of SVM will be explained in this section.

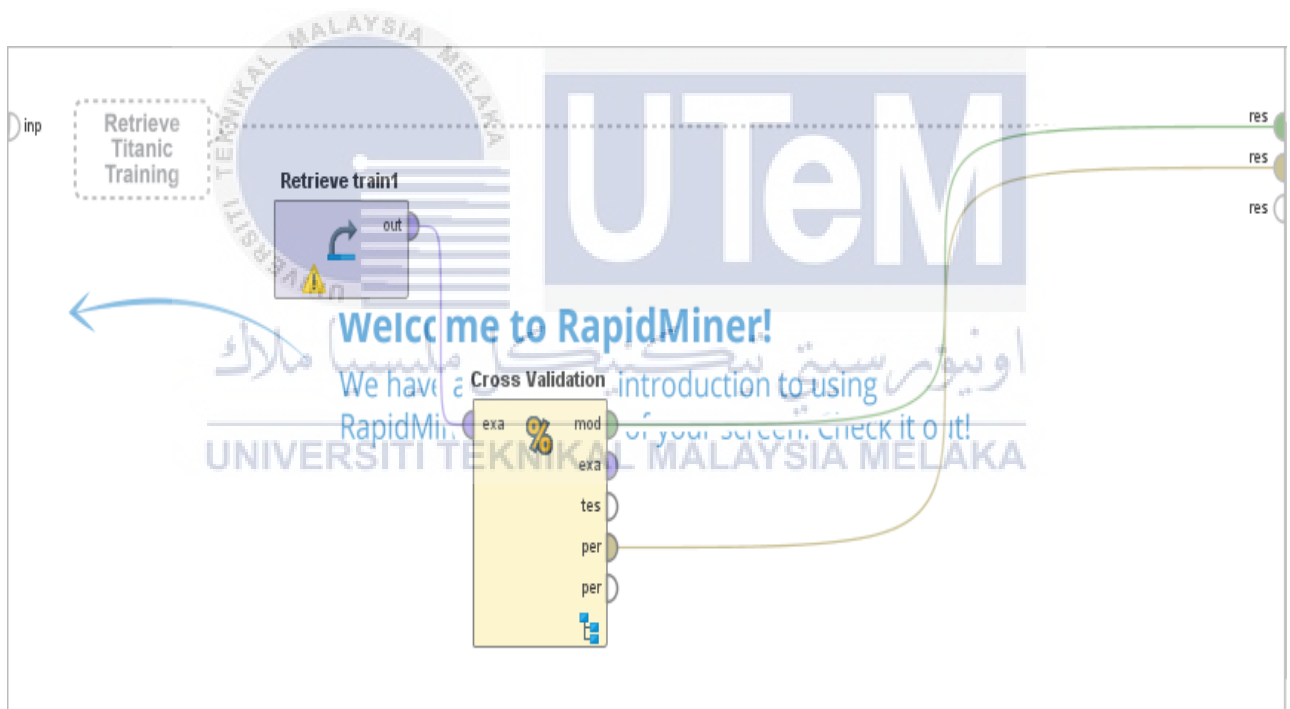


Figure 5.6: Basic Design Process

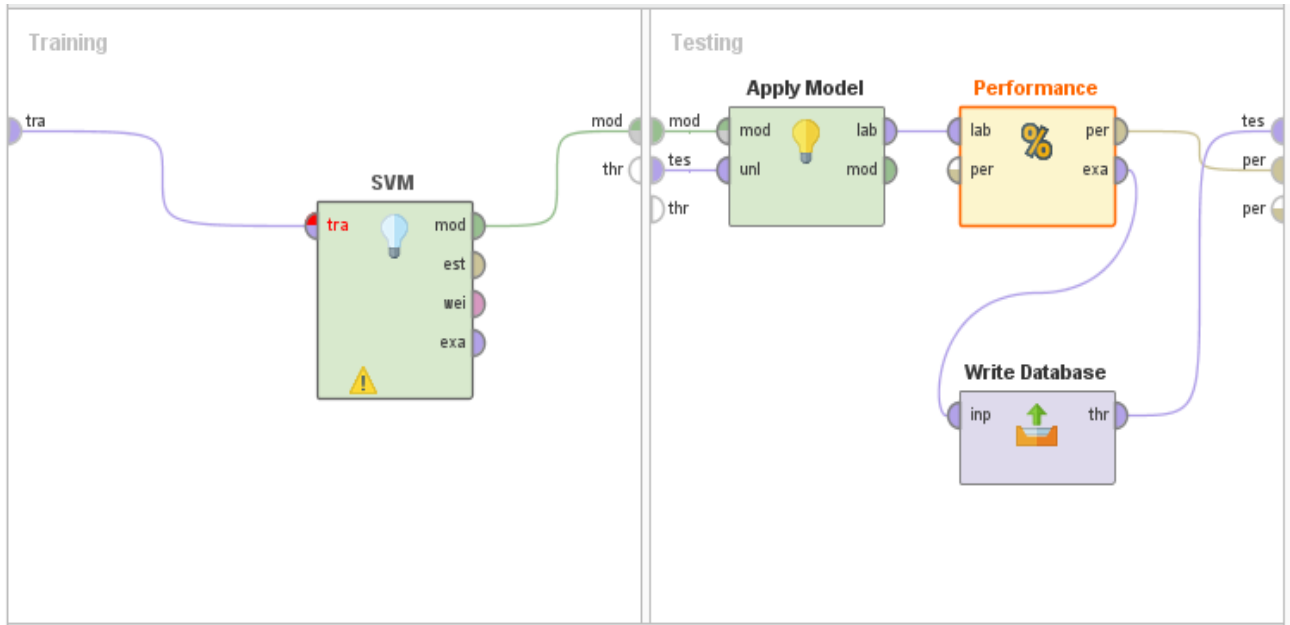


Figure 5.7: Design Process of Training and Testing Phase.

Based on Figure 5.6 and 5.7, there are basic design of process. After we add data, we need to connect it to Cross Validation. The Cross Validation operator is a nested operator. It has two sub processes which is a training sub process and a testing sub process. The training sub process is used for training a model. The trained model is then applied in the testing sub process. The performance of the model is also measured during the testing phase. This operator performs a cross-validation in order to estimate the statistical performance of a learning operator. Instead, we can set the parameters of the operator.

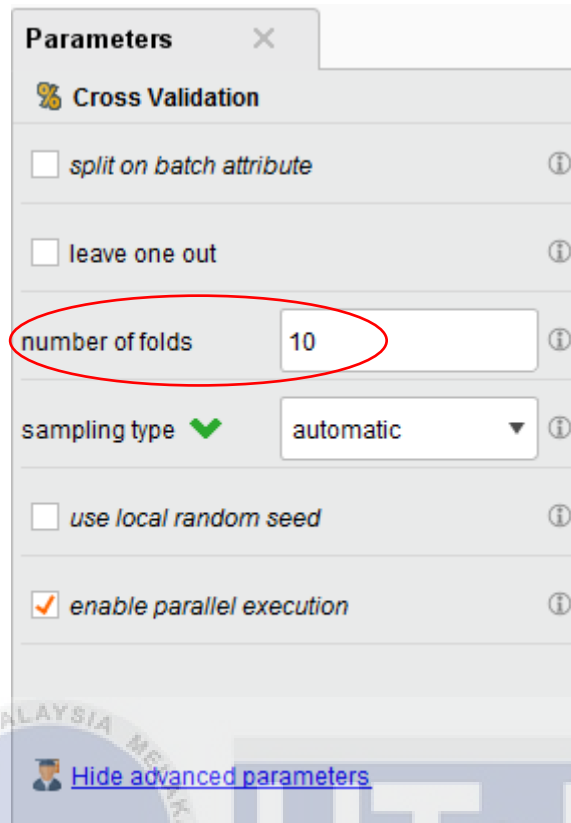


Figure 5.8: Parameters of Cross Validation

