

MOBILE MALWARE DETECTON USING RNN-LSTM THROUGH OPCODE



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS LAPORAN

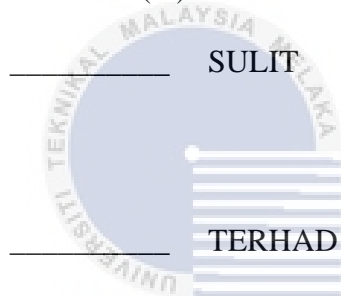
JUDUL: MOBILE MALWARE DETECTION USING RNN-LSTM THROUGH OPCODE

SESI PENGAJIAN: [2020/ 2021]

Saya: ___AHMAD RAZIN BIN AZMAN___

mengaku membenarkan tesis Projek Sarjana Muda ini disimpan di Perpustakaan Universiti Teknikal Malaysia Melaka 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

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

TERHAD

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

TIDAK TERHAD

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

(TANDATANGAN PELAJAR)

Alamat tetap: LOT 8721 JALAN
NAKHODA KANAN LORONG 21 KG
NAKHODA 68100 BATU CAVES
SELANGOR

(TANDATANGAN PENYELIA)

TS. DR. MOHD ZAKI BIN MAS'UD

Tarikh: 08/09/2021

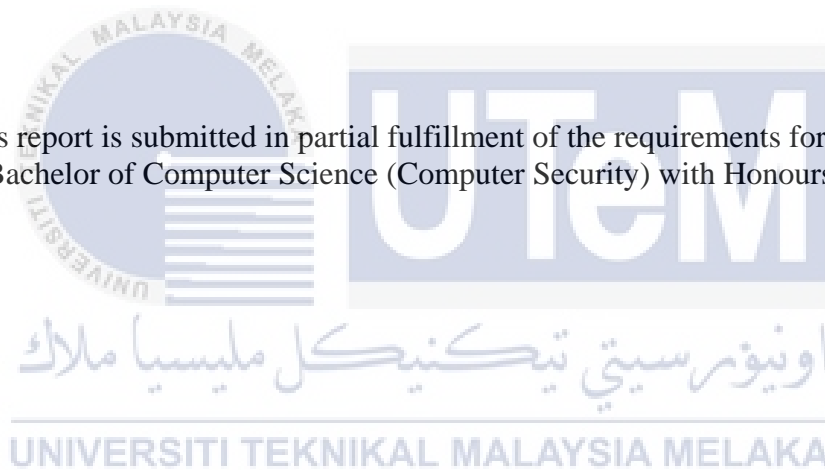
Tarikh: 08/09/2021

CATATAN: * Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa.

MOBILE MALWARE DETECTION USING RNN-LSTM THROUGH OPCODE

AHMAD RAZIN BIN AZMAN

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

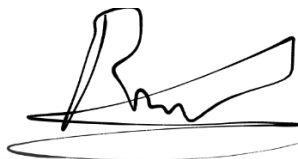


FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

DECLARATION

I hereby declare that this project report entitled
MOBILE MALWARE DETECTION USING RNN-LSTM THROUGH OPCODE
is written by me and is my own effort and that no part has been plagiarized
without citations.



STUDENT :

(AHMAD RAZIN BIN AZMAN)

Date : 08/09/2021



I hereby declare that I have read this project report and found

this project report is sufficient in term of the scope and quality for the award of
Bachelor of Computer Science (Computer Security) with Honours.



SUPERVISOR :

(TS. DR. MOHD ZAKI BIN MAS'UD)

Date : 08/09/2021

DEDICATION

To My beloved parents, for always give me support and encourage me on everything.
My helpful supervisors, for always give me those brilliant ideas and precious time to guide me throughout completing this final year project. My fellow friends, for always give me good advice and helping me.



ACKNOWLEDGEMENT

First and foremost, in the name of Allah, I would want to convey my heartfelt gratitude to numerous persons who have helped me during my degree studies. TS. DR. Mohd Zaki Bin Mas'ud, my supervisor, I want to express my heartfelt appreciation for all of your expertise and helpful suggestions in assisting me in finishing my final year project. Also, thank you for your compassion and for devoting so much of your time to me during this endeavour. I'd also want to thank my evaluator, PM TS. DR Siti Rahayu Bt Selamat, for taking the time to assess me and provide honest criticism on my research.

