Survey on Malware Analysis using Machine Learning
20BCY10075

bhavya.guglani2020

July 2022

1 Introduction

Technology has the potential to be great. When used properly, cutting-edge digital solutions enable us to improve customer engagement, automate the purchasing and billing processes, and boost the effectiveness of our internal and audit communications. These new solutions are producing new risks and problems for organisations to manage and overcome, just like all newly developing technologies do. These technologies offer great benefits to society while, at the same time, imposing grave risks ranging from malicious use to exacerbating socioeconomic inequalities.One of the main threat today is malware .

Malicious or malware are some programs transferred in computers while the owner is unaware of them and created to harm, interrupt or damage computers, networks or files. Malware exists in different forms. Such programs fall mainly into the following classes: Viruses, Worms, Trojans, Rootkits, Spyware, Adware, Sniffers, Keyloggers, Spam, Ransomware. Many of them exist in multiple classes, but they are not mutually exclusive.

Malware has been around since the dawn of computing. The first known virus, the Creeper worm, was an experimental self-replicating program that spread to remote systems first appeared in the 1970s. Elk Cloner, a boot sector virus that targets Apply II computers, emerged in the early 1980s. This led to the new beginnings of a huge industry.[3] Malware threat continue to spread vertically (number and volume) and horizontally (type and function) due to opportunities presented by technological advances. The internet, social networks, smartphones, IoT devices, and more create intelligent and sophisticated malware. In recent years, ransomware and crypto mining malware has emerged as the most prolific species, with Cryptoloot using the victim's computing power to mine crypto currencies without their knowledge. Attackers use malware to target PCs, mobile devices, cloud storage systems, supervisory control and data acquisition systems (SCADA), Internet of Things (IoT) networks, and even big data platforms.

New types of malicious code are constantly changing their signatures, so static methods are not suitable for detecting them. Over the last two decades, the emergence of machine learning techniques has greatly contributed to new malware detection due to their generalizability.

Machine learning models are based on two stages: training and prediction. The training stage relies on a training dataset that contain samples of benign and malicious files. It involves three phases. The first phase is to extract a large number of features from various files in the training data set. The second phase consists of rejecting non-pertinent features based on appropriate selection techniques. The third phase consists of using one or more classification models that will learn to distinguish between malicious and benign files. These models subsequently become able to give accurate predictions dealing with new executable files in the prediction stage. The choice of both appropriate input features and classification model leads to the improvement of prediction rates.

Either signature-based or behavior-based approaches are used to identify malware. The obfuscated virus may readily bypass the rapid and effective signature-based malware detection methods. To the contrary, techniques that are behavior-based are more resistant to the obfuscation approach. However, behavior-based approaches take a lot of time.[17]

There is a never-ending race between malware developers and developers.Both research communities operate in parallel, one is developing the malware detection system and other is designing the malicious software to compromise the detection software for attacking the computer and networks resources.The malware analyst aim to develop a technique which could not only detect the attack of the malware but also devise a solution for it. Traditionally, networks used Web Application Firewalls (WAFs), Intrusion Prevention Systems (IPS), Radware DefensePro and AppWall to protect networks and applications from threats such as this type of malware. Along with brute force attacks such as DDoS, malware poses a dangerous threat to network security. Keeping users up to date with the latest protection techniques and threats is difficult, so robust network-layer security protocols are required. In this paper we propose a study of analysis of different machine learning techniques used in malware detection. Below is the contribution of the paper:

- This study examines the development and current situation of systems that can detect malware.
- A comparison is drawn between different machine learning algorithms.
- Discussion between different machine learning classifier is done on this research paper.

2 Related Works

In their work from 2001, Schultz et al.[2] employed machine learning to detect malware. They used the signature extraction method as a starting point. Following an analysis of all the files in the training set, byte sequences that were

present in the malicious files but absent from the benign ones were taken to create the signatures. These byte sequences were later combined to provide a distinctive signature for each infected file. Bazrafshan et al [20] focuses on malware detection and identifies three main malware detection methods, i.e. based on signatures, behavior and heuristics, the latter also using machine techniques learn automation. They also define feature classes used by heuristics that have been tested for malware detection, i.e. API calls, control flow graphs, n-grams, optical codes, and associative features.

In the year 2007, Danial Bilar [16] introduced opcode as a malware detector, to examine opcodes frequency distribution in malicious and non-malicious files.Ye et al.[9] examines various aspects of the malware detection process, focusing on feature extraction/selection and classification/clustering algorithms.

Although the document does not go into detail about the framework itself, it does provide a detailed description of the use of machine learning algorithms to detect infected operating systems. Tang et al.[7] showed how a "clean" state profile can be generated using unsupervised learning algorithms and that deviations from this profile can be used as a tool for detecting software. malware (unnamed tool). technique using graphs of API calls. .Ghabir et [18] proposes a technique for Advanced Persistent Threat (APT) detection using an API call. In this technique, the author uses KNN, SVM and a synthetic machine learning algorithm to train the classifiers. This recommended technique shows an accuracy of 8.8 in detecting APT malware. Then, different MR SVM, MRED SVM and Fisher SVM machine learning algorithms were applied to the over-reduced features and produced an accuracy of 99.9. The author made the comparison between static and dynamic API calls using machine learning algorithms. After exhaustive search investigation, it was found that these articles use different feature processing techniques and machine learning algorithms to develop the malware classifier. N. Usman, S. Usman, F. Khan et al. [21] a feature modeling with data mining technique is proposed that identifies vulnerabilities to the websites and assigns a quantitative score to the vulnerability. Use a tool called Malware Code Finder to detect malicious code hosted on suspicious websites.