Figure 5.8 shows the parameter of the Cross Validation operator. We set the number of folds to 10 as it specifies the number of folds (subsets). The example set should be divided into each subset that has equal number of examples as the same number of iterations takes place. Each iteration involves training a model and testing that model.

In training phase, we put the classifier that we want to test which is SVM. The standard SVM takes a set of input data and predicts, for each given input, which of the two possible classes comprises the input, making the SVM a non-probabilistic binary linear classifier. While in testing phase, we placed Apply model and Performance operator. The purpose for this operator is to produce the model of the classifier and the performance or result of the process.

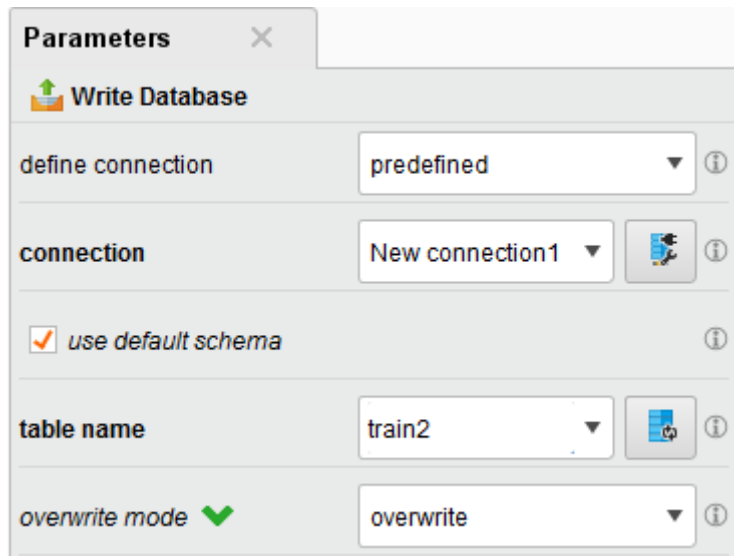


Figure 5.9: Parameters of Write Database

Figure 5.9 shows the parameters of Write Database operator. This operator writes an ExampleSet to an SQL database. The connection must be defined to make sure the connection is valid. For the table, write the table name as created in SQL database otherwise it will create itself. This parameters help to store results from rapid miner to database.