I'd also like to thank my loving parents and family members, who have always loved and inspired me to accomplish my final year project. I shall never forget their belief in my ability to succeed in my studies.

Last but not least, I want to express my gratitude to my friends who have always given up their valuable time to assist me and make suggestions for my final year project.



ABSTRACT

The popularity of Android and the development of third-party app stores have led to Android malware growing in recent years. Emerging Android malware families are progressively implementing advanced detection avoidance tactics, necessitating more effective Android malware detection methodologies. Hence, in this project analyse an opcode features-based framework to identifying and categorizing Android malware using RNN-LSTM. This method allows for automatic feature discovery without the need for previous expert or subject knowledge for pre-defined features. Identify mobile malware using opcode is the aim of this paper. Not only that, in this research create and analyse RNN-LSTM models for mobile malware detection through opcode. In this research, synthesis all material that have related to the mobile malware detection from any journal. Summarizing the material, analyse, interpret and make implications for researcher in order to properly draw a conclusion to provide a solution. After that, in this project experimental setup is required, providing an isolation environment to prevent malware harmful PC host. The activities in the isolation environment for doing static analysis on malware samples using the jadx-gui tool involve Android package extraction and code disassembly. In this isolation environment, 1000 malware samples and 1000 benign samples will use the most recent version of Python to extract opcode. Google Colaboratory RNN-LSTM design is the way to train all data sets (80% training 20% testing). This research aims to analyse and evaluate the output of a dataset to obtain the value of the True Positive rates (TPR) and False Positive Rates (FPR).

ABSTRAK

Populariti Android dan pengembangan kedai aplikasi pihak ketiga menyebabkan malware Android berkembang dalam beberapa tahun terakhir. Keluarga malware Android yang muncul secara progresif menerapkan taktik penghindaran pengesanan lanjutan, memerlukan metodologi pengesanan malware Android yang lebih berkesan. Oleh itu, dalam projek ini menganalisis kerangka kerja berasaskan opcode untuk mengenal pasti dan mengkategorikan perisian hasad Android menggunakan RNN-LSTM. Kaedah ini membolehkan penemuan ciri automatik tanpa memerlukan pengetahuan pakar atau subjek sebelumnya untuk ciri yang ditentukan sebelumnya. Mengenal pasti perisian hasad mudah alih menggunakan opcode adalah tujuan makalah ini. Penyelidikan ini bukan sahaja membuat dan menganalisis model RNN-LSTM untuk pengesanan malware mudah alih melalui opcode. Dalam penyelidikan ini, sintesis semua bahan yang berkaitan dengan pengesanan malware mudah alih dari jurnal mana pun. Meringkaskan bahan, menganalisis, mentafsir dan membuat implikasi kepada penyelidik agar dapat membuat kesimpulan dengan betul untuk memberikan penyelesaian. Setelah itu, dalam projek ini diperlukan penyediaan eksperimen, menyediakan persekitaran pengasingan untuk mengelakkan host PC berbahaya dari malware. Kegiatan dalam lingkungan pengasingan untuk melakukan analisis statik pada sampel malware menggunakan alat jadx-gui melibatkan pengekstrakan paket Android dan pembongkaran kod. Dalam persekitaran pengasingan ini, 1000 sampel malware dan 1000 sampel jinak akan menggunakan versi terbaru Python untuk mengekstrak opcode. Model RNN-LSTM menggunakan Google Colaboratory adalah cara untuk melatih semua set data (latihan 80% ujian 20%). Penyelidikan ini bertujuan untuk menganalisis dan menilai hasil dari set data untuk mendapatkan nilai True *Positive rates (TPR)* dan False *Positive Rates (FPR)*.