Hossein Sayadi et al.[10] Hardware-assisted malware detection (HMD) techniques that use hardware-related background information have shown promising results, reducing detection latency by aligning the feature space with a hardware feature counter. Houman et al. [12]Take an in-depth look at different ML classifiers of different types to examine each model's effectiveness in detecting malware. Additionally, we examine the effectiveness of various ensemble learning techniques in increasing the accuracy and performance of malware detectors. Tinoosh Mohsenin et al. [11] proposes a 2SMaRT 2s to not only distinguish malware from benign applications, but also to identify the class of malware at runtime using appropriate low-level functions.

Salehi et al [13] implemented deep analysis API calls using past arguments to analyze and classify a large number of malware. Bayer et al. [5] proposed a clustering method to reduce the complexity of the distance calculation from n2 to n 2/2. Owezarski [6] proposed an automated method for the classification and characterization of anomalies on the unattended network. This investigation

can detect DoS, DDoS and Port Scan attacks. Halta [6] presents an automatic detection system for bots infected with BFH (Bot Through Honeypot Finder).

Heuristic methods of static malware detection have been used in the past and are considered to be one of the first jobs in this area. Ashu et al.[14] applied grouping by file size and opcode frequency as functions to improve average accuracy with various machine learning models using the Malicia dataset. Drew [1] et al used Strand, a gene sequence classification that provides a robust classification strategy that can easily handle unstructured data with any text, such as source code or machine code. Popov [1] suggested using word2vec, a popular tool recently developed for analyzing natural language texts, for malware classification. Word2vec generates word embeds from malicious opcodes after feeding machine instructions into capstone disassembler and generating opcodes.

Much of the research uses neural networks for malware detection and analysis. Increasing the number of hidden layers in the neural network did not bring significant gains in accuracy. Decomposition of obfuscated binaries may not provide satisfactory inputs for classification. Vinayakumar R[22] describe the major issue with the classical machine learning based malware detection system is that they rely on the feature engineering, feature learning and feature representation techniques that require an extensive domain level knowledge.

Santos, Devesa, Brezo, Nieves, and Bringas (2013)[4] introduced a static-dynamic approach to detect the malwares by using the machine learning approaches. Here, information from execution traces, including frequency of occurrence, is integrated with the help of malware detectors. Additionally, rules are categorized into the following groups: File, Protection, Persistence, Network, Error, Process, and System Information. [c]—l—l—l—l—l—l—l—l—l—l—l—

Author, Year	**Key contribution**	**P1(Algorithm used)**	**P2 (Signature based)**	**P3(PE File)**	**P4(Accuracy)**	**P5(Feature Extraction)**	**P6(Use of Opcodes)**	**P7(Data Availablity)**	**P8(Complex Process)**
Schultz 2001	Employ machine learning technique to detect malware using signature extraction method	Yes	No	No	Yes	No	No	No	Yes

Danial Bilar 2007	Introduced opcode as a malware detector, to examine opcodes frequency distribution in malicious and non-malicious files	Yes	No	No	No	Yes	No	Yes	Yes
Ye 2007	Examines various aspects of the malware detection process, focusing on feature extraction	Yes	Yes	No	No	Yes	No	Yes	No
Bazra -fshan 2013	Bazrafshan focus on malware detection and identify three main methods for detecting malicious software	Yes	Yes	Yes	No	No	No	Yes	No
Wang 2009	Proposed a SVM based malware detect -ion method for detecting new PE malware	Yes	No	Yes	Yes	No	No	Yes	Yes
Shabtai 2009	Shabtai provide a taxonomy for malware detection using machine learning algorithms	Yes	No	Yes	Yes	Yes	No	Yes	No
Balda -ngo -mbo 2013	suggested extraction methods depending on DLLs, PE headers, and API functions	Yes	No	Yes	Yes	Yes	No	No	Yes
Oweza -rski 2014	Proposed an automated method for anomalies classification and characterization	No	No	No	Yes	Yes	No	No	Yes
Ashu 2014	Applied file -size based grouping and opcode frequency	No	No	No	Yes	No	Yes	No	No

Shala -ginov 2015	Showed the possibility of malware detec -tion using specially-tuned Neuro-Fuz techn -ique	Yes	Yes	No	Yes	No	No	No	Yes	
Rieck 2013	proposed machine clustering tool which works on machine learning and it collects behaviour reports.	Yes	No	No	Yes	No	No	Yes	Yes	
Xiao and Mun -oz 2015	developed optim -ization methods of poisoning data for feature selection and logistic regression	Yes	No	No	Yes	Yes	No	Yes	Yes	
Vinay -aku -mar 2020	describe the major issue with the classical machine learning based malware detection system	Yes	No	No	Yes	Yes	No	Yes	Yes	
Tang 2014	Showed how a “clean” state profile can be gene -rated using unsupervised lear -ning algorithms	Yes	No	Yes	No	Yes	No	Yes	Yes	
Santos 2013	Introduced a static-dynamic approach to detect the malwares	Yes	No	No	Yes	No	No	Yes	Yes	
Halta 2018	Presents an automatic detection system for bots infected with BFH	Yes	No	No	Yes	No	No	No	Yes	

Drew 2019	used Strand, a gene sequence classif -ication that provides a robust classif -ication strategy that can easily handle unstruct -ured data	Yes	No	No	Yes	No	No	Yes	Yes
Popov 2017	proposed to use word2vec	Yes	Yes	No	Yes	No	Yes	No	Yes
Ghabir 2019	Proposes a technique Advanced Per -sistent Threat	Yes	No	No	Yes	Yes	No	Yes	Yes
Pirsco -veanu 2015	Pirscoveanu used the Cuckoo sandbox for capturing behavio -ural features of malware files.	Yes	No	No	Yes	Yes	No	No	Yes

comparison of different research papers

3 Methodology

3.1 Background

Malware is defined as software designed to infiltrate or it can tamper with a computer system without the owner's permission. Malware is actually a general definition, a type of computer threat. A simple classification of malware contains unique file infections and malware.The classification of malicious programs is based on their specific activity:Worms, backdoors, trojans, rootkits, spyware, adware, etc.