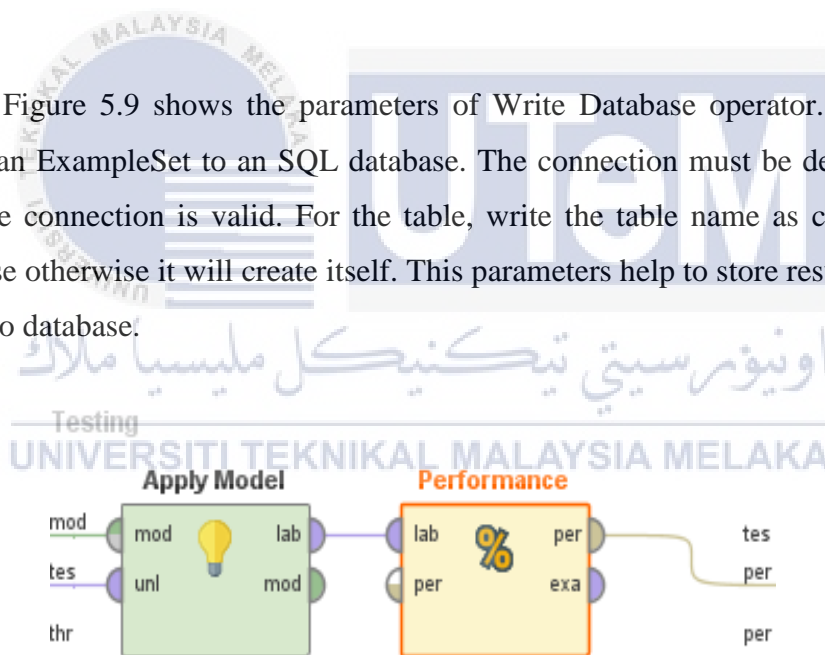


Figure 5.10: Apply Model and Performance Operator.

Figure 5.10 shows the operator of apply model and performance. Apply model operator applies an already learnt or trained model on an ExampleSet. It also help to build model of classifier algorithm which is SVM. While performance operator used for statistical performance evaluation of classification tasks. This operator delivers a list of performance criteria values of the classification task.

5.4 Connect to MYSQL

In this part, we connect the rapid miner to MySQL workbench. The results from rapid miner can be stored in database to calculate the accuracy, detection rate and false alarm.

