FILE-LESS MALWARE DETECTION

A DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

MASTER OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY

Submitted by:
Himanshu Anand

(2K20/ISY/09)

Under the supervision of

PROF. KAPIL SHARMA
HOD, DEPT. IT DTU

DEPARTMENT OF INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

BawanaRoad, Delhi-110042

May 2022

CANDIDATE DECLARATION

I hereby certify that the work which is presented in the Major Project II entitled **“FILE-LESS MALWARE DETECTION”** in fulfilment of the requirement for the award of the Degree of Master of Technology in Information Technology and submitted to the Department of Information Technology, Delhi Technological University, Delhi is an authentic record of our own, carried out in May 2022, under the supervision of **Prof Kapil Sharma**.

Himanshu Anand
2K20/ISY/09

CERTIFICATE

To the best of my knowledge, the above work has not been submitted in part or full for a Degree or Diploma to the University or elsewhere. I, Further certify that the information given by the student is correct.

Himanshu Anand	**Prof. Kapil Sharma**
2K20/ISY/09	**HOD, DEPT. IT DTU**
Place: Delhi	
Date:	

ABSTRACT

Today, Everything is present digitally on our computer system and every organisation uses the computer for its daily work, Nearly 50 billion devices are currently connected to the Internet. Every device which is connected to the internet is vulnerable to cyberattack, to protect them from any attack multiple techniques are introduced like, Anomaly-based detection, Specification-based detection and Signature-based detection but with the evolution, in cybersecurity measures, the threat has also evolved with time, especially in the field of malware.

Typically, malware is based on the file system which can be detected by the antivirus software. To overcome this file-less malware is developed by the attackers which do not use any file system, so it bypasses any signature-based detection. File-less malware can be dangerous for any organisation because of its persistence to over come from the danger of file-less malware few method are developed like, Detection on the basis of system behaviour, detection on the basis of rules and detection on the basis of attack. To make the computer system secure continuous analysis of the malware is necessary, So that malware can be detected easily.

This project uses 4 different machine learning algorithms i.e Logistic Regression, K-Neared Neighbour, Decision Tree and Support Vector Machine all the algorithm comes under supervised learning and are capable of detecting any type of labeled value.

Our dataset contains 10 different file-less malware and we have applied the all the algorithm in it for the detection part.

ACKNOWLEDGEMENT

I would like to take this opportunity to express our profound gratitude and deep regard to my mentor **Prof. Kapil Sharma**, HOD (Department of Information and Technology), Delhi Technological University (Formerly Delhi College of Engineering), New Delhi for his exemplary guidance, valuable feedback and constant encouragement throughout the duration of the seminar. His valuable suggestions were of immense help throughout our project work. His perspective criticism kept us working to make this project in a much better way. Working under him was an extremely knowledgeable experience for me. And I would also like to thanks all other faculty members of my department for their astute guidance, constant encouragement and sincere support for this seminar work.

Himanshu Anand
2K20/ISY/09

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Malware is a malicious program that gets inside a device with or without user permission. The term malware is derived from the words 'malicious' and 'software'. It is a major problem in today's world where everything is connected through internet. Its ability to spread and remain hidden is continuously increasing. [1,2] While more organisations attempt to solve the problem, the number of sources spreading malware grows at an exponential pace and is out of reach. The majority of malware reaches the device when downloading files from the Internet. Once it gets inside the device, it searches for operating system bugs and executes unwanted command on the system, eventually slowing down the system's efficiency or leaking the data it contains.

Malware has the ability to corrupt other executable code, system directories, drive boot partitions, and generate unwanted network traffic, resulting in a denial of service. When a user executes an infected file, it becomes a resident in memory and infects all subsequent files that are being executed. Malware take control of an operating system and exploit other computers on the network if they have a flow. Such malicious programs are often known as worms, and they have a negative impact on computer performance, resulting in a slowdown [3].

Some malware is extremely simple to find and uninstall using antivirus. These antivirus programs keep a database of malware signatures, which are binary patterns that are exclusive to malicious code. Files suspected of being infected are examined for the existence of virus signatures. This method of identification was effective before the attacker started creating file less malware malware. These malware variants escape detection by using cryptographic methods to evade signature-based detection.

To detect malicious code, security products such as virus scanners check for characteristics byte sequences (signature). The detector's quality is measured by the detection techniques used. A successful malware detection strategy should be able to detect malicious code that is concealed or inserted in the original software, as well as detect previously unknown malware. Commercial virus scanners have very poor resistance to new attacks because malware authors are constantly developing new obfuscation techniques to help the malware avoid detection.

1.1. Malware

Malware is a well known name in the world of cyber security. It is malicious software that is designed by the cyber attacker to gain unauthorised access to the computer for gathering sensitive information, gain control of the computer system, or obstruct the computer operations. Today, malware has become a major threat and it is growing and evolving day by day. As different organisations are developing different methods of detecting the malware, the attackers are also upgrading their malware and distributing it.

Generally, malware gets inside the system by the files downloaded from the internet once it gets into the host computer, it scans the vulnerabilities of the operating system and then it performs undesired processes resulting in slower performance of the system. Malware also has ability to infect the other software files which are present on the computer and chocks the network line which result in DOS (Denial of service).

Some malware can be detected with the help of Antivirus and after detection they can be erased easily. Software which are used for detecting the malware stores the malware signature. While scanning the computer with antivirus suspected files are checked for the presence of malware signature this approach only works if the attacker does not encrypt the malware signature.

1.2. Types of Malware

Virus: Like the flu virus, the computer virus is engineered to spread from one host to another and to reproduce itself. Similarly, computer viruses do not evolve and propagate without code, such as a file or email, because flu viruses cannot replicate without a host body. It can be transferred from one host to another by the use of portable devices which are used for data transfer.

Worms: Worms is a malicious self-replicating software that spreads their copies without any human interference i.e. without any file execution and they do not attach to any software or document. They use network connection to replicate from one system to another by sending there copies and due to which bandwidth can get affected.

Spyware: The term “Spyware” is used for the collection of software which are used for monitoring and gathering information about the host e.g. frequently visited websites, banking details, which key is pressed by the user.

Adware: It is software that gets installed on a host computer by attaching it to free software and then starts showing advertisements or downloading it on the computer without the user permission.

Trojan: Trojan horse mimics like original software but loaded with the malware after it gets loaded in the host computer then the attacker can monitor the activity.

BotNet: Botnet is a collection of devices which are connected over the internet and connected with other bots. Together they can be used for doing DDOS(Distributed Denial of Service) attack, send spam and it also provide owner of these bots to access the device and their connections.

RootKits: It is a specific type of malware which is highly vicious.it gives access path to other worms, Trojans or malware because it gets the root access of the host computer and gives access to attacker allowing him to access it.

Mobile Malware: These are the malicious files which only focus on mobile phones or devices which have internet connection, by releasing their personal data on the device.

1.3. History of Malware

S.No	Malware Type	History
1	**Viruses**	In 1986, the virus name “Brain” started infecting thefloppy disk.
2	**Worms**	In 1988, a Student named Robert Morris released theworm on the internet.
3	**Spyware**	In 1995, it was used in a post that was created tomake fun of Microsoft's business model.

S.No	Malware Type	History
4	**Adware**	In 1970, Arpanet company got infected with the virus,that virus displayed a message called “Im the Creeper. Catch me if you can”.it was the first occurrence of Adware.
5	**Trojans**	In 1975, John Walker developed the Trojan called“ANIMAL”.
6	**Botnet**	In 1999, two botnet programs “Sub7” and “Pretty Park” was released into IRC network. Main task of bots was to connect to IRC channel and listen malicious commands
7	**RootKits**	In 1999, Greg Hoglund created a Trojan called “ NTRootkit”.
8	**Mobile Malware**	In 2000,it was discovered by an antivirus lab in Russia.

Table 1.1: History of Malware

1.4. Life Cycle of Malware

“Figure 1.1” shows the lifecycle of a malware.