The catastrophic failure or dramatic slowdown of a single computer or network can be intentional or accidental. A virus or Trojan can remove critical system components and thereby disable the operating system, overload the network with a DDoS attack, or otherwise impair system functionality. Sometimes malware is incompatible with the software and hardware of the system it is running on, leading to server crashes or a drastic increase in spam traffic. In 1999, the CIH virus, also known as Chernobyl, disrupted all infected systems. It erased the flash BIOS data, making it impossible to boot the computer. Several

hundred thousand computers were victims of the "bomb". The cost of repairing these machines far exceeded the cost of buying a new laptop.

3.1.1 Types Of Malware

The following are the types of malware in the today's world:

TYPE	WHAT IT DOES
WORM	Spreads through the network by replicating itself
TROJAN HORSE	Disguises itself as desirable code
SPYWARE	Collects user activity data without their knowledge
RANSOMWARE	Disable victim's access to data until ransom is paid
ADWARE	Search unwanted Advertisements
FILELESS MALWARE	Makes changes to files that native to the OS
ROOTKITS	Gives hacker remote control of a victim's device
KEYLOGGERS	Monitors user's keystrokes
BOTS	Launches a broad flood of attacks

3.1.2 Malware Analysis

Understanding how a piece of malware works and its potential consequences is the process of malware analysis. It's important to understand that malware can have a wide range of activities and that malware code can vary greatly. These could manifest as Trojan horses, worms, malware, and viruses. Each sort of malware collects data about the infected device without the user's knowledge or consent.

3.1.3 Working Mechanism of Malware Analysis

The given below flowchart shows the general stages of the workflow of the Malware Analysis.

The first stage include the Static Properties Analysis.Static properties include strings embedded in malicious code. This type of data may be all that is needed to create IOCs, and it can be collected very quickly because the program does not need to be running.

Second stage compromise of Interactive Behaviour Analysis. Behavioral analysis is used to examine and interact with a sample of malware in a lab. Analysts try to understand the sample registry, file system, processes, and network activities. The process is time consuming and complicated and cannot be done efficiently without automated tools.

Thirdly, we perform fully automated analysis quickly and easily evaluates suspicious files. The analysis can determine potential consequences if malware infiltrates the network and then produces an easy-to-read report that provides quick answers to security teams. Fully automated scanning is the best way to deal with malware on a large scale.

At the last stage, we do manual coding using the principle of reverse engineering.At this stage, analysts can decode the code using debuggers, disassemblers, compilers, and specialized tools to decrypt the encrypted data, determine the logic behind the malware algorithm, and understand any hidden functionality that the malware has not yet exposed. Reversing code is a rare skill, and performing code reversals takes a long time. For these reasons, malware studies often skip this step and miss out on a lot of valuable information about the nature of the malware.

3.1.4 Types of Malware Analysis

Broadly, there are two types of malware analysis: static and dynamic analysis. A complete overview of the malware analysis is given in the below flowchart,

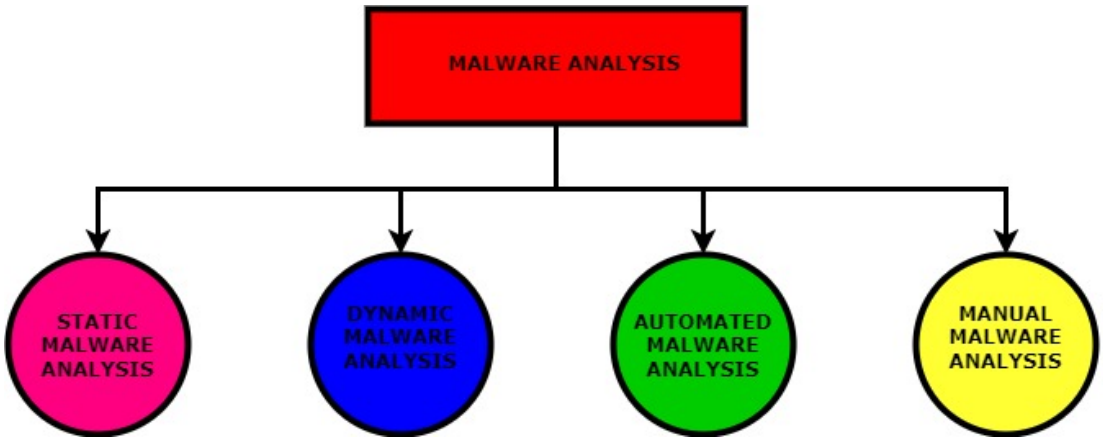
(a). Static Analysis: Malware code consists of two types of elements: static and dynamic. Static analysis focuses on the former, examining static properties such as metadata, headers, embedded assets, etc. A quick static analysis often reveals enough information to create an indicator of compromise (IOC), the malicious nature of document recording software. Assembly code is inspected to discover the file execution flow, harmful activity pattern, and other information that may be used to identify new malware or malware variants. Investigating assembly code to discover the characteristics and patterns of execution takes a lot of effort. Additionally, code obfuscation techniques significantly complicate the analyst's task.