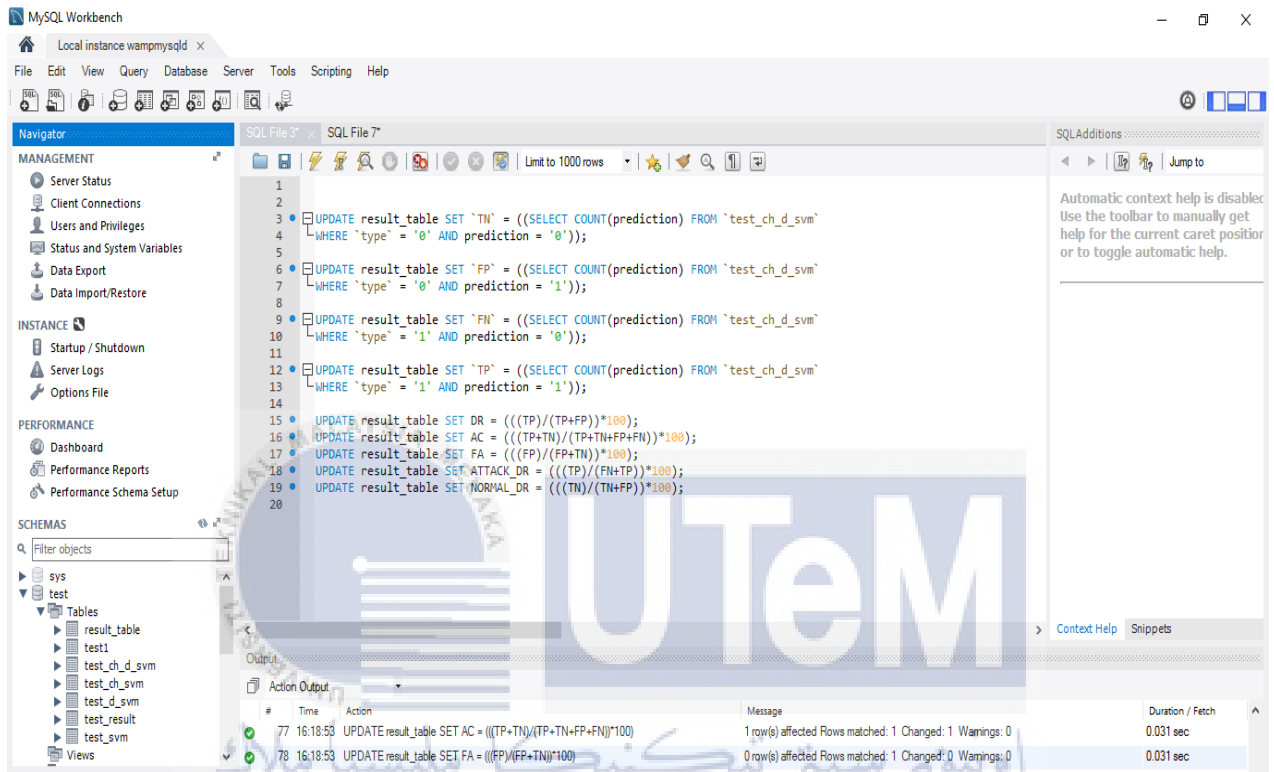


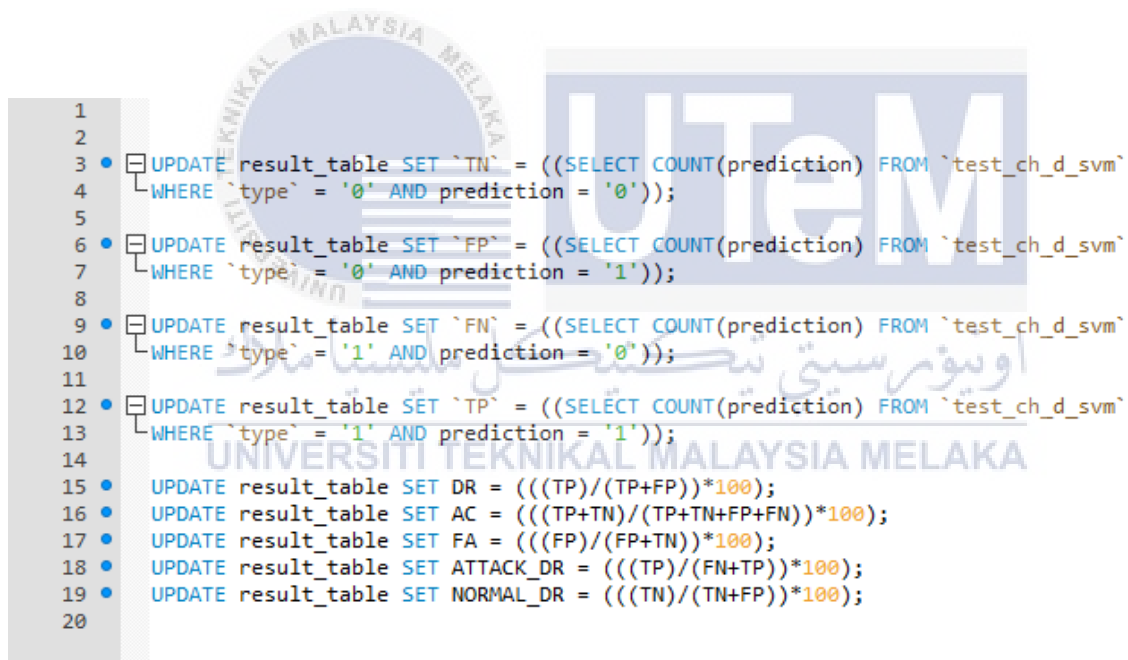
Figure 5.11: Interface of MySQL Workbench

The statistical analysis begins with a database named as *psm2* is created during the installation of MySQL Workbench. Databank dataset which already has unique features, are used to interpret the dataset which involves in training and testing phase. In MySQL, we go to the Database tab and choose Query Database to get into our database that we already created.

There are 6 tables created in the database as following:

- *result_table*
- *test1*
- *test_svm*
- *test_ch_svm*
- *test_d_svm*
- *test_ch_d_svm*

After the table was created, then the rapid miner and MySQL are connected by using Write Database operator in rapid miner. Then results are stored in the scheme created.



```
1
2
3 • UPDATE result_table SET `TN` = ((SELECT COUNT(prediction) FROM `test_ch_d_svm`
4   WHERE `type` = '0' AND prediction = '0'));
5
6 • UPDATE result_table SET `FP` = ((SELECT COUNT(prediction) FROM `test_ch_d_svm`
7   WHERE `type` = '0' AND prediction = '1'));
8
9 • UPDATE result_table SET `FN` = ((SELECT COUNT(prediction) FROM `test_ch_d_svm`
10  WHERE `type` = '1' AND prediction = '0'));
11
12 • UPDATE result_table SET `TP` = ((SELECT COUNT(prediction) FROM `test_ch_d_svm`
13   WHERE `type` = '1' AND prediction = '1'));
14
15 • UPDATE result_table SET DR = (((TP)/(TP+FP))*100);
16 • UPDATE result_table SET AC = (((TP+TN)/(TP+TN+FP+FN))*100);
17 • UPDATE result_table SET FA = (((FP)/(FP+TN))*100);
18 • UPDATE result_table SET ATTACK_DR = (((TP)/(FN+TP))*100);
19 • UPDATE result_table SET NORMAL_DR = (((TN)/(TN+FP))*100);
20
```

Figure 5.12: SQL Query Calculations of Results

Figure 5.12 shows the code of calculation of the results. The calculation of True Negative, False Positive, False Negative, True Positive, Detection Rate, Accuracy, False Alarm, Attack Detection Rate and Normal Detection Rate are as below.