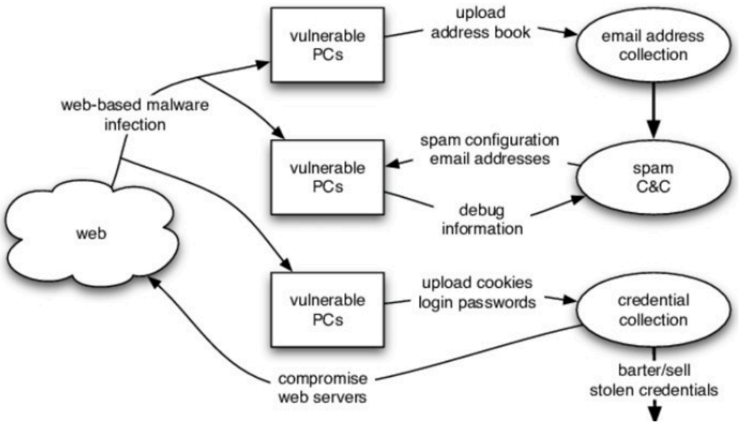


Figure 1.1: Life Cycle of Malware

1.5. Malware Analysis Techniques

Malware Analysis involves study of malicious files to understand certain details of malware, like malware behaviour, its development over time, and selected targets. Malware analysis results will allow cyber security experts to improve their approach to defend against malware strike. Malware inspection strategies are largely divided into three components "Figure 1.2": static analysis, dynamic analysis, and hybrid analysis. Furthermore, analysis based on memory is another very useful method of malware inspection.

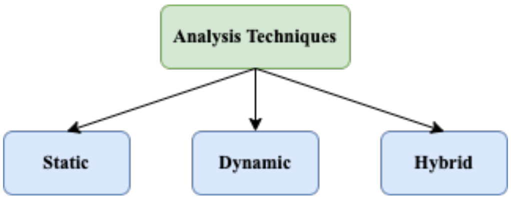


Figure 1.2 : Malware Analysis Techniques

1.5.1. Static Analysis

This approach applies to the analysis without running the Portable Executable Files (PE Files). To avoid being analysed, malware usually utilise binary packers, like "UPX" and "ASP Pack Shell". Before examining, a PE file requires to be unwrap and unzip. You can use a dismantler tool to decompile Windows executable files, such as "IDA Pro" and "OlleyDbg", which expose assembly instructions, provide malware knowledge, and take out sequence to recognise the intruder. In static analysis, the identification of pattern can be extracted, such as "Windows API" calls, string signature, control flow graph, opcode frequency and byte pattern n-grams. Almost all programs use "Windows API" which requires to contact the operating system. For example, in "OpenFileW" and "Kernel32.dll" is a "Windows API" that generate a latest file or opens a file which is previously created. API calls thus disclose the behaviour of programs and could be viewed as an important mark in the detection of malware.

1.5.2. Dynamic Analysis

It is also known as evaluating actions. In this we study the malware in a managed environment like virtual machine, simulator and imitator. In the digital world, the corrupted files must be examined for the easy explanation for that few malware is protected by anti-virtual machine and anti-imitator approach. The malicious file normally works when certain environments are detected and no malicious activity is seen. compared to static analysis dynamic analysis is more effective, since there is no need to test disassembled infected file. Additionally, complex detection is capable of detecting known and unknown malware.

1.5.3. Hybrid Analysis

Static analysis and dynamic analysis gather malware information from a hybrid scanning. Reliability researchers reap the welfare of all scanning, both static and dynamic, by using hybrid analysis. Therefore, the ability to correctly detect malicious programs is growing. The benefits and weaknesses of both analyses are their own. Compared with dynamic analysis, static analysis is inexpensive, quick and safer. However malware, evades this by using methods of obfuscation. Dynamic analysis on the other hand, is accurate and can solve methods of obfuscation. In addition, it is capable of detecting malware variants and families of unknown malware. Time intensive and resource-consuming

1.6. Malware Detection Techniques

Malware Detection methods are loosely split-up into two groups[12]: detection based on anomalies and detection based on signatures. In order to assess the maliciousness of a programmer under analysis, detection based on anomaly method uses consciousness of what account for the usual behaviour.

Detection based on specification is a particular form of detection based on anomaly. In order to assess the maliciousness of the programmer under investigation, approach based on specification use a guideline or rule set of what is legitimate behaviour. Programs that violate the specification are deemed to be irregular and typically malicious. In order to assess the malicious existence of the software under review, the detection based on signature uses the characterisation of what is perceived to be malicious.

As one would expect, this malicious activity characterisation or signature is the clue to the efficacy of a detection based on signature system. The connection between various forms of malware detection techniques is seen in “Figure 1.3”. One of three different methods may be used for each of the detection techniques: static, dynamic or hybrid. How a technique gathers information to detect malware describes a specific process or analysis of a technique based on anomaly or technique based on signature. Static analysis uses program (static)/process (dynamic) under inspection "PUIsyntax" or structural properties to identify its maliciousness. For instance, a static approach to detect based on signature would only leverage systemic knowledge to figure out the maliciousness, whereas a dynamic approach take advantage of PUI period data. In general, until the program under review is running, the static approach is designed to detect malware. Conversely, during execution of a program or after execution of a program, a dynamic approach aims to detect malicious behaviour.

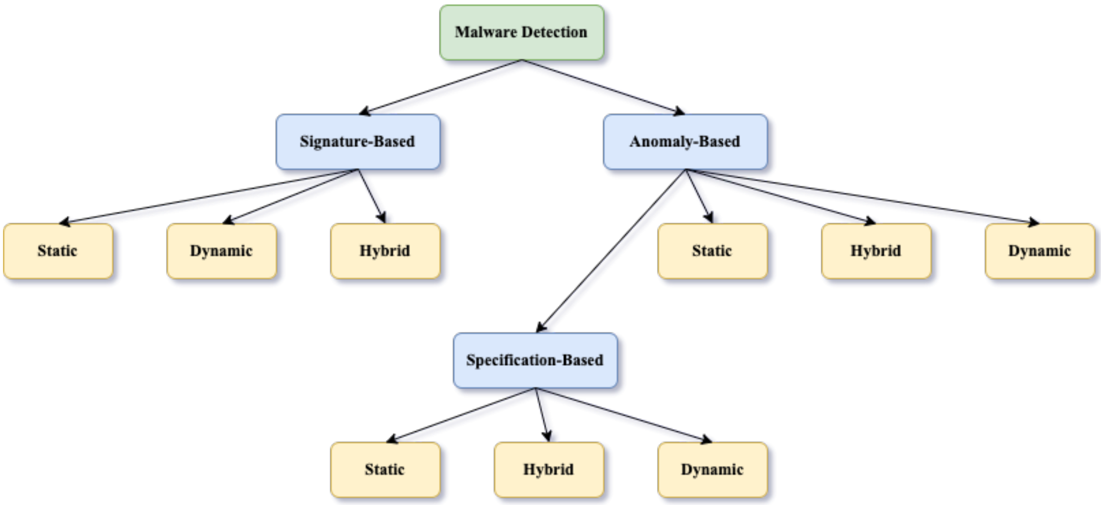


Figure 1.3 : Malware Detection Techniques

1.6.1. Detection Based on Anomaly

Detection based on anomaly usually takes place in two stages, first is a cycle of preparation and second is a process of detection. The detector is trying to learn about everyday acts during the training stage. In the course of preparation process, the identifier may grasp the behaviour of the horde or the PUI, or an amalgamation of both.

Potential to identify zero-day attacks is its main advantage of anomaly based detection. Zero-day accomplishments are explained by Weaver, et al.[11]. Zero day attacks, alike to zero-day strike, are attacks previously unknown to the malware identifier. The main elementary downside of this procedure are its elevated erroneous alarm rate and the convolution necessitate in identifying what characteristics should be grasp in the tutoring period.

i. Detection Based on Static Anomaly

In the case of static anomaly detection, the malicious code detection is based on the attribute of the file formation of the program under investigation. The key dominance of static detection based on anomaly is that it can be used to detect malware without allowing the host device to run malware that carries software.

ii. Detection Based on Dynamic Anomaly

For the identification of dynamic anomaly, the data obtained from the execution of the program is used to identify malicious code. During its execution, the identification stage observer the program under investigation, looking for unpredictability with what has been grasp during the tutoring stage.

1.6.2. Detection Based on Specification

Detection based on specification is a detection based anomaly system that seeks to illustrate the classic elevated erroneous alarm rate found with most detection based on anomaly. Detection based on specification recognition aims to predict the program or device parameters instead of attempting to guess the application or system implementation. The training stage in the detection based specification is the achievement of a set of rules that states all the rational manner that any program can display for the the program being inspected. The key downside too detection based on specification is that the wide spectrum of valid activities that the system will display is also difficult to describe thoroughly and precisely. One might imagine that even with a relatively complicated structure, the absolute and exact description of its true behaviour can be troublesome. While it can be possible to articulate the specifications of the system in natural language, it is often difficult to communicate this in a manner that is suitable for the machine.

i. Detection Based on Static Specification

Detection based on static specification uses the systemic properties of the PUI during the detection stage to determine its maliciousness.

ii. Detection Based on Dynamic Specification

Approaches to assess the maliciousness of an executable are known as dynamic specification-based usage actions observed at run time.