(b). Dynamic Analysis: Dynamic analysis allows malware to deploy in a controlled environment while observing its behavior. By establishing a baseline for the host environment before and after dynamic scanning, you can learn more about malware behavior. Because running a malware file directly on the host machine might damage the operating system, malware samples are run in a controlled virtual environment . Utilizing virtualization software such as VMware or Virtual Box, a virtual environment may be built. Several activities, including the creation of new files, the deletion of system or user files, new log entries, registry key changes, URLs accessed, API Calls, the downloading of malware, the sending of data to command and control systems, etc. are seen when a malware file is running in a monitored environment. The file is classified as benign or harmful based on these behaviours.

(c). Automated Analysis: Automated analysis guides the malware through an automated workflow where its various behavioral and static properties are tested. The Falcon Sandbox and the AI-based SANDBOX are some of the tools that can help you with this. It is advisable to run all these tools together to get a holistic view of what the malicious app is capable of.Automation is capable of producing comprehensive reports and supplying data to incident response systems, ensuring that the analyst receives just the most crucial signals. Some of the examples of automated malware analysis tools are the AI-powered SNDBOX and Falcon Sandbox.

(d). Manual Analysis: In a manual analysis, an analyst can choose to decompose the code manually using tools such as debugger, decompiler, and de-

crypter. Manual parsing is also known as code inversion because you essentially start with the latest software, go back through the code, and then get to the original logic.



3.2 Malware Detection Techniques

Malicious programmes that can affect a computer system or network assets have to be found and defended against using malware detection techniques. To identify and categorise the malware samples into the correct family, the input is expressed in a variety of ways. It is necessary to have the appropriate knowledge or understanding of the malicious file that corresponds to the harmful file's real behaviour. Different malware samples are examined for this purpose using static, dynamic, or hybrid methodologies.

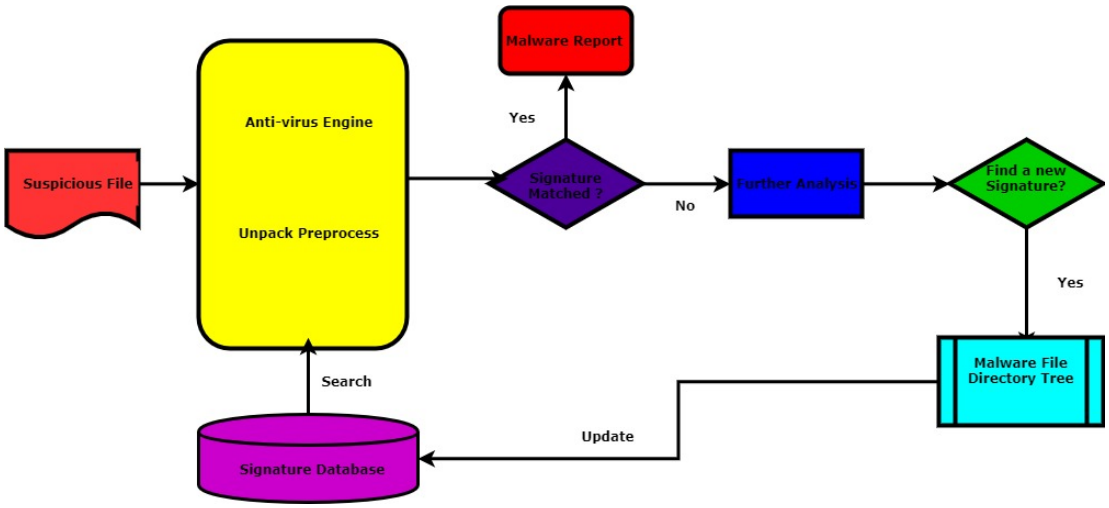
Focusing on malware detection, Bazrafshan et al. [20] highlight three key approaches for finding malicious software: those based on signatures, behaviours, and heuristics, with the last also utilising machine learning techniques. They also specify the classifications of characteristics, such as API calls, control flow graphs, n-grams, opcodes, and hybrid features, that are employed by the reviewed algorithms for malware identification.

3.2.1 Signature-based Malware Detection Technique

Many security products rely on file signing to detect malware and other malicious files. This technique involves reading or scanning a file and testing whether the file matches a set of predefined attributes. These attributes are called the "signatures" of the malware.

Using sequences of the functions, Karnik et al.[17] (2007) suggested a method for malware identification.The group of opcodes was represented by one element in a sequence. And,function sequence acted as the infected file's signature to identify the malware's versions.

A systematic method for developing detection signatures based on user agent abnormalities in malware HTTP traffic was provided by Kheir (2013) [3] They started by removing user agent header information from HTTP transmission. They then carried out a preliminary high-level clustering phase to classify user agents that are probably going to exhibit similar patterns. The agents that can be characterised by a common set of signatures were then grouped together by a second clustering process that was applied to each group of user agents. Finally, incremental K-means clustering was used to put user agents in the same clusters that have comparable pattern sequences.