5.5 Conclusions

Generally, this chapter discussed on experimental set up and design process of Add Data and the overall process to conduct the statistical analysis tests. The result on this project will be discussed in the next chapter which is Chapter 6.



CHAPTER 6

RESULT AND ANALYSIS

6.1 Introduction

In this chapter, the findings and results of the project will be discussed. In the previous chapter, the Implementation had been discussed based on the previous chapter.

6.2 Results of Suggested Approaches

The tests are carried out managed to evaluate and validate the proposed approach based on drawback posed in previous work in terms of accuracy, detection rate.

6.2.1 Basic SVM Process

This section will show the result of the basic process of Support Vector Machine model.

accuracy: 80.62% +/- 8.59% (mikro: 80.62%)

	true 0	true 1	class precision
pred. 0	79	30	72.48%
pred. 1	1	50	98.04%
class recall	98.75%	62.50%	

Figure 6.1: Results of testing SVM Basic Process

Based on Figure 6.1, it shows the results of the basic SVM Process. The accuracy rate of detection is 80.62%. The True Positive Rate for Class 1 (malware) is 69.85% while for False positive Rate is higher which 98.75%. For the precision of prediction class 1, the value is 62.50% and the negative prediction value is 72.48%. The rate of detection is 80.62% which is quite high. For the basic design, we just use the SVM classifier. As we can see, the result for SVM is high which mean it's compatible with the dataset.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

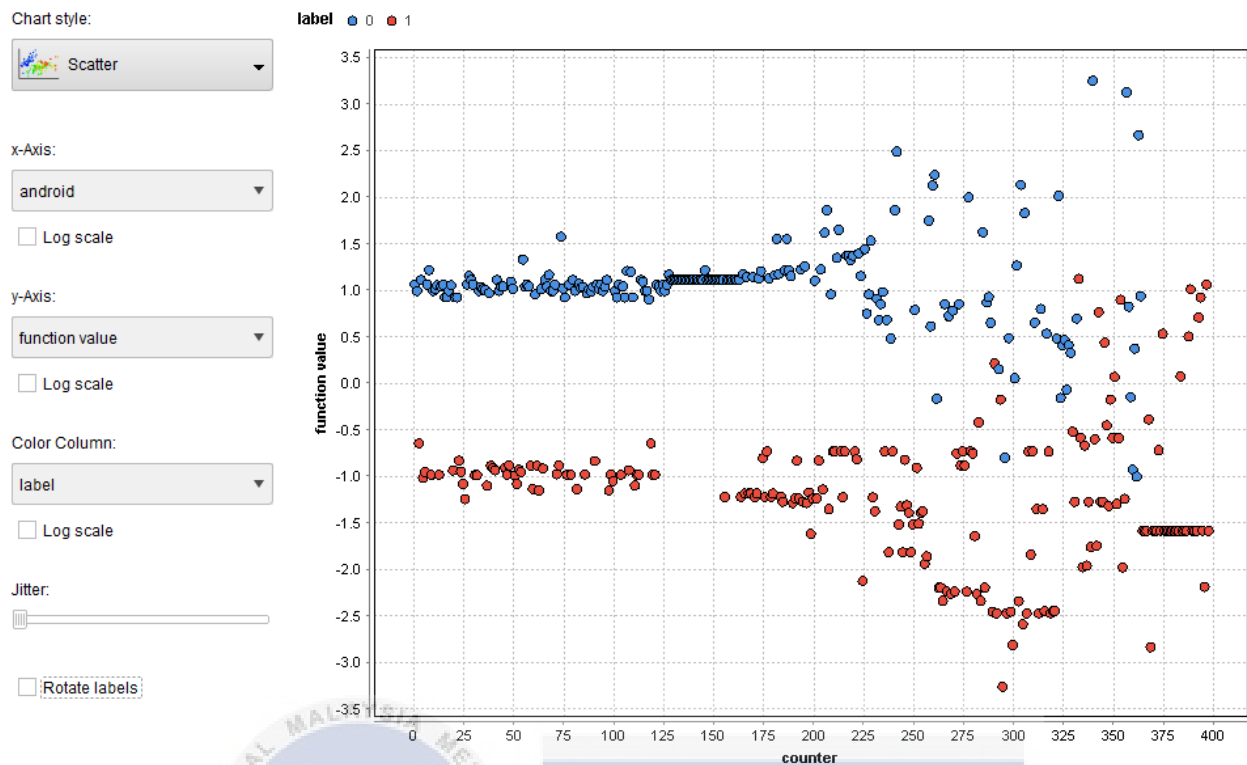


Figure 6.2: Charts of training SVM

Based on Figure 6.2, the scatter shows the original objects are rearranged by using a set of mathematical functions, known as kernels. The process of rearranging the object is known as mapping (transformation). The mapped object (malware) is linearly separable and thus, instead of constructing the complex curve (non-malware).