1.6.3. Detection Based on Signature

Detection based on signature aims to imitate malware's malicious behaviour and uses this imitate in malware identification. Detection based on signature information is defined by the set of all these models. This malicious conduct model is sometimes referred to as the impression. Preferably, any malware displaying the malicious conduct stated by the impression should be able to recognise a signature. Signatures require an archive, much like any information that resides in huge amounts that needs storage. As it relates to malware detection, this data warehouse represents all of the information the impression based system has. When the procedure endeavour to determine whether the PUI holds a familiar impression, the repository is scanned. At present, in producing impressions that reflect the malicious activity shown by the program, we rely primarily on human expertise. If an impression has been developed, it is attached to the knowledge of the impression-based process.

One of the main drawback of detection based on signature is that it is unable to track zero-day attacks, an attack for which the registry does not have the corresponding impression.

i. Detection Based on Static Signature

Detection based on static signatures is described by examining the code pattern observation software that will disclose the malicious target of the application. The objective is to obtain a code that determines the actions of the program. A coherent analysis of this code provides an estimation of the execution behaviour of the executable under investigation. The sequence of the code can be represented in the Signatures.

ii. Detection Based on Dynamic Signature

Detection based on dynamic signatures is distinguished by the sole use of knowledge gathered during the PUI implementation to evaluate its maliciousness. detection based on dynamic signatures looks for activity patterns that would expose a program's true malicious intent.

1.7. File-Less Malware

File-less malware is being developed by attackers to make it very difficult for antivirus users to detect it. File-less malware is put down straight to the memory under the name of the hard disc file. Hacker also attempts to persist on the computer after putting down a malicious program or memory code. Device security mechanisms such as "PowerShell" and "Windows Management Instrumentation (WMI)" for the execution and delivery of file-less malware.

File-less malware leaves no sign of antivirus program identification, making it very difficult for antivirus software and security specialist to identify file-less malware strike. Since file-less malware does not involve the upload of a file, it is very difficult to stop, monitor, and delete the file.

"Figure 1.4” shows the working of file-less malware

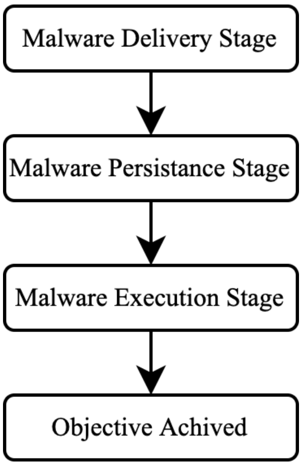


Figure 1.4 : File-less Malware Working

Delivery Stage: For making user to click the link which are attached with the phishing e-mail, file-less attacks social engineering is being used. hiding in a flash on a website or in a document created by an authorised program is a malicious script. Attackers use trustworthy tools and they want to make sure that no files or activities are checked by traditional security detection technology.

Persistence Stage: Many file-less malware methods are short-lived and in order to gain persistence, attackers employ a range of evasive techniques. Storing malicious code, like WMI store, SQL tables, and Windows registry, in odd locations connected with the basic service or operating system. Direct Malicious Script passed to PowerShell as a command line, kept in the registry

and run by the OS planner.

Execution Stage: In specific, the malware relies on internal windows such as PowerShell, JavaScript, and Macro Contract Execution and other approved Windows executable tools when all permanent structures are in place.

1.8. Analysis of File-Less Malware

There are three types of file-less malware, i.e. malware which reside in memory, malware which reside in windows registry, and root kit file-less malware. File-less ransomware can obscure its location and make it difficult to detect both conventional antivirus tools and security experts.

1.8.1. Malware Reside in Memory

The malware that lives entirely inside main memory, avoiding the operating system file systems. So they hide inside and stay in the approved process files or authentic windows data until they are activated.

Technique: Poweliks uses computer registries to attain immortality, rest uncharted. Key is run after getting two registries. First of all in the case of a JavaScript program, the cipher details drafted under merit is an autorun arrival that study and decipher ciphered JavaScript file.

1.8.2. Malware Reside in Windows Registry

Library which stores all the low-level configuration of the operating system, and few of the most important applications is known as registry. The malware writers managed to encrypt the complete malevolent code in the register to make it undetected. Some operating systems can use the registry

to make use of a thumbnail cache for persistence. After completion of mischievous task file gets destroyed automatically.

Technique: The JavaScript file is added to the registry and the approved Windows file, “mshta.exe", is carry out using WMI rather than “mshtml.dll:”.

1.8.3. Root Kits File-less Malware

An attacker will run a malware like this after he gets it. executive level right to conceal the malicious code in the Windows operating system kernel. It’s though this isn't a 100 percent file-less virus, either it works here.

1.9. File-Less Malware Detection Techniques

PowerShell and WMI can be used to conduct surveillance, persistence, lateral flight, remote command processing, and data transfer in the event of file-less ransomware, making it difficult to track down evidence left behind after a hack. In their study, the researchers[13] proposed several approaches for detecting malware infection. To accurately identify those attacks, the first two techniques require a security specialist to review the details recommended by the researcher, while the third approach is only a theory that has yet to be applied.

1.9.1. Detection based on System Behaviour

In order to identify file-less ransomware, the system needs to note two things. First, processes that have extended rights after residing in memory and second, monitor security events for program execution via command-line console or PowerShell.

1.9.2. Detection based on Rules

Many malicious programs spread over the Internet via the attacker’s target or botnet to locate a vulnerable victim are loaded with “Microsoft Office applications such as winword.exe, excel.exe, and powerpnt.exe". In addition, it could be possible to detect certain programs that trigger

"cmd.exe" or "powershell.exe". The observation implement may therefore operate under a regulation that can differentiate between a gentle process and a malevolent process.

1.9.3. Detection based on Attack Behaviour

The architecture can be designed in the client-server paradigm, where all client endpoints are installed, and the cloud servers. The method is categorised in three levels, such as event tracking, event marking and event learning. In this process, the customer will collect all events cause by the host system to control the full flow of the pursuit. The customer also allocate the progress tag of the intruder to each event in an appropriate way. Finally, many analytic engines on the server operate on tagged events provided by the client to detect suspicious activity on the host computer.

Labeled affair will be unprocessed information for illumination algorithms and study of pattern actions to deter or identify malicious action by accord between incident sources.

CHAPTER 2

RELATED WORK

This paper [5] addresses how malware is a threat to information security and discusses the common file-based malware like viruses, worms, Trojans and also discusses the emergence of a new type of malware i.e. file-less malware which is different from file-based malware. how it uses shielding techniques to hide from the antivirus software. file-less malware runs completely on the memory and leaves a tiny footprint in the host computer. Also, tell us about how attackers are taking advantage of system administration tools like power-shell and windows management instrumentation (WML) to execute the malware.

This paper [6] discusses how malware analysis has always been an important topic for research since the early days and how different malware is developed and how they defend themselves from the anti-virus. How they are taking advantage of windows management instrumentation (WMI) or PowerShell. they discussed categories of file-less malware i.e. RAM-Resident file-less malware and Script-based file-less malware. They have also discussed file-less malware detection techniques like sandboxing, execution emulation, heuristics, and Yara. In the end, they have concluded that preventing the file-less malware attack requires multilayers and an integrated approach that covers the entire life-cycle of file-less malware.

In this [7] David Patten discussed the history of the malware and how file-less malware is evolved from the previous file-based malware. it does not depend on writing complex code instead it depends on the program which is installed on the computer and uses it to retrieve the information or cause damage. In the end, David Patten concluded that file-less malware is a recent threat and in order to overcome this research in this field should be conducted.

In paper [8] they have explained that file-less malware is of a different category that is dependent upon memory for running. Due to which it makes impossible to detect them with the conventional signature-based method or heuristics-based detection. In the end they have concluded that file based malware and file-less malware both are different types of attack. While there are multiple techniques present to detect the file based malware but due to lack of information of file-less malware it is a challenging task to detect it. So instead of detecting the file less malware precautionary methods should be taken to avoid the attack.

In this [9] paper they have discussed the different types of malware and detection methods i.e Signature-based detection and heuristic-based detection. In signature-based detection, they have explained how the antivirus software matches the signature code with the detected file to capture the malware and in heuristic-based method(behavior-based) firstly it learns about the files in the learning phase after that it labeled them as malicious or not. They have also talked about the malware analysis techniques i.e static analysis, dynamic analysis, and hybrid analysis.

In this [10] paper Krishna B L discusses the file-less ransomware which is a combination of file- less malware and ransomware and due to file-less malware nature how difficult it is to detect or defend against these types of attacks. In this paper, he has also discussed the mechanism of the file-less ransomware and how it works internally and discusses the different types of file-less ransomware families like PoshCoder/PowerWare, UIWIX and synAck. for detecting the file-less ransomware only checking if there are any changes in the registry will be helpful and in the end, he concluded that security is lore and it is 100 percentage impossible.