Table of Contents

DECLARATION	i
DEDICATION	ii
ACKNOWLEDGEMENT	iii
ABSTRACT	iv
ABSTRAK	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF ABBREVIATION	x
CHAPTER 1: INTRODUCTION	1
1.1 Introduction.....	1
1.2 Research Background.....	1
1.3 Problem Statement.....	2
1.4 Project Question.....	2
1.5 Project Objective.....	3
1.6 Project Scope.....	3
1.7 Project Contribution.....	3
1.8 Report Organization.....	4
1.9 Conclusion.....	5
Chapter 2: LITERATURE REVIEW	6
2.1 Introduction.....	6
2.2 Keyword.....	6
2.2.1 Deep Learning.....	6
2.2.2 Mobile Malware.....	6
2.2.3 Neural Network.....	7
2.2.4 Operation Code (OPCODE).....	7
2.2.5 RNN-LSTM.....	7
2.3 Related Work.....	7
2.3.1 Introduction to mobile malware.....	8
2.3.2 Mobile malware detection base-method signature and anomaly.....	9
2.3.3 Machine Learning in mobile malware.....	10
2.3.4 Deep Learning method in mobile malware.....	10
2.4 Propose solutions.....	13
2.5 Conclusion.....	13
Chapter 3 : METHODOLOGY	15
3.1 Introduction.....	15

3.2 Research Methodology	15
3.3 Experimental Setup	16
3.4 Project Milestone	18
3.5 Conclusion	20
Chapter 4: ANALYSIS	21
4.1 Introduction	21
4.2 Dataset	22
4.3 Static Analysis	23
4.4 Application code review: Opcode	34
4.4.1 Process of application code review through opcode	34
4.4.2 Opcode Analysis	36
4.5 Conclusion	39
Chapter 5: DESIGN RNN-LSTM MODEL	40
5.1 Introduction	40
5.2 Flowchart	40
5.3 Pseudocode	40
5.4 Modelling RNN-LSTM	42
5.4.1 Input Dataset of Malware and Benign	42
5.4.2 Drop columns	43
5.4.3 Split Dataset	43
5.4.4 RNN-LSTM Model	45
5.5 Results of data training and testing	46
5.6 Conclusion	51
Chapter 6: CONCLUSION	52
6.1 Introduction	52
6.2 Research Contribution	52
6.3 Research Limitation	53
6.4 Future Research	53
6.5 Conclusion	55
REFERENCES	56
APPENDIX I	57
APPENDIX II	58
APPENDIX III	58
APPENDIX IV	59
APPENDIX V	60

LIST OF TABLES

Table 1.1 Problem Statement	2
Table 1.2 Project Question.....	3
Table 1.3 Project Objective.....	3
Table 2.1 Type of Malware.....	8
Table 2.2 Related Works.....	11
Table 3.1 Project Milestone	18
Table 4.1 Dataset Used in Previous Research.....	22
Table 4.2 Similarity Online Analysis and Static Analysis	32
Table 4.3 Basic opcode grammar.....	37
Table 5.1 Summarise of Experiment.....	51
Table 6.1 Recommended Specification	54