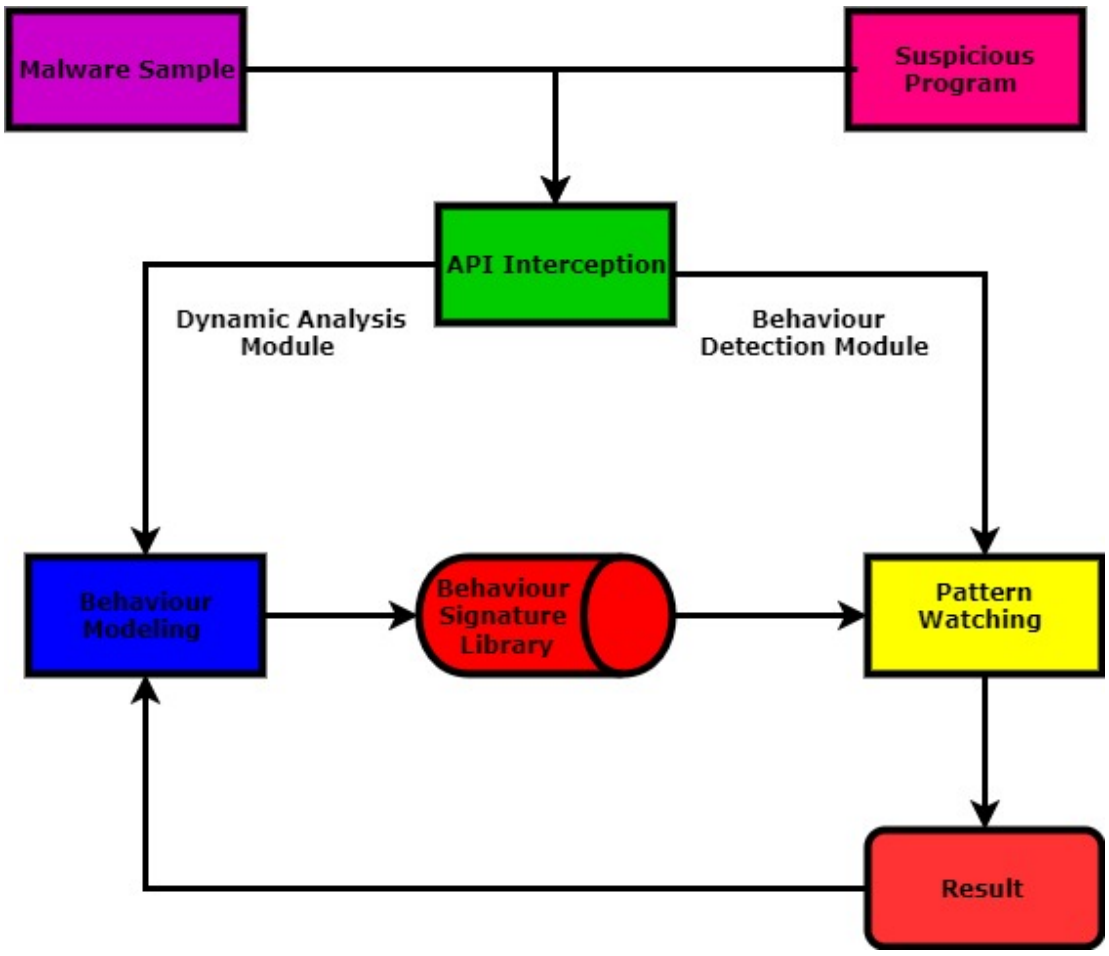


3.2.2 Behaviour-based Malware Detection Technique

In the behavior-based method, malware is identified based on the harmful actions it takes while it is being executed. When a file displays anomalous behaviour that differs from the pre-stored behaviour of normal files, that file is deemed malicious. Because malware files share some dangerous behaviours, this malware detection method may also detect fresh malware files. The malware detector in the anomaly-based detection method is solely educated by looking at the benign files. The importance of machine learning increases for spotting sophisticated infections.

In this sense, behavior is defined by various features such as APIs, browser events, system events, network events, etc. The behavior-based approach classifies these parameters into her three main categories: file activity, registry activity, and network activity.

Trinius et al.(2011) [17] described a new representation Malware Instruction Set (MIST) to represent the behaviour of malware. A network behaviour classification approach was advised by Guo and colleagues (2014)[4] to identify mobile malware based on its behavioural traits. The two parts of this effort are network behaviour detection and analyzer training.



3.3 Problem Statement

In recent years, the number of malware attacks has increased dramatically. Costs related to ransomware losses are expected to reach $8 billion in 2018. Although 9 % of companies affected by ransomware attacks did not pay the ransom, the damage caused by the loss of data and the number of man-hours spent to mitigate the infection. can be significant. Since its peak in 2016, the frequency of ransomware attacks has decreased. Instead, cryptography is on the rise, which uses the computing power of victims to mine cryptocurrencies for the benefit of attackers. Identifying known malware is usually easy, but the biggest problem is dealing with unknown binaries. Determining whether an unknown/new executable is malicious usually requires consulting a professional analyst who can assess the state of the executable. If expert analysis shows that an executable file is malicious, signature patterns (based on static, dynamic or

hybrid characteristics) can be created to prevent future attacks (similar to You can provide detection capabilities to your detection tools. In other words, the analysis turns. "invisible malware" into signatures that can be detected in the future. Manual analysis is very reliable. but it is not scalable and suitable for all files due to the high cost of manual analysis.

4 Discussion of Exisiting solution

The science of machine learning involves utilising algorithms to evaluate data, discover patterns from it, and then use these patterns to forecast or decide whether to utilise further samples.Since behavior-based approaches adhere to regulations and taking specific advice from an expert may not be beneficial to us if the virus adopts a novel strategy or is completely new household The benefit of using machine learning is identifying zero-day malware with extensive data analysis malicious and safe files, and allow the algorithm to discover the patterns that set them apart from one another.