In this [11] paper they discussed about the file-less malware in detail and explained how the file- less malware gets executed on the host system and takes leverage of operating system tools to run in memory without getting detected from the traditional antivirus software and also explained in brief about the difference between file based malware and file less malware. They have also discussed about detection techniques for the file-less malware like detection by monitoring the system behaviour, rule-based detection and by learning the attack behaviour. they also proposed a model for incident response in which a cyber-security team has to follow these steps:

I. Preparation: It involves creating a response of cyber security expert.

II. Detection: In this step they have to confirm that attack has taken place.

III. Collection: In this they have to collect the evidence from compromised system.

IV. Investigation: In this step team will try to isolate the compromised system to avoid the
further spread of malware.

V. Incident Closure: In this the complete their report and improves the future preparation.

In this [12] paper explained that from past few years how malware is developing and spreading. There are some methods to detect them like memory introspection, process activity monitoring and application enforcement. But all their method takes time. Therefore, they have proposed a method for anomaly detection. So they have used OSC a modified perceptron algorithm, which add an extra layer in the training algorithm to confirm that almost all samples labelled as benign are properly categorised. They have taken initial data of approximately 500k commands from the Bit-defender cyber threat lab and they maintain the ratio of 5:1 between the command i.e. 5 commands are clear and one command is not clean. After training the model with unique features they got better detection rate than the simple model.

In this [13] paper, the author suggest a PowerShell method called PSDEM. Large number of cyber attacker uses power shell in which malware is deliver by a spam mail, with the help of Microsoft word to infect the victim's system.In suggested method there are two layers of de-obfuscation to generate rhea original PowerShell script. PowerShell scripts are extracted from the obfuscated document in first layer and in second layer contains de-obfuscation scripts. PSDEM help in increasing the accuracy for exam in the malicious scripts present in MS word Document.

In this [14] paper, author suggested a tool which can be used for analysing the file-less malware or volatile threats which directly works in memory.

They have created a GUI tool which works in following steps and Figure 2.1 shows the methodology of tool.

i. Tool ask for the user login id and password for authentication

ii. After login there are two options

  a. User can take capture the memory image

  b. Can analyse the already captured RAM

iii. After selecting option 2 user need to select memory image path

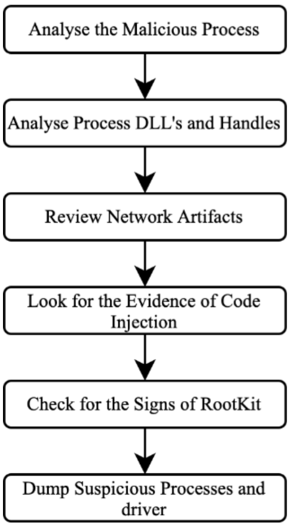

Analyse the Malicious Process
Analyse Process DLL's and Handles
Review Network Artifacts
Look for the Evidence of Code Injection
Check for the Signs of RootKit
Dump Suspicious Processes and driver

Figure 2.1 : Methodology of Tool

iv. Now tool runs the volatility plugin

v. After selecting the profile user can further analyse.

vi. They concluded that this GUI based tool can help in detecting the advance malware like file less malware and other similar volatile threats.

CHAPTER 3
PROPOSED METHODOLOGY

3.1 Tools Used

In this section, we give an overview of all the software requirements that were necessary for implementing our work, such as the programming language(s), the imported libraries etc.

3.1.1 Programming Platform: Python 3.6

Python is widely used for programming. Developed by a programmer named Guido van Rossum in 1991, it has been extensively developed and used for many large-scale projects. Also, it is an interpreted language. An interpreted language is a high-level language run and executed by an interpreter (a program which converts the high-level language to machine code and then executing) on the go; it processes the program a little at a time. It involves programming at high level, great for beginners, and a programmer can focus on what to do, and less on how to do that, due to its easy syntax and huge variety of import libraries.

3.1.2 Libraries Used

Multiple libraries and open-source packages, that are required to implement the framework, involves python's open-source Keras as backend and other libraries useful for machine learning applications such as scikit and pandas.

i. Pandas

Pandas is a basic tool for our data. Pandas familiarises you with your data by cleaning, transforming, and analysing it. Pandas help us to know our data as we can clean, transform, and analyses the data. Like, if we have to use a dataset in CSV format, Pandas extracts data from the CSV file to a Data-Frame (table). Then we can perform things like:

a. Calculation of statistics:

- Average, median, maximum and minimum of columns.
- Correlation of columns
- Distribution of data

b. Removing missing values and filtering rows and columns using some criteria.

c. Visualisation of data like plot bars, lines, histograms, bubbles, etc.

d. Storing the clean and transformed data into a CSV file or other form of a database.

ii. Scikit-Learn

Many supervised and unsupervised learning algorithms are provided by Scikit-learn with a good interface in python. It focuses on robustness and support requirements in production systems. It implies focus on ease of use, quality coding, performance, documentation, etc.

iii. Mat-plot Lib

It gives an outstanding visualisation in python for 2D plots. Matplotlib is built on NumPy arrays for multi-platform visualisation and uses SciPy stack which is for broader use. It was presented by John Hunter in 2002. Visualisations greatest advantage is that we can visually see large data in easy-to-understand graphs, etc. It has plots like line, bar, scatter, histogram, etc.

iv. Pickle

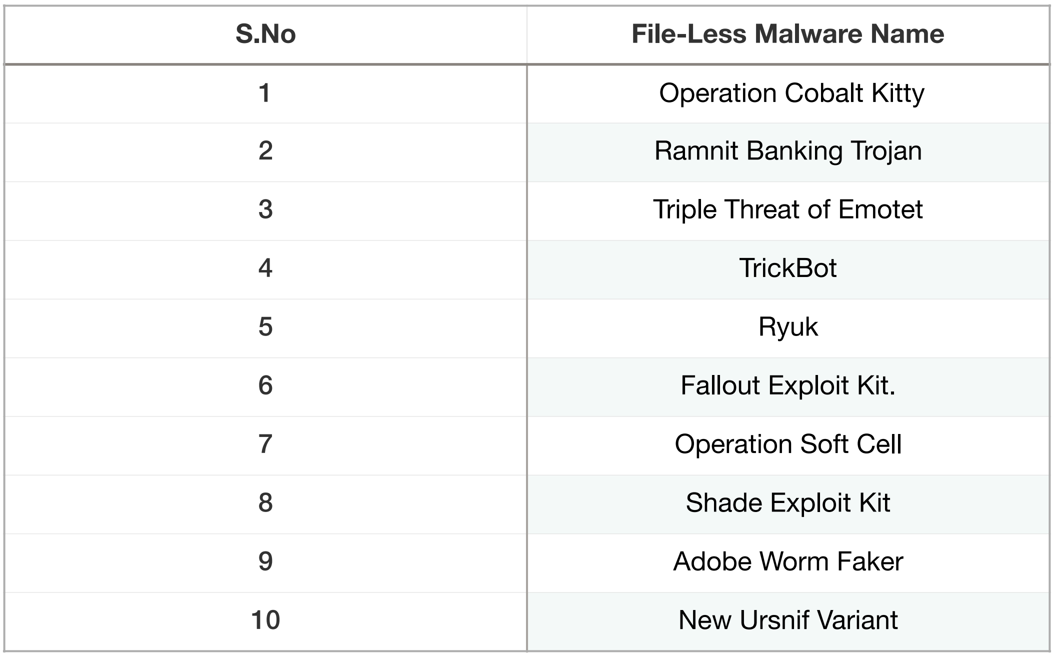
Pickle is a module which helps to change or modulate the object structures in a way that is friendly to python and which makes it easy to work upon. All types of python objects can be pickled, with the help of pickle library and then written and stored on the disk.

3.2 Dataset Collection and Preparation

As File-Less malware is relatively a new topic in research area and also there are very few dataset present currently and they are also not available for publicly so we have created the data set of file-less malware.

Collecting the data-set was not enough as we are going to use machine learning techniques for the detection of file-less malware so preprocessing is also required.

Initially the dataset contained the file name and its SHA-1 hash value we added the file name and SHA-1 hash value of non malicious file present in windows directory.