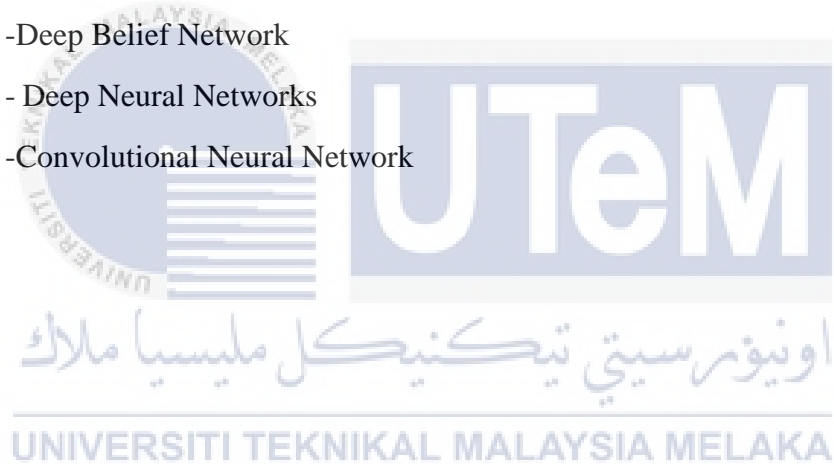


LIST OF FIGURES

Figure 3.1 Research Methodology	15
Figure 3.2 Experimental Phase	16
Figure 3.3 Process Opcode Extraction	17
Figure 3.4 Gantt Chart	19
Figure 4.1 Analysis phase	21
Figure 4.2 jadx-gui	23
Figure 4.3 Opcode	24
Figure 4.4 VirusTotal Results	24
Figure 4.5 Applications Permissions VirusTotal	25
Figure 4.6 Risk Assessment Hybrid Analysis	25
Figure 4.7 Applications Permission at AndroidManifest.xml	26
Figure 4.8 Function code to monitors all incoming SMS messages	26
Figure 4.9 function code to install any package	27
Figure 4.10 function code to upload file zjms.txt and zjphonecall.txt into their server	27
Figure 4.11 function code to get device info	27
Figure 4.12 function code to call using victim devices	28
Figure 4.13 function code to send message using victim devices	28
Figure 4.14 function to open internet connection	28
Figure 4.15 function to execute the code after reboot	29
Figure 4.16 function code to get incoming call and save it	29
Figure 4.17 function code to delete package	29
Figure 4.18 android.location.LocationListener library	30
Figure 4.19 function code Location g	31
Figure 4.20 Geo-location API	31
Figure 4.21 Function of Malicious code	34
Figure 4.22 Function of code in smali	34
Figure 4.23 extracting opcode in notepad	35
Figure 4.24 Function of code to install package	36
Figure 4.25 Function of code to install package in smali	36
Figure 4.26 line of code in smali to install package	37
Figure 4.27 dataset in csv file	38
Figure 5.1 RNN-LSTM Flowchart	40
Figure 5.2 Dataset of sample	42
Figure 5.3 Number of malware and benign	42
Figure 5.4 data_in and labels shape	43
Figure 5.5 pad_sequence output	44
Figure 5.6 Final shape of data	44
Figure 5.7 Model RNN-LSTM summary	46
Figure 5.8 Graph of accuracy	47
Figure 5.9 Graph of loss	48
Figure 5.10 Graph of accuracy	49
Figure 5.11 Graph of loss	49
Figure 5.12 Graph of Accuracy	50
Figure 5.13 Graph of loss	51

LIST OF ABBREVIATION

RNN	- Recurrent Neural Network
LSTM	- Long Short Term Memory
DEX	-Dalvik Executable
OPCODE	-Operation Code
API	- Application Programming Interface
TPR	-True Positive Rate
FPR	- False Positive Rate
TCP	- Transmission Control Protocol
HTTP	- Hypertext Transfer Protocol
DBN	-Deep Belief Network
DNN	- Deep Neural Networks
CNN	-Convolutional Neural Network



CHAPTER 1: INTRODUCTION

1.1 Introduction

The research history, issue statement, research topic, research priorities, research scope, research technique, and analytic walkthrough for the full research are all included in this chapter.

1.2 Research Background

In this project each mobile malware sample will be statically analysed to detect their behaviour. The static analysis decompiles the chosen .apk files and extracts and examines the associated functions. This research is used extensively for checking licenses, API calls and determining the code structures and components of a given .apk file. When the files of .apk are decompiled, there are some files, such as META-INF, lib, res, properties, AndroidManifest.xml, classes.dex, and resources.arsc, that are stored there. In static analysis, the AndroidManifest.xml and classes.dex are popular as they show any suspicious application's true purpose.

In structured research, AndroidManifest.xml and classes.dex are typically used so they display the true intent of any questionable programs. In the first step, a managed environment is developed by VMware to evaluate mobile malware without the possibility of infection on the host PC.

Secondly, in order to receive classes and manifest data, the .apk files are deleted. Manifest file holds configuration files, operation and permissions, while the class file contains all the Java codes used. Next, you can decompile the classes.dex file into a Java class file called .jar. Analysis of codes and methods in Java class will display malicious requests. Some typical malicious activities include root authorisation, the stealing of confidential data, such as IMEI and country numbers, the dispatch and reception of C&C server orders. The findings obtained in static analysis are last but not least used in the fifth stage to represent the chain of malevolent practices. The final step is important because it allows the researcher to track the patterns of the malware during an attack.

In this research, in order to increase the accuracy of the mobile malware detection using method in deep learning call Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) through Operation Code(opcode). This study plan to establish some advancement focused on the methods that have been suggested previously.