4.1 Malware Detection using Machine Learning

The categorization and clustering of malware today is greatly aided by machine learning (ML). The classification of benign and malicious files has been the subject of extensive research in the literature. By taking into account additional attributes of both malicious software and benign samples, ML algorithms provide developers greater flexibility and room to build more accurate models. Malicious URL identification, intrusion detection, and virus detection are only a few applications of machine learning in the realm of computer security. Malware samples are examined, and a feature set of data is gathered for the classifier's training. Fig. The schematic diagram of the machine learning-based malware detection system is shown in Figure 3. The picture depicts the fundamental design of machine learning classifiers, similar to those used in different problem domains.

4.2 Static Analysis

Static analysis consists of examining the code or structure of an executable file run it. This type of analysis can check if a file is malicious and provide information about its functionality. It can also be used to create simple signature sets.For example, hashing is the most common way to uniquely identify malicious programs. In other words, a hashing program creates a unique hash, sort of like a fingerprint that identifies a user program. The two most common hash functions are Message-Digest Algorithm 5 (MD5) and Secure Hash Algorithm 1 (SHA-1).

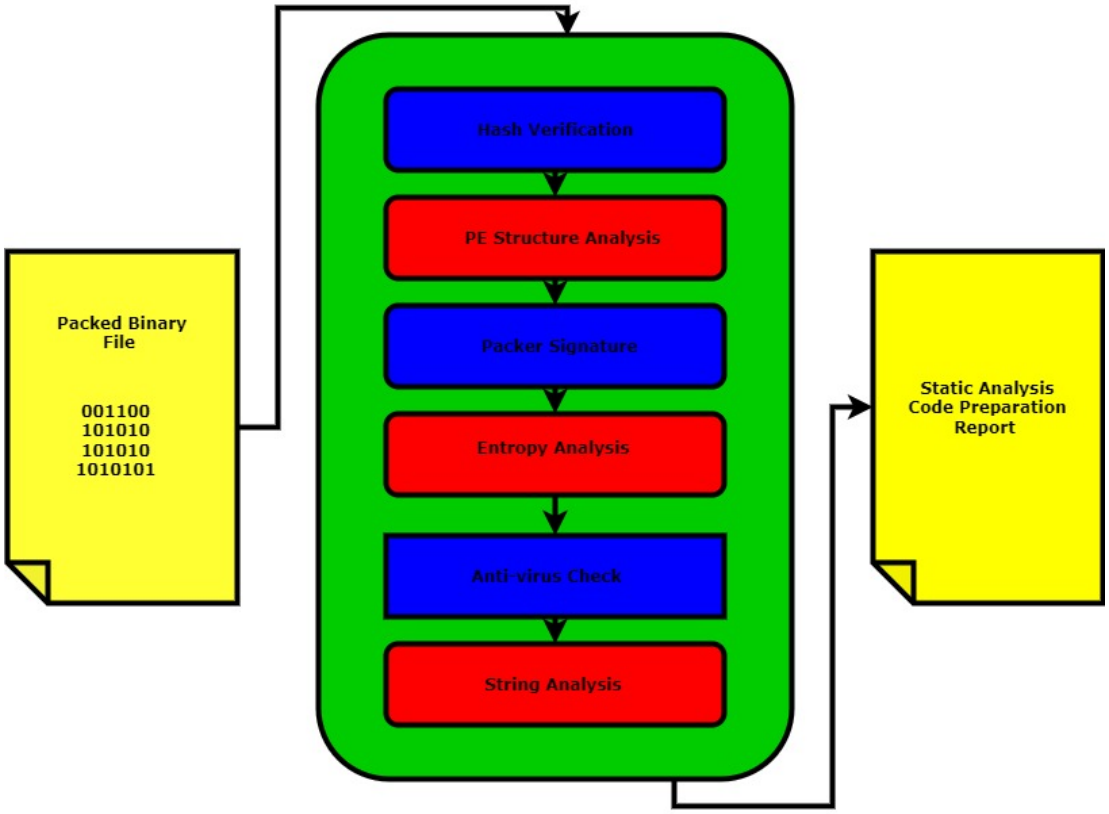
Disassemble/debugger tools such as IDA Pro and OllyDbg display malware code as Intel X86 assembly guides.It provides a lot of insight into what malware is doing and provides patterns for identifying attackers. Use a memory dump

tool such as LordPE [9] or OllyDump is used to retrieve protected code.It resides in system memory and outputs to a file. This is a useful technique for analyzing packed files Executable files that are difficult to disassemble.Static analysis is very expensive and unreliable. Also, if a binary executable (obtained by compiling the source code) is used for static analysis,information such as data structures and variable sizes leaked, complicating malware code analysis

Moses et al.[19], studying the shortcomings of static analysis method. In their work, they introduced a scheme based on code obfuscation The fact that static analysis alone is insufficient to detect or classify malware. Furthermore, they suggested that dynamic analysis should complement static analysis Vulnerable to conversion from code obfuscation.

Static analysis with ML is a fast and reliable mechanism It considers different characteristics of PE32 executable files to classify malicious and benign samples. Machine learning-powered static malware analysis can be used as part of your cyber threat intelligence (CTI) activities to automate the detection of indicators of compromise from static functions in PE32 Windows files.[15]

Given below is the workfow of the static analysis.