S.No	File-Less Malware Name
1	Operation Cobalt Kitty
2	Ramnit Banking Trojan
3	Triple Threat of Emotet
4	TrickBot
5	Ryuk
6	Fallout Exploit Kit.
7	Operation Soft Cell
8	Shade Exploit Kit
9	Adobe Worm Faker
10	New Ursnif Variant

Table 3.1 : File-Less Malware Present in Dataset

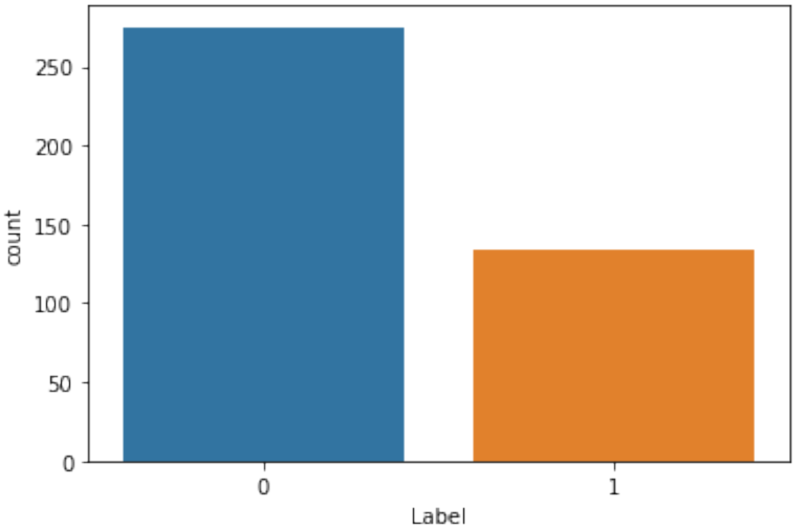
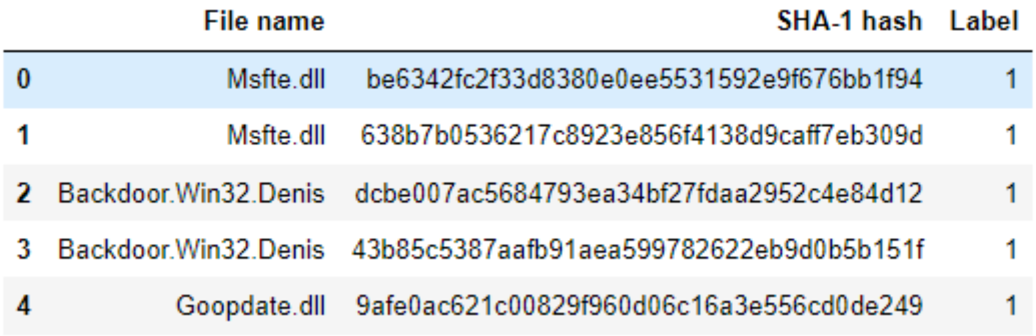


Figure 3.1 : Size of the Dataset

After that we added a 1 extra column named as labels and filled it with ‘1’ and ‘0’ where ‘1’ stands for malicious and ‘0’ stands for non malicious file “Figure 3.1” shows the initial size of dataset “Figure 3.2” shows initial dataset.



	File name	SHA-1 hash	Label
0	Msfte.dll	be6342fc2f33d8380e0ee5531592e9f676bb1f94	1
1	Msfte.dll	638b7b0536217c8923e856f4138d9caff7eb309d	1
2	Backdoor.Win32.Denis	dcbe007ac5684793ea34bf27fdaa2952c4e84d12	1
3	Backdoor.Win32.Denis	43b85c5387aafb91aea599782622eb9d0b5b151f	1
4	Goopdate.dll	9afe0ac621c00829f960d06c16a3e556cd0de249	1

Figure 3.2 : Initial Dataset

We have used 4 different types of supervised learning algorithms for detection

S.No	Machine Learning Algorithm
1	Logistic Regression
2	K-Nearest Neighbour
3	Decision Tree
4	Support Vector Machine

Table 3.2 : List of Algorithm Used

3.3 Methodology Used

Moving on to how we have used the dataset, our approach can be divided into two steps: first is preprocessing and second steps is the prediction part "Figure 3.2" shows the workflow diagram.

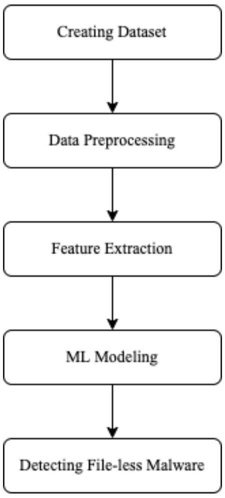


Figure 3.3 : Diagram of Proposed Method

In “Figure 3.1” it is visible that the ratio of malicious file vs non malicious file is not that great so we have applied smote algorithm over our dataset to improve the ratio of malicious file vs non malicious file “Figure 3.3”. show the ratio after applying smote algorithm.

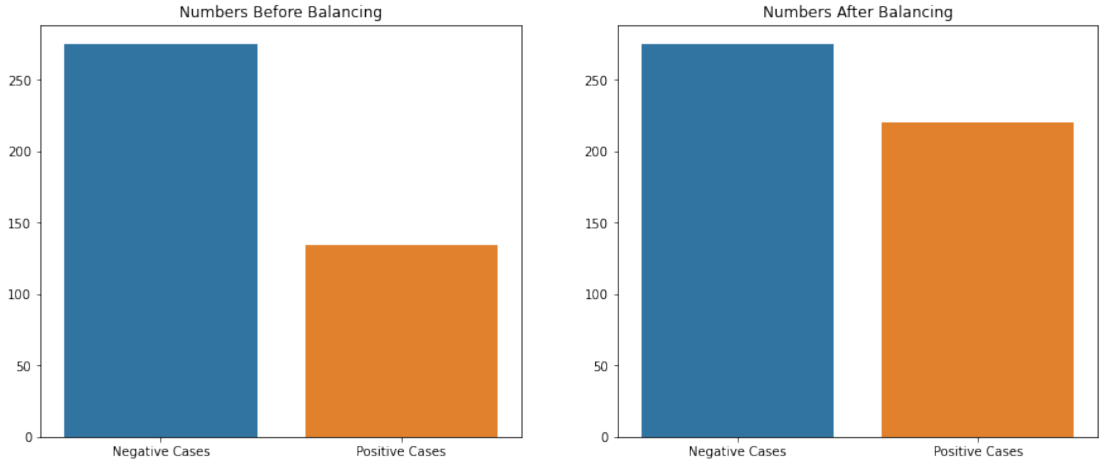
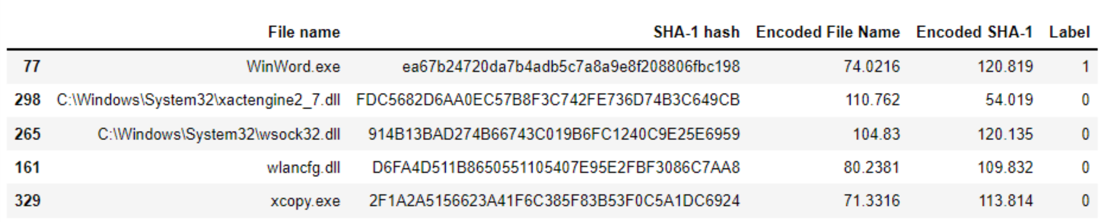


Figure 3.4 : Ratio After Applying Smote Algorithm

As we are using supervised learning algorithm for prediction input required by them needs to be in integer form but our dataset contains the string value. So we have designed a encoder which takes the string as an input and gives its integer value as an output “Figure 3.5” shows the final dataset after preprocessing.



	File name	SHA-1 hash	Encoded File Name	Encoded SHA-1	Label
77	WinWord.exe	ea67b24720da7b4adb5c7a8a9e8f208806fbc198	74.0216	120.819	1
298	C:\Windows\System32\xactengine2_7.dll	FDC5682D6AA0EC57B8F3C742FE736D74B3C649CB	110.762	54.019	0
265	C:\Windows\System32\wsock32.dll	914B13BAD274B66743C019B6FC1240C9E25E6959	104.83	120.135	0
161	wlancfg.dll	D6FA4D511B8650551105407E95E2FBF3086C7AA8	80.2381	109.832	0
329	xcopy.exe	2F1A2A5156623A41F6C385F83B53F0C5A1DC6924	71.3316	113.814	0

Figure 3.5 : Final Dataset

CHAPTER 4

RESULTS AND ANALYSIS

In this section we have presented a comparative description of the performance of the supervised learning algorithms over the File-less malware dataset. Each algorithm has its own specialty according to which it gives results. The accuracies after training are mentioned Table 4.1. It can be seen that the accuracy of K-Neared Neighbours is the highest.