1.3 Problem Statement

Due to the rising use of complex detection avoidance methods and the need to update signature databases on a regular basis, previous research has shown that traditional signature-based approaches, which are employed by most antivirus scanners, are unsuccessful in detecting new infections. Various techniques based on analysing dynamic application activity, requested permissions, API calls, and other aspects have been presented. However, expert analysis or domain expertise are still frequently used to design or select the discriminative aspects that are provided to the machine learning system that makes the final classification decision. To train on, machine learning requires huge data sets that are complete, unbiased, and of high quality. At times, they may have to wait for fresh data to be created. Machine Learning is also self-contained, yet it is susceptible to errors. Assume you're trying to train an algorithm with data sets that aren't big enough to be useful. You get biased predictions as a result of a biased training set. As a result, customers are bombarded with irrelevant advertisements. Such errors may initiate a cascade of errors that go unnoticed for a long period in the setting of machine learning. It takes a long time to figure out what's causing the problem, and even longer to solve it, once they're discovered. Table 1.1 summarises the problem statement for the project.

Table 1.1 Problem Statement

No	Problem Statement
1	The capability traditional signature base approaches in detecting mobile malware
2	The massive data sets to train on in the machine learning.
3	The high error-susceptibility of machine learning in classification of mobile malware

1.4 Project Question

In reality, there are a great deal of to detect mobile malware and each of them have a different behaviour through opcode. Hence, it is important to study the mobile malware behaviours during static analysis and the best way to detect it. Next, we can start to identify the suitable method and algorithm in RNN-LSTM that uses a higher accuracy in detecting mobile malware.

Table 1.2 Project Question

No	Project Question
1	What is the accuracy of a non-mobile malware and mobile malware?
2	How far RNN-LSTM contribute in mobile malware detection?

1.5 Project Objective

There are three objectives of this project. Table 1.2 below shows the summary of the project objectives for this project.

Table 1.3 Project Objective

No	Project Objective
1	To detect mobile malware through opcode
2	To develop RNN-LSTM model for mobile malware detection through opcode
3	To evaluate the RNN-LSTM mobile malware detection

1.6 Project Scope

This project is developed in order to detect malware at executable by calculating it's accuracy. The dataset of the sample mobile malware will be collect and decompiled, there are some files, such as META-INF, lib, res, properties, AndroidManifest.xml, classes.dex, and resources.arsc, that are stored there.. The programming language will be use is python and the operating system is Ubuntu for better isolated environment to make this project successful.

1.7 Project Contribution

This project is important as it can be used by any researcher to conduct more research or to established best method in detection of the mobile malware using deep learning through RNN-LSTM for enabling them to compare which form of method can produce a better accuracy.

1.8 Report Organization

This section is provided for the description of the report organization. Overall, the report contains six (6) chapters:

Chapter 1: Introduction

This chapter consists of the research background, problem statement, project question, project objective, project scope, and project contribution.

Chapter 2: Literature Review

Reviews on the terminologies related to the project topic on the basis of related works, critical review of the current problems and proposed solutions have been included in this chapter.

Chapter 3: Project Methodology

This chapter describes the flow or methodology used in the process of completing this project as well as how it develops its analysis.

Chapter 4: Analysis

This chapter provide project design and process step by step must be state in this chapter.

Chapter 5: Design RNN-LSTM

This chapter provides the details of the implementation of the project including the description on how the project is carried out and how the result is produced.

Chapter 6: Conclusion

The last chapter addresses the conclusion and discussion of the project. Summary of the conclusion will also be stated in this chapter

1.9 Conclusion

In conclusion, this chapter has given an explanation and a better understanding on the objectives of the project, regarding how it would benefit in the cyber security field in the future. Next, this research will be focusing on finding the best method in deep learning and producing method of detection mobile malware with higher accuracy.



Chapter 2: LITERATURE REVIEW

2.1 Introduction

A literature review is a thorough overview of prior studies on a particular topic. The literature review examines scientific journals, books, and other references that are applicable to a specific research subject. This previous study should be enumerated, defined, summarized, critically evaluated, and clarified in the analysis. It should provide a theoretical foundation for the study and assist you (the author) in determining the scope of the study. The literature review respects the findings of prior scholars, assuring the reader that your work is well-thought-out.