Important static properties used for static analysis are machine, properties,

number of sections, TimeDateStamp,PointerToSymbolTable, NumberOfSymbols, magic,MajorLinkerVersion, MinorLinkerVersion etc.[8]

4.2.1 Feature Selection

The training part of the malware detection technique consists of Feature Extraction. Feature Extraction is a process in which features are extracted from the malicious or begnin files . The extracted features are then used to train the classifier and to determine whether the input file is malicious or not. The method is quite efficient for large datasets as it helps in recognising accurate pattern. The objective is typically to reduce the feature space and minimize the overall cost of measurement acquisition.

The features on the basis of the feature extraction process is done are as following :

1. Binary Sequences : A binary can be charcterized by computing feature on its byte-level content.A common static approach is to examine particular byte sequences within a PE.

2. Opcodes : Opcodes may be recovered using static analysis by looking at the assembly code, identifying the machine-level operations carried out by a PE. One of the most often utilised features is opcode frequency. It counts the instances in which a given opcode is used by PE or occurs in the assembly.

3. API and System Calls: Similar to opcodes, but on a higher level, APIs and system calls allow the examination of sample behaviour. By examining the disassembled code (to obtain a list of all calls that might possibly be performed) or the execution traces, they can be extracted statically or dynamically.

4. Network Activity: Observing how the PE interacts with the network reveals a wealth of important details. Addresses that are contacted and traffic that is created can provide important information, such as about how to communicate with a command and control centre.

5. File System: The characteristics of file systems that are taken from the Windows Registry are a particularly relevant kind. One of the primary sources of environmental data for a PE is the registry, which also serves as a vital tool for operating system hookups, such as persistence.

4.3 Dynamic Analysis

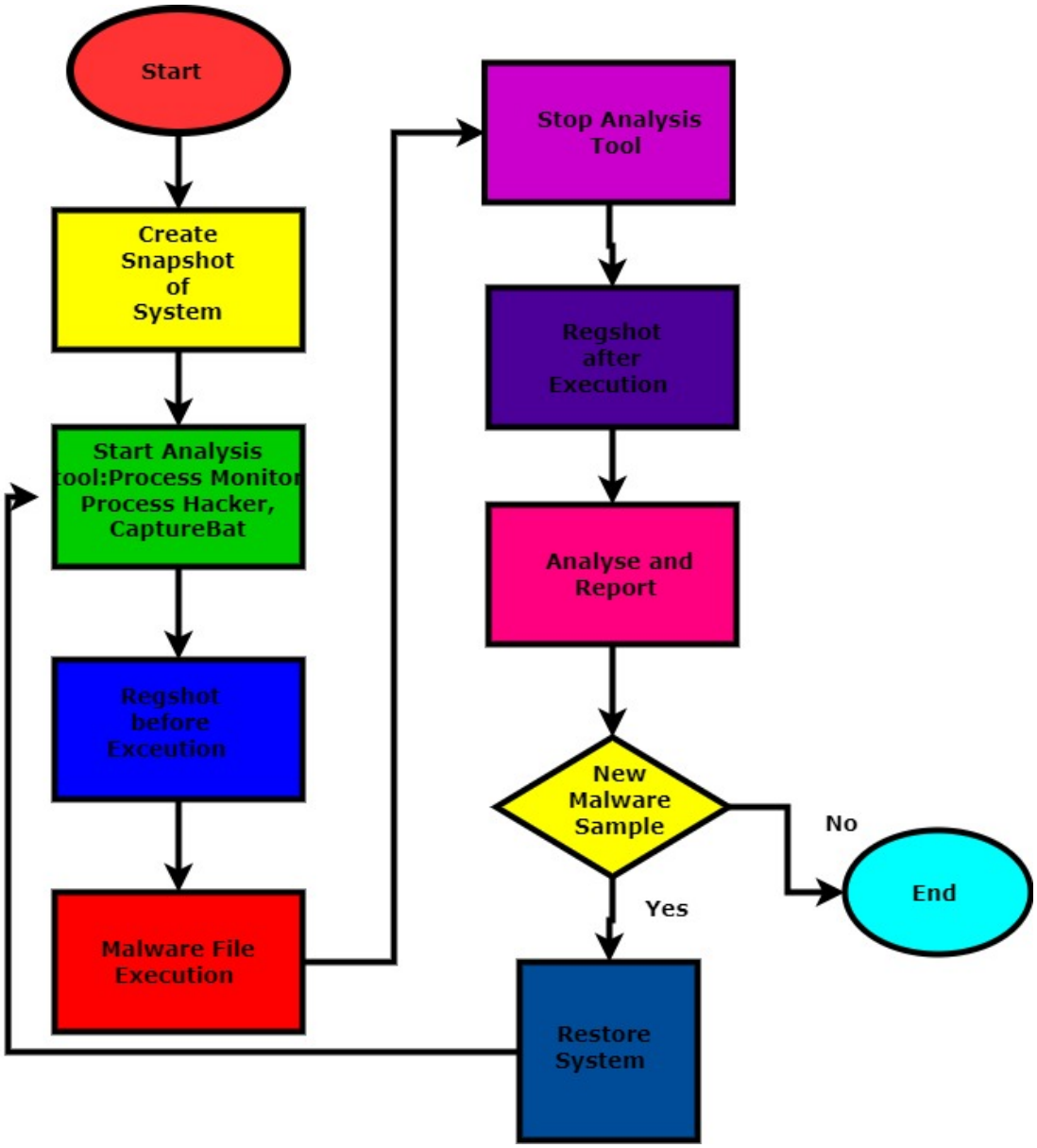
Dynamic analysis is the testing and evaluation of programs by running data in real time. The goal is to find bugs in the program while it's running, rather than repeatedly examining the code offline.