6.2.2 Chi Square + SVM Process

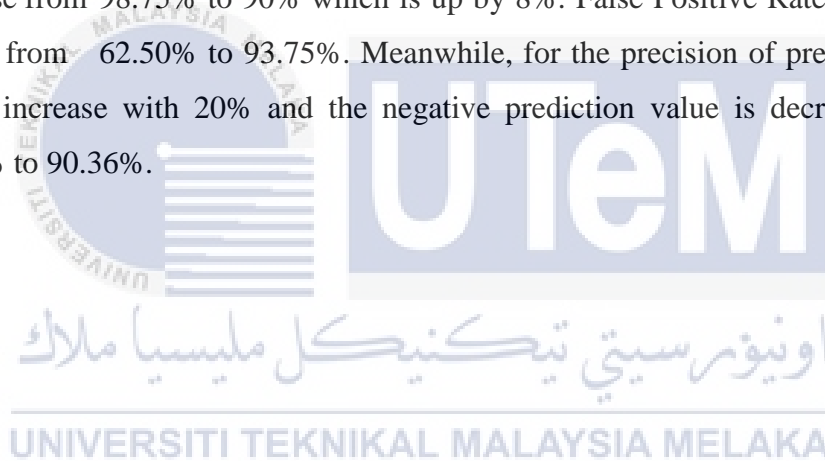
For the Chi Square Process, we add the Weight by Chi Square Statistic to calculate the relevance of the attributes by computing for each attribute of the input Example Set the value of the chi-squared statistic with respect to the class attribute.

accuracy: 91.88% +/- 7.42% (mikro: 91.88%)

	true 0	true 1	class precision
pred. 0	72	5	93.51%
pred. 1	8	75	90.36%
class recall	90.00%	93.75%	

Figure 6.3: Results of testing Chi Square +SVM

Figure 6.3 shows the results of Chi Square Process. As we can see, the accuracy rate of detection is 91.18% which is higher than the basic SVM process. So, it means that the results are increase with 10%. The True Positive Rate for Class 1 is decrease from 98.75% to 90% which is up by 8%. False Positive Rate also increases rapidly from 62.50% to 93.75%. Meanwhile, for the precision of prediction class 1 is also increase with 20% and the negative prediction value is decrease 8% from 98.04% to 90.36%.



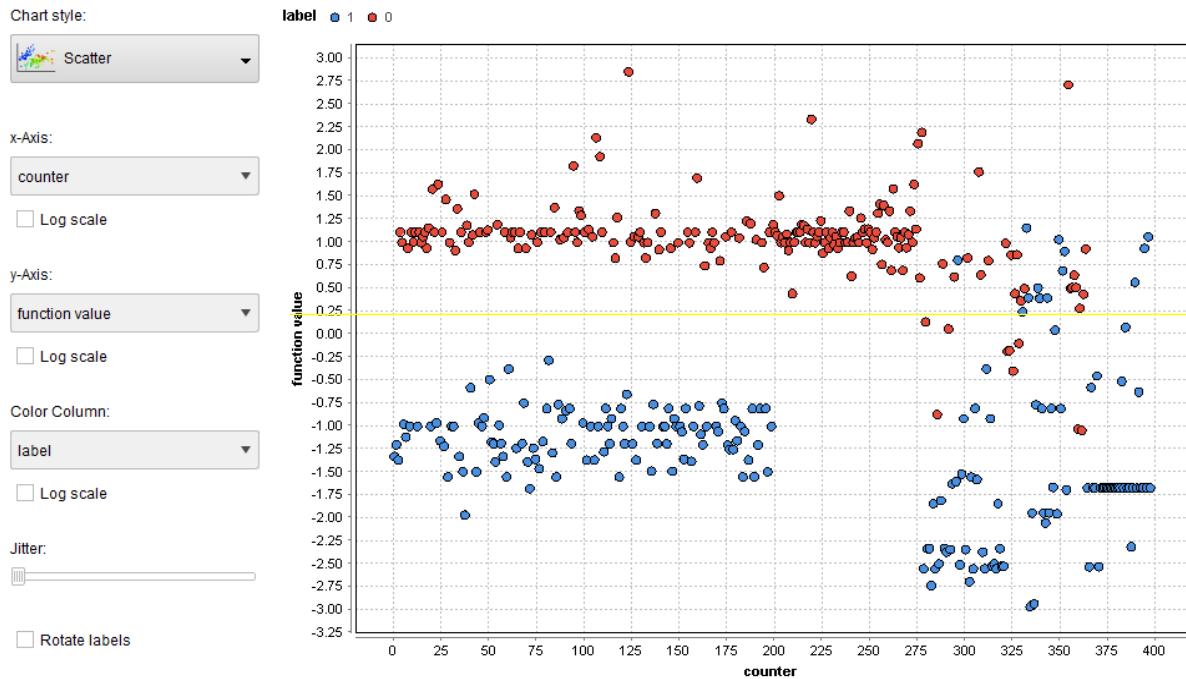


Figure 6.4: Charts of SVM with Chi Square

Figure 6.4 shows the charts of SVM with Chi Square process. As we can see, the points of scatter lie on the hyper plane which means that the data has been correctly classified. So, when new testing data is add, whatever side of the hyper plane it lands will decide the class that we assigned to it.