By referencing a prior work in the field of research, it is believed that the author has read, analysed, and assimilated the work into the current work. A literature review provides the reader with a "landscape," allowing them to fully comprehend the field's innovations. The reader will see from this landscape that the author has incorporated all (or the overwhelming majority) of recent, important works in the field into her or his research.

2.2 Keyword

2.2.1 Deep Learning

Deep learning is a branch of machine learning in which vast volumes of data are learned using multi-layered neural networks modelled after the human brain. Deep learning algorithms conduct calculations and make predictions consistently within each layer of the neural network, increasingly 'learning' and improving the precision of the result over time.

2.2.2 Mobile Malware

Mobile malware, as the name implies, is malicious software designed to attack mobile phone operating systems. There are several common kinds of smartphone malware variants, as well as different delivery and infection processes.

It was only a matter of time before hackers shifted strategies as more people moved away from desktop operating systems in favour of handheld devices. At the moment, smartphone attacks are a tiny fraction of those that threaten desktop computers. Mobile security risks are quickly becoming a growing problem as more critical and potentially high-value activities are carried out on mobile devices.

2.2.3 Neural Network

Artificial neural networks (ANNs) and synthetic neural networks (SNNs) are a branch of machine learning that are at the core of deep learning algorithms. Their name and form are derived from the human brain, and they resemble the way biological neurons communicate with one another.

2.2.4 Operation Code (OPCODE)

An opcode (abbreviated from operation code) is the part of a computer language instruction that determines the operation to be executed. It is also known as instruction machine code, instruction code, instruction syllable, instruction parcel, or opstring. Many instructions, in addition to the opcode itself, also specify the data they would process in the form of operands. Opcodes can be used in abstract computer machines as part of their byte code requirements, in addition to being used in the instruction set architectures of different CPUs, which are hardware computers.

2.2.5 RNN-LSTM

Long Short Term Memory (LSTM) is a supervised Deep Neural Network type that excels at time-series prediction. It's a kind of RNN (Recurrent Neural Network). An LSTM model examines data from the previous "n" days (timestep) (also known as lag) and forecasts how the sequence will proceed in the future.

RNN is a kind of artificial neural network (ANN) that has a recurring relation to itself. RNN learns the influence of previous input $x(t-1)$ as well as current input $x(t)$ when estimating the output at time "t" using this repeated relation (t). This provides RNN with a sense of time. At time "t," the secret layer activations measured at time "t-1" are used as an input.

2.3 Related Work

2.3.1 Introduction to mobile malware

With the proliferation of mobile devices, we have entered the mobile era, witnessing a rapidly growing popularity of smartphones. The mobile device is no longer confined to the communication services in traditional sense(Wang et al., 2019). Malicious software intended to target cell phone operating systems is known as mobile malware. There are several various types of mobile malware, as well as different distribution and intrusion methods (Kumar et al., 2019). Table 1.3 below are several class of malware (Jul, 2019)

Table 2.1 Type of Malware

Type of malware	Explanations
Virus	Viruses are known to penetrate mobile computers and smartphones without the user's permission. After successfully infiltrating the device, the viruses bind to some program files and begin executing malicious functions that have been coded.
Worm	Worms are typically designed to replicate themselves inside a computer system. It then goes on destroying data and files on the server or mobile devices.
Trojan	Trojans are programmed to steal banking information or passwords while also causing a denial of service (DoS) assault on the server.
Backdoor	Backdoors are created by programmers to make it easier for them to administer programs remotely. When it is used for malicious purposes, however, attackers may send ransomware, viruses, and even gain access to a computer device in order to carry out malicious activities.
Spyware	Spyware is software that monitors a computer's operations and can also be used to steal a victim's login credentials.
Adware	Adware poses no risk to computers or handheld devices because it is only used to deliver advertisements, which can be malicious at times.
Ransomware	Ransomware is a form of malicious software that encrypts the data and files of its victims. Victims will be asked to pay a large

	amount of money to the perpetrators in order to open or decrypt the files and documents.
Rootkit	Rootkit, on the other hand, is a malicious application that is installed in a computer system to allow uncertified staff access. The attackers will then remotely execute files or change device settings.
Botnets	Botnets were frequently used by attackers to carry out large-scale network attacks, such as DoS attacks, that flooded resources.
Keylogger	The keylogger works by recording all of the keystrokes. After that, the registered values are used to retrieve login credentials and other financial data. Previous research has shown that mobile malware takes on those characteristics after infecting a mobile computer.