S.NO	MACHINE LEARNING ALGORITHM	ACCURACY
1	Logistic Regression	74.32
2	K-Neared Neighbours	81.08
3	Decision Tree	78.37
4	Support Vector Machine	78.37

Table 4.1. : Accuracies

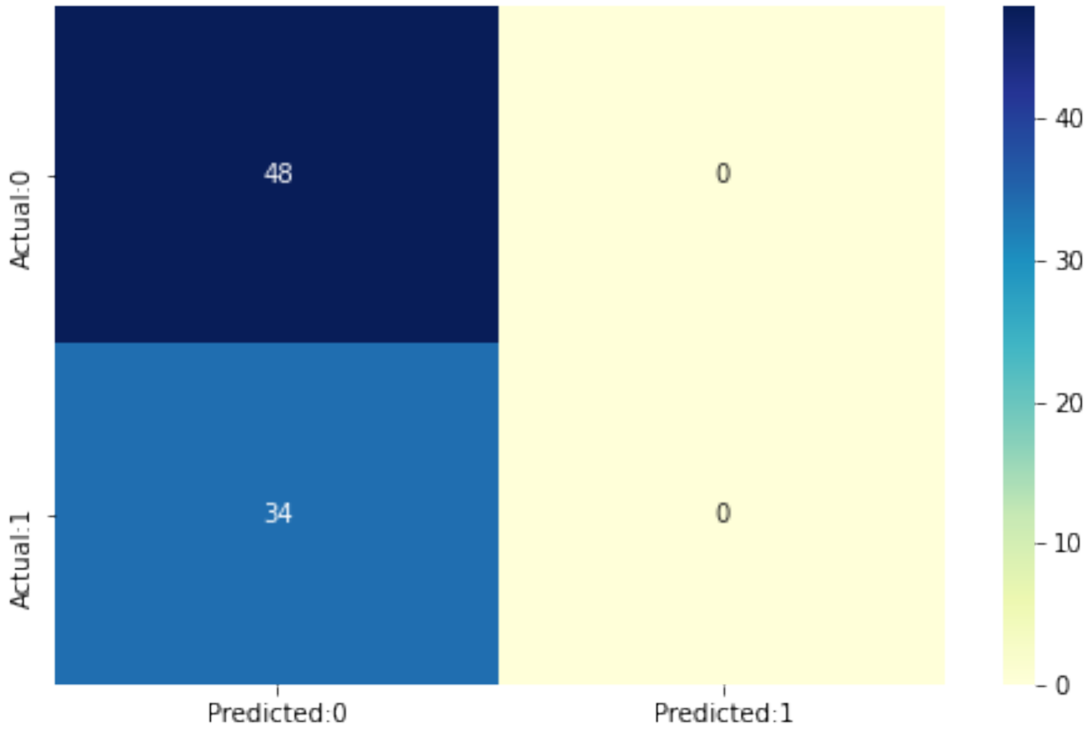


Figure 4.1. : Confusion Matrix of Logistic Regression

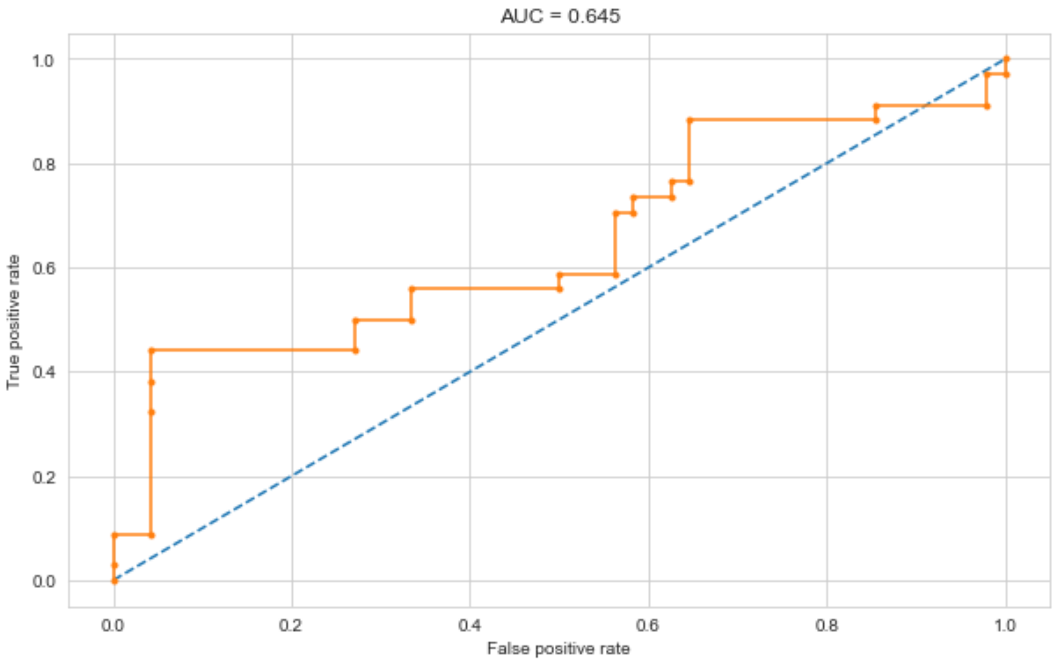


Figure 4.2. : AUC Graph of Logistic Regression

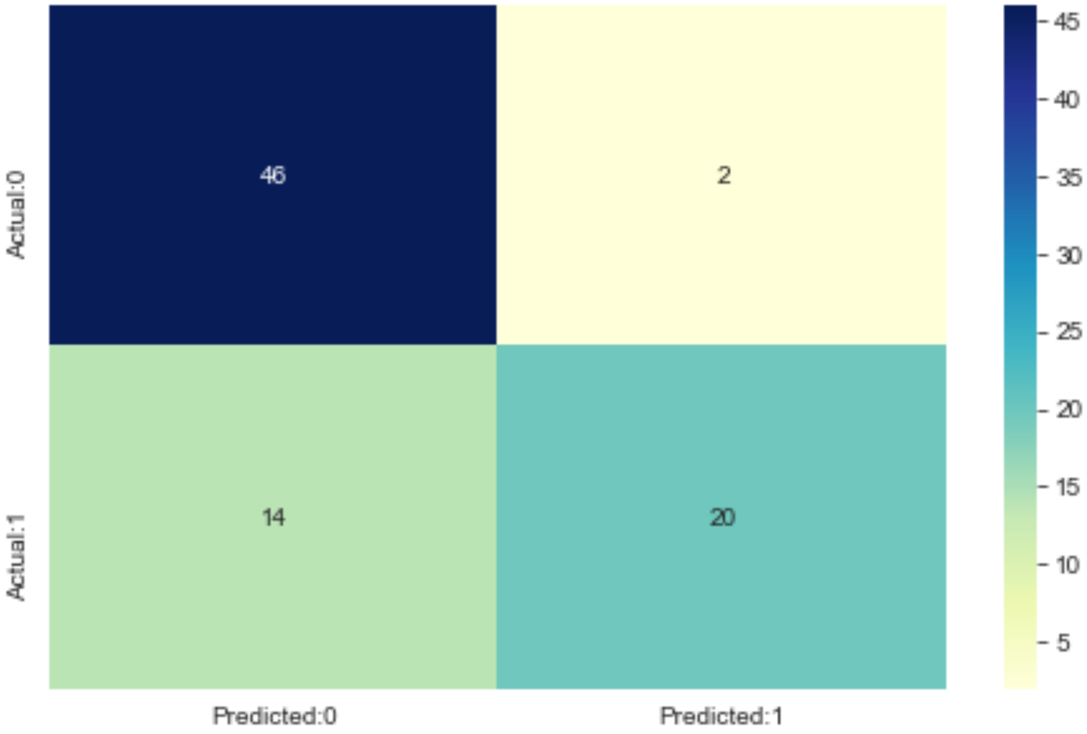


Figure 4.3. : Confusion Matrix of K-Nearest Neighbours

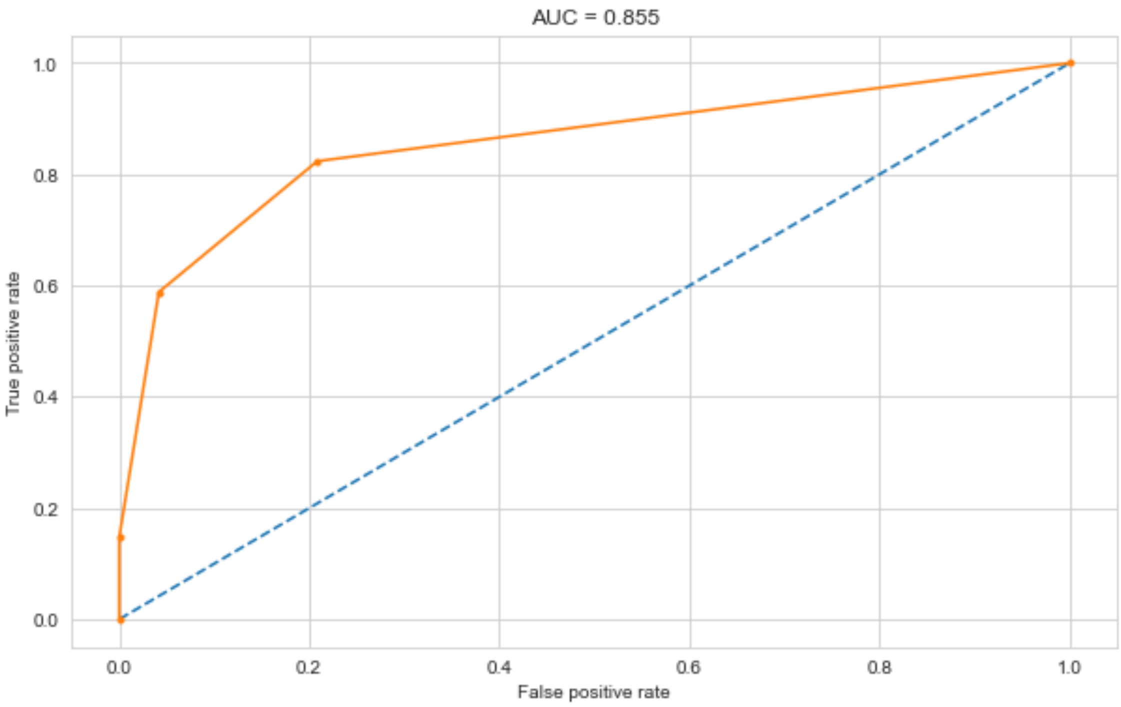


Figure 4.4. : AUC Graph of K-Nearest Neighbours

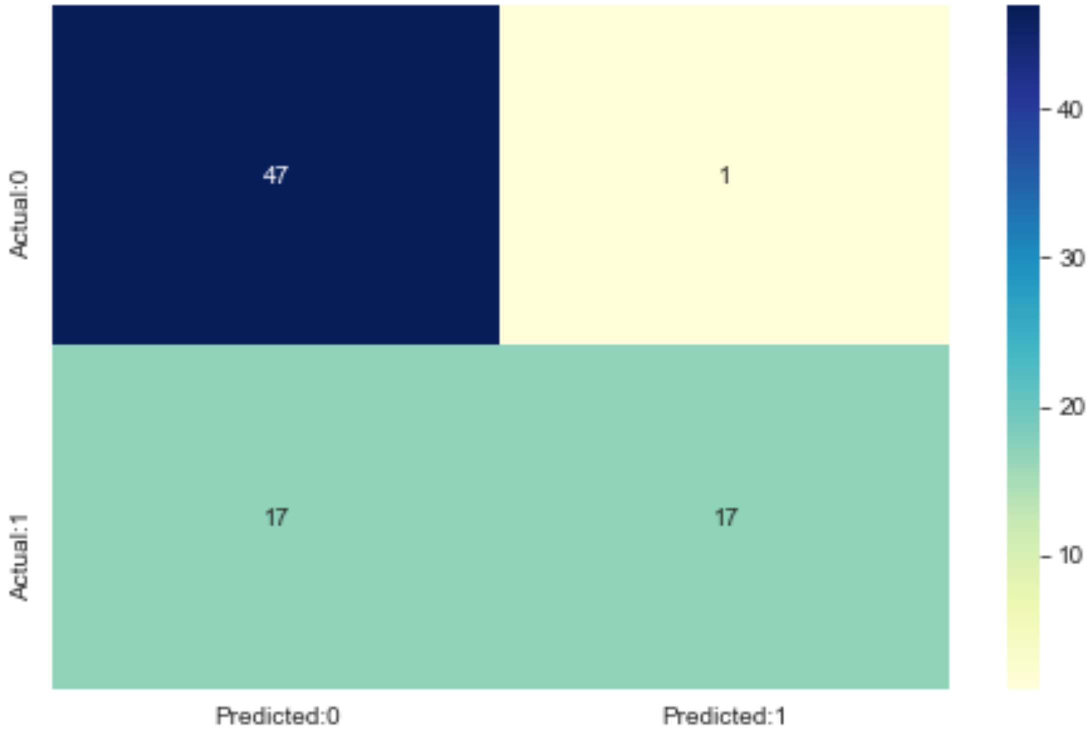


Figure 4.5. : Confusion Matrix of Decision Tree

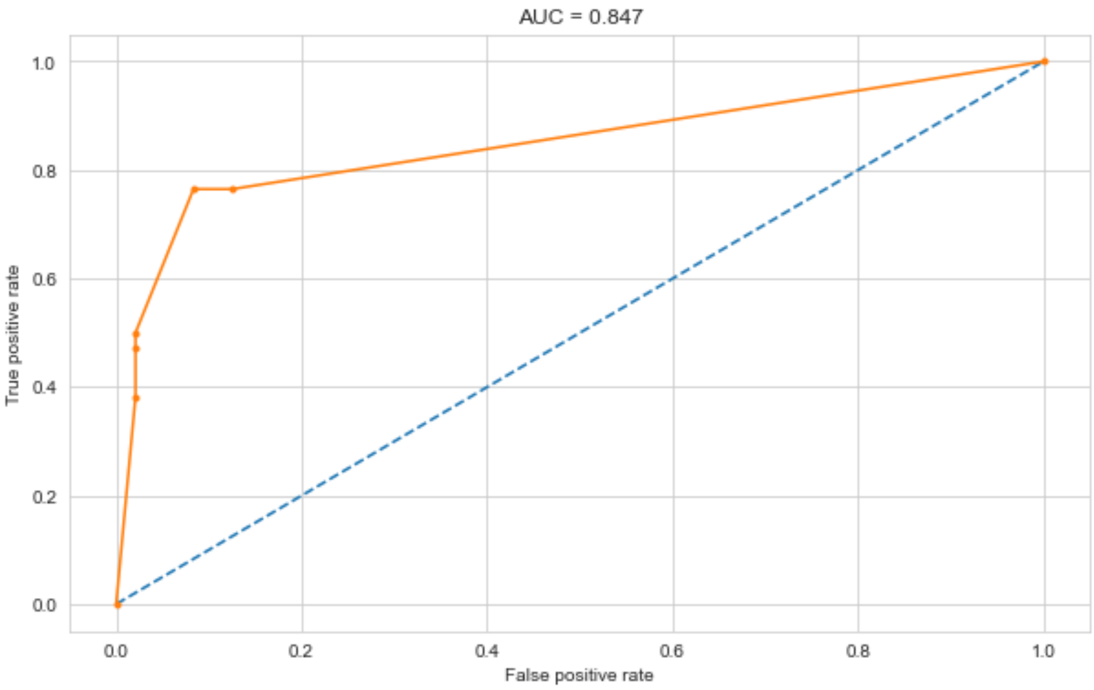


Figure 4.6. : AUC Graph of Decision Tree

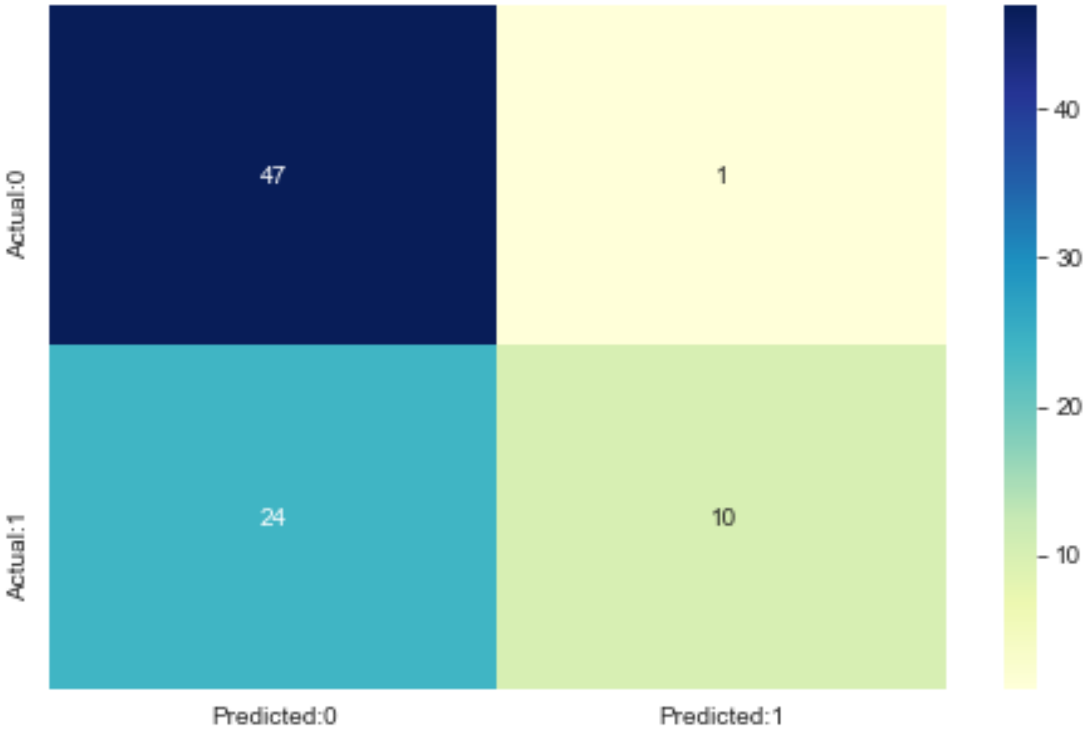


Figure 4.7. : Confusion Matrix of Support Vector Machine

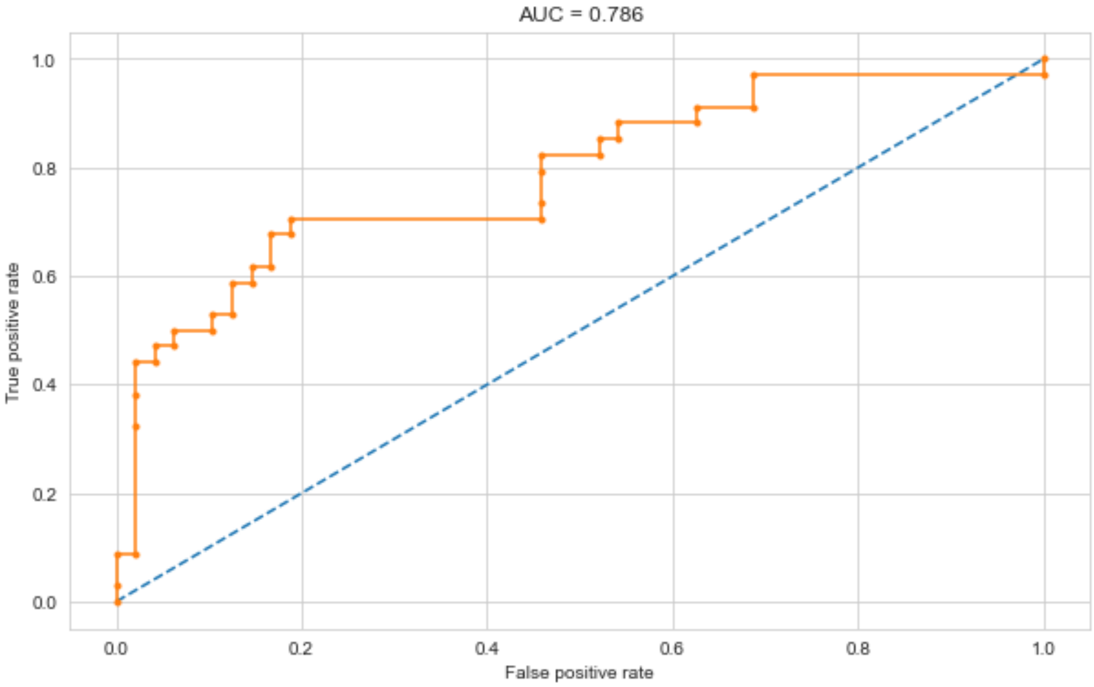


Figure 4.8. : AUC Graph of Support Vector Machine

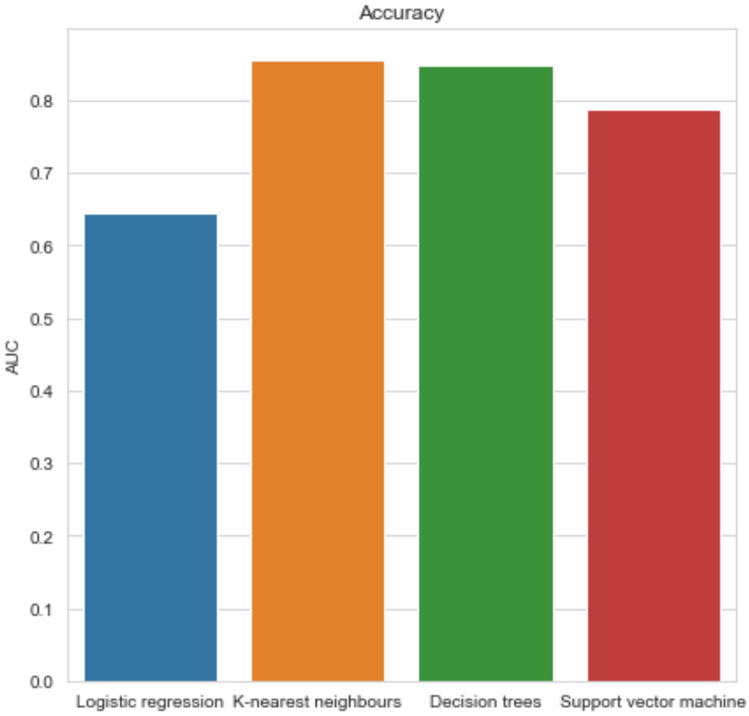


Figure 4.9. : Bar Graph of Accuracies

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

It is quite evident that the field of machine learning has a capability of creating a very powerful impact on the society. The algorithms used for file-less malware detection are accurate and robust enough to work in complex environments. Our efficient implementation utilises supervised learning for detection. Since our system has low processing power and we