For the Chi Square Process, the operator calculates the relevance of the attributes. This operator also only calculates the nominal labels. The higher the weight of an attribute, the more relevant it is considered. The chi-square statistic is a nonparametric statistical technique used to determine if a distribution of observed frequencies differs from the theoretical expected frequencies. Chi-square statistics use nominal data, thus instead of using means and variances, this test uses frequencies. The value of the chi-square statistic is given by

$$X^2 = \text{Sigma} [(O-E)^2 / E]$$

Where X^2 is the chi-square statistic, O is the observed frequency and E is the expected frequency. Generally the chi-squared statistic summarizes the contradiction between the expected number of times each outcome occurs (assuming that the model is true) and the observed number of times each outcome occurs, by summing the squares of the discrepancies, normalized by the expected numbers, over all the categories.

Table 6.1: Weight of Chi Square Statistics

attribute	weight
android.permission.MODIFY_PHONE_STATE	0.203
android.permission.WRITE_SECURE_SETTINGS	0.203
android.permission.DEVICE_POWER	0.336
com.android.launcher.permission.UNINSTALL_SHORTCUT	0.336
android.permission.RECORD_AUDIO	0.716
android.permission.ACCESS_DOWNLOAD_MANAGER	1.003
android.permission.ACCESS_DOWNLOAD_MANAGER_ADVANCED	1.003
android.permission.BATTERY_STATS	1.003
android.permission.BROADCAST_PACKAGE_REMOVED	1.003
android.permission.CALL_PRIVILEGED	1.003
android.permission.CHANGE_COMPONENT_ENABLED_STATE	1.003
android.permission.CLEAR_APP_CACHE	1.003
android.permission.DELETE_CACHE_FILES	1.003
android.permission.DUMP	1.003
android.permission.INJECT_EVENTS	1.003

We also try run with other classifier such as Decision Tree, Random Forest and Naïve Bayes to see if the dataset is compatible with the machine learning. The results of each test are described in the following table.

Table 6.2: Result of Training Different Algorithm

Algorithm	Basic Process			Chi Square Process			
	Rate	Accuracy	TPR	FPR	Accuracy	TPR	FPR
Decision Tree	90.00%	86.25%	86.25%	62.50%	87.50%	86.25%	78.75%
Random Forest	75.00%	87.50%	87.50%	62.50%	91.25%	93.75%	88.75%
SVM	80.62%	98.75%	98.75%	62.50%	88.75%	98.75%	78.75%
Naïve Bayes	78.75%	91.25%	91.25%	66.25%	86.25%	97.50%	75.00%

6.3 Calculation Result

The Calculation of the result were calculated in MySQL Workbench. From machine learning, we connected the result to database and build the query to calculate the Accuracy, Detection Rate, False Alarm Attack Detection Rate and Normal Detection Rate.

Table 6.3: Result of Testing Table

	FN	FP	TP	TN	FA	DR	AC	ATTACK_DR	NORMAL_DR
SVM	3	0	18	19	0	100	92.5	85.71	100
CHI SQUARE + SVM	1	1	20	18	5.26	95.24	95	95.24	94.74

6.4 Conclusions

In summary, this chapter includes the findings and results based on parameters used on each statistical analysis approaches. On the next chapter, we will discuss in detail about project conclusion which includes strength and weakness of the project.



CHAPTER 7

CONCLUSIONS

7.1 Introduction

The previous chapter discussed on the testing activities and test results obtained. This chapter will conclude the work done in the project. It begins with the project discussions based on literature review, methodology, design, implementation and testing carried out in this project. Besides, project limitation and future works will be discussed.

7.2 Project Summarization

This research has achieved the first objective namely PO1 which is to propose Chi-square as Feature Selection method to identify relevant features accurately. PO1 is achieved in Chapter 2 by doing literature review on the parameters used by other researchers previously to identify the parameters that are used to detect the malware. Besides, PO1 is achieved in Chapter 4 by analysing the malware attribute to identify the parameters that can be used to detect malware specifically. Thus, by achieving PO1, project contribution which is the parameter for detecting malware attack is achieved by propose Chi square with SVM method.

The Second project objective namely PO2 which is to obtain android malware activity more accurately using Support Vector Machine is achieved in Chapter5 by developing an enhanced detection algorithm that is capable to detect both known and unknown malware. As PO2 is achieved, project contribution of this project which is information and knowledge about classification android malware also can be gained from this project also achieved. The effective in detecting the android malware by using SVM which is high rate of detection also can be reference to other researcher.

As the objective of the project is achieved, the problem of difficulty in detecting android malware as malware can evade detection technique is solved. There are difficulty in order to get the high rate of detection as is has different hash values that is not stored in the database which allows it to evade the signature based detection technique easily. Therefore, we also implement Chi Square Feature Selection to make it easy to calculate and interpret the data.