2.3.2 Mobile malware detection base-method signature and anomaly

The two major methods of detecting and alerting on risks are signature-based and anomaly-based detections. Anomaly-based detection is used for variations in behaviour, while signature-based detection is used for known attacks. Signature-based identification is based on a list of established signs of compromise that has been pre-programmed (IOCs). Malicious network attack actions, email subject line text, file hashes, identified byte sequences, and malicious domains are all examples of IOCs. Signatures can also provide network traffic warnings, such as identified malicious IP addresses trying to gain access to a device.

In comparison to signature-based detection, anomaly-based detection may identify unexpected irregular behaviour. Anomaly-based detection involves first creating a normalized context for the system and then matching behaviour to the baseline. An warning is activated when an incident seems to be out of the ordinary. Anything that deviates from the normalized baseline will set off an alert, such as a user signing in during non-business hours, an influx of new IP addresses trying to link to the network, or the addition of new devices to a network without authorization. Based on other research that analyses HTTP requests and TCP Flows to determine whether the apps is malicious.

The network behaviours of malware can still present non-trivial anomalies that can be identified by advanced detectors which provides us with a keen insight in malware detection (Wang et al., 2019). That research is using anomaly base-detection on malicious network traffic. Some of previous research are using same base-method detection that using network traffic for mobile detection (Feng et al., 2020). Most of the research using signature base-method detection. The researcher using extraction API method calls by using Maldozer framework(Karbab et al., 2017). In other method are using opcode features API(Kumar et al., 2019) functions are derived from smali files, which are dex files that have been disassembled. The smali file is divided into process blocks, and the Dalvik opcode frequency of each method is determined by scanning Dalvik bytecodes. Furthermore, during bytecode scanning, the presence of dangerous API invocations in the system is tested, and the frequency of dangerous API invocation for each method is determined.

2.3.3 Machine Learning in mobile malware

There are a lot of method to detect mobile malware such as machine learning. Machine learning is a branch of computer science that is distinct from conventional computing methods. Algorithms are collections of directly coded instructions used by computers to quantify or solve problems in conventional computing. Machine learning algorithms, on the other hand, enable computers to train on data inputs and then use statistical analysis to produce values that are within a certain range. As a result, machine learning makes it easier for machines to build models from sample data and simplify decision-making processes based on data inputs.

There is a research that analysed malicious network traffic using machine learning by decision tree model (Wang et al., 2019) . But a lot of research like to use deep learning as a mobile malware detection for better accuracy and less false alarm.

2.3.4 Deep Learning method in mobile malware

Deep learning is a form of machine learning in which large amounts of data are learned using multi-layered neural networks that are inspired by the human brain. Inside each layer of the neural network, deep learning algorithms perform calculations and make predictions continuously, 'learning' and refining the accuracy of the result over time.

A lot of research using deep learning to detect mobile malware. Network traffic dataset will input into CACNN layer. There were two components of the CACNN layer. One is a conditional classification model for determining whether or not an application is malicious.(Feng et al., 2020). There are a lot of method in deep learning that are using by previous research. Multimodal neural network is one of the method that uses five features vectors and is inputted separately to the initial networks which consist of five DNNs (Deep Neural Network). The initial networks are not linked to each other, and the merger layer, which is the first layer of the final network, is connected to the last layers of the initial networks. The classification results are generated by the final network, which is a DNN. Each of the initial networks' DNNs has an input layer and two hidden layers, with each receiving connections only from the previous layer.(Kumar et al., 2019).

On the other hand, previous are also using Maldozer framework that based on an artificial neural network. In this framework to allow malware detection and family attribution, the raw sequences of API method calls, as they appear in the DEX file, are used as input. Using only the sequences of raw method calls in the assembly language, MalDozer can automatically identify malicious patterns during testing. MalDozer detects malware with high precision through various datasets.(Karbab et al., 2018)

One of the common approach are using different deep architectures model such as Deep Belief Networks (DBN) and convolutional neural networks.(Yuan et al., 2016) Table 1.4 below shows the summary of the related work below:

Table 2.2 Related Works