didn't have sufficient data, we couldn't run deep learning algorithm. We can increase the processing power but problem of limited dataset still remains. So, more research is still required in this area for collecting the data regarding file-less malware.

REFRENCES

[1] West-Brown, Molra J., Don Stikvoort, Klaus-Peter Kossakowski, Georgia Killcrece, and Robin Ruefle. *Handbook for computer security incident response teams (CSIRTs)*. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 2003.

[2] CERT/CC, Carnegie Mellon University. http: // www. cert. org/ stats/ cert \ stats.html \#incidents , last updated: April 2006.

[3] Idika, Nwokedi, and Aditya P. Mathur. "A survey of malware detection techniques." *Purdue University* 48, no. 2 (2007).

[4] Khushali, Vala. "A Review on Fileless Malware Analysis Techniques." *International Journal of Engineering Research and Technology* 9 (2020).

[5] Sanjay, B. N., D. C. Rakshith, R. B. Akash, and Vinay V. Hegde. "An approach to detect fileless malware and defend its evasive mechanisms." In *2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS)*, pp. 234-239. IEEE, 2018.

[6] Patten, David. "The evolution to fileless malware." *Retrieved from* (2017).

[7] Alzuri, Alain, David Andrade, Yadelis Nunez Escobar, and Brian M. Zamora. "The Growth of Fileless Malware." (2018).