In conclusion, the analysis on android malware, detection processes and detection techniques have been done in order to detect the malware. Based on the research done in literature review, the enhancement is done on classification detection technique Since Support Vector Machine have the capability of reducing the original feature set, beginning the process with fewer features can affect the final performance. The project methodology consists of six phases namely Literature Review, Data Collection, Data Validation, Feature Selection and Classification, Evaluation and lastly Documentation phase.

The Design chapter consists of the experiment approach, and feature selection and classifier algorithm. The experiment approach consists of the method to carry out the experiment for data collection is explained. After the data was collected, the identified malware will be used as the parameters to detect unknown malware. The flowchart of the feature selection and classification are also designed. In the next chapter namely implementation and testing discussed the test plan, and test dataset used to test the detection algorithm. The test results are explained. The purpose of

testing the detection algorithm is to verify whether without implement Chi Square are effect the rate of detection or not. Thus Chi-square is chosen as the better statistical analysis in detecting the malware.

7.3 Project Limitation

In this thesis, we have proposed the basic SVM Process and implement Chi Square to detect the malware and determine the efficiency of the techniques analyse the behaviour of packets to obtain a better results in terms of accuracy, detection rate and precision and recall values. Certain limitation or weakness had been identified while carrying the tests. This project is only covers on detecting android malware only. As the detection algorithm produced focuses only on detecting a malware, it will not cover on malware prevention. Besides, limited of features data point is likely to be a reason precision and recall unable to distinguish and determine the behaviours of packets.

7.4 Future Works

1. Develop the algorithm to detect other malware.

The detection algorithm can be developed further to a more robust system that can detect all types of android malware attacks. This can be achieved by identifying the common parameters of all malware types. By using the sample large of data, we will see the results of the detection rate and accuracy whether it is compatible or not to detect all the malware attacks.

2. Improve to cover on malware prevention.

The detection algorithm can be improved to cover on malware prevention by generating alert to inform users that there is an attack. The algorithm can generate a ringtone alert or a written notification to inform the user that there is a malware

attack. By having this feature, the user can react to the problem immediately to prevent the malware from attacking the phone or tablet.

7.5 Conclusions

In summary, this chapter includes the project summarization, strength and weakness of the project, project contribution which is the benefits in the project and the thesis organization, project limitation and future works.



REFERENCES

- [1]. Urcuqui, C. & Navarro, A. (2016). Framework for malware analysis in Android. *Sistemas & Telemática, 14*(37), 45-56
- [2]. Alazab, M., Ventkatraman, S., Watters, P. and Alazab, M. (2011). “Zero-day Malware Detection Based on Supervised Learning Algorithms of API call Signatures.” Proceeding of the 9th Australian Data Mining Conference (AusDM’11).
- [3]. Chalurkar, SN. Meshram, B.B. (2012). “Detection of traditional and new types of Malware using Host-based detection scheme.” *International Journal of Advanced Research in Computer Engineering & Technology, 1*(4), 341-346.
- [4]. Yusof, R., Selamat S.R., Mas’ud, M.Z., Sahib, S., Abdollah M.F., Ramly, M. (2009). “Analysis of Features Selection and Machine Learning Classifier Android Malware Detection.”
- [5]. Nachirat Rachburee, Wattana Punlumjeak. (2015). “A Comparison of Feature Selection Approach between Greedy, IG-ratio, Chi-Square, and mRMR in Educational Mining.” *International Conference on Information Technology and Electrical Engineering (ICITEE), Chiang Mai, Thailand.*
- [6]. Bahassine S., Madiani A., Kissi M., (2016). “An Improved Chi-Square Feature Selection for Arabic Text Classification using Decision Tree.”
- [7]. Goertzel K.M. (2009). “Tools Report on Anti-Malware.” *Information Assurance Tools Report, IATAC.*
- [8]. Idika, N. and Mathur, A.P. (2007). “A Survey of Malware Detection Techniques.” *Purdue University.*
- [9]. Joshi, J. (2008). “Network Security: Know It All.” 1st. Ed. Massachusetts: Morgan Kaufmann Publishers. 102-105
- [10]. Landage, J. and Wankhade, M.P. (2013). “Malware and Malware detection Techniques: A Survey.” *Int. Journal on Computer Science and Engineering, 2*(12), 62-68.

- [11]. Victor Wahanggara and Yudi Prayudi. (2015). "Malware Detection through Call System on Android Smartphone Using Vector Machine Method". Fourth International Conference on Cyber Security, Cyber Warfare and Digital Forensic.
- [12]. Junmei Sun, Kai Yan, Chunlei Yang. (2015). "Malware Detection on Android Smartphones using Keywords Vector and SVM". Hangzhou Institute of Service Engineering.
- [13]. Tan Bao, Takeshi Takashi, Daisuke Inoue. (2016). "Integration of Multi-modal Features for Android Malware Detection Using Linear SVM." 2016 11th Asia Joint Conference on Information Security.
- [14]. Morales, J.A (2008). "A behaviour based approach to virus detection." Ph.D Thesis, Florida International University.
- [15]. Speedguide (2016). [Online]. Vulnerable ports [Accessed 20 March 2016]; Available at; http://www.speedguide.net/ports_sg.php?page=0&sort=&category=&seek=