[8] Sihwail, Rami, Khairuddin Omar, and KA Zainol Ariffin. "A survey on malware analysis techniques: Static, dynamic, hybrid and memory analysis." *Int. J. Adv. Sci. Eng. Inf. Technol* 8, no. 4-2 (2018): 1662-1671.

[9] Krishna, B. L. "Comparative Study of Fileless Ransomware." (2020).

[10] Kumar, Sushil. "An emerging threat Fileless malware: a survey and research challenges." *Cybersecurity* 3, no. 1 (2020): 1-12.

[11] Bucevschi, Alexandru Gabriel, Gheorghe Balan, and Dumitru Bogdan Prelipcean. "Preventing File-Less Attacks with Machine Learning Techniques." In *2019 21st International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, pp. 248-252. IEEE, 2019.

[12] Liu, Chao, Bin Xia, Min Yu, and Yunzheng Liu. "PSDEM: A Feasible De-Obfuscation Method for Malicious PowerShell Detection." In *2018 IEEE Symposium on Computers and Communications (ISCC)*, pp. 825-831. IEEE, 2018.

[13] Gadgil, Priya, and Sangeeta Nagpure. "Analysis Of Advanced Volatile Threats Using Memory Forensics." In *Proceedings 2019: Conference on Technologies for Future Cities (CTFC)*. 2019.

[14] Nahmias, Daniel, Aviad Cohen, Nir Nissim, and Yuval Elovici. "Trustsign: trusted malware signature generation in private clouds using deep feature transfer learning." In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-8. IEEE, 2019.

[15] Weaver, Nicholas, Vern Paxson, Stuart Staniford, and Robert Cunningham. "A taxonomy of computer worms." In *Proceedings of the 2003 ACM workshop on Rapid Malcode*, pp. 11-18. 2003.