Dynamic analysis involves running a program and observing its behavior system. This is typically done when static analysis reaches a dead end for one of the following reasons: Obfuscation or exhaustion of available static analysis techniques,non static analysis, tracking the actual actions performed by the program. However, the analysis run in a secure environment to avoid exposing your system to unnecessary risk.Both the computer running the analyzer and

the rest of the computers on the network.A dedicated physical or virtual machine is set up for this purpose.

From the system call sequences, Ahmed et al [24] retrieved two independent dynamic feature types (spatial and temporal) and combined them to represent malware. To categorise malware, they used five classifiers: the native Bayes algorithm, decision tree, inductive rule learner, and support vector machine.

Carline (2017)[3] presented an approach to perform dynamic analysis on virtual machines to extract program run-time traces from both benign and malicious executables. she By testing two algorithms, she analyzed the sequence of opcodes executed to detect malware.(1)A random forest classifier to classify all count-based data, and (2) hidden Markov models classify data based on the temporal relationship of opcode sequences.



4.3.1 Supervised and Unsupervised Learning

Supervised Learning Supervised learning is a type of machine learning in which a machine is trained using properly "labeled" training data, and the machine predicts an output based on that data. Tagged data means that some input data is already tagged with correct output.

In supervised learning, the training data provided to the machine acts as

a supervisor, teaching the machine to correctly predict the output. Apply the same concepts that students learn under the supervision of a teacher.

Supervised learning is the process of providing both input data and correct output data to a machine learning model. The goal of supervised learning algorithms is to find a mapping function that maps an input variable (x) to an output variable (y).

Unsupervised Learning Unsupervised learning is machine training that uses unclassified or unlabeled data and allows an algorithm to act on that data without instructions. Here, the machine is tasked with grouping unordered data based on similarities, patterns and differences without prior training on the data. Unlike supervised learning, no teacher is provided, meaning the machine is not trained. Therefore, the machine is limited to finding the hidden structure of unnamed data.

4.3.2 Machine Learning algorithm

I. Linear Regression A linear regression algorithm is a type of supervised learning algorithm that performs regression tasks and is one of the most popular and well-known algorithms in data science. Regression analysis is a type of predictive modeling that finds relationships between input and target variables. The most common type of regression is linear regression, which shows the strength of the correlation between two variables and whether the correlation is positive or negative.

II. Decision Tree Decision tree classification builds a decision tree by computing the information gain for each attribute in the data set. The attribute with the greatest information gain becomes the root. Then another leaf on the root. Next, class prediction is done using the created decision tree. In a decision tree, each interior (non-leaf) node examines features, feature values,correspond to each branch, and leaf nodes represent class labels. His two split functions, Infogain and Gini Index, are used for model training.

III. Logistic Regression Logistic regression is a parametric binary classification algorithm. LR learns coefficients from training data to create a logistic regression classifier. In general, LR estimates empirical values of parameters in qualitative response models.

IV.Artificial Neural Network Artificial neural networks are process models that mimic how the human brain works, minimizing error rates and making decision boundaries easier to find.

V. SVM The SVM algorithm creates a hyperplane to divide the input data instances of records into different classes. For binary classification, a vector of points can be visualized in a two-dimensional input space that separates the input data instances into two distinct classes: Benign and malware classes.

VI. K-NN K-Nearest Neighbor (K-NN) classification algorithms classify input instances given the class labels of the k-nearest neighbor training instances. The class of input instances is predicted as the class of many instances. Find the class label of the input instance from the nearest K closest instances using Euclidean, Manhattan, Hamming, and Minkowski distance measures.

5 Evaluation

Based on recent proposed research on malware detection and classification models along with a review of the approaches and techniques used the shortcomings and usefulness of these approaches and techniques have been noted. For this reason,our review was able to identify challenges and open points.

5.1 Obfuscation Techniques

These methods were initially employed to safeguard software's intellectual property.Last but not least, the creators of harmful software used those methods to migrate their virus to new, more difficult-to-detect forms.Instruction reordering, dead code insertion, registry reassignment, and other methods the features of malware have been updated via instruction substitution together with regards to performing the same tasks.

5.2 Zero-day attack

To the best of our knowledge, there may be a zero-day after each fresh assault carried out by zero-day malware before this malware is found .The malware detection models that have been created based on historical knowledge are ineffective when trying to identify zero-day malware since unknown malware has new features that serve its aims.Signature-based methods fall short in spotting Zero-Day attacks. Additionally, it cannot identify sophisticated new malware.[13]

5.3 False Positive/Negative Rate

False positives are far more significant than false negatives in successful malware detection models, even if an increase in false positive or false negative rates lowers the model detection accuracy. On a user's computer, the operating system may become unbootable and other software may stop working if a valid file is wrongly detected as harmful.

5.4 Response

Some of the evaluations, such as anomaly-based technique, are restricted to a specific level of malware. Some methods failed to identify the ideal amount of characteristics required to train a classifier.

6 Future Directions

Due to the clever and cunning nature of malware, which may be quickly detect dynamic malware analysis, we need to need a controlled environment for dynamic analysis, which is not detectable. The nature of malwares nowadays is packed, and it is frequently able to perform statically. We require a framework,

which is dynamic at first, then applying once malware has been unpacked the extraction of static characteristics from it.

In the future, we would also want to categorise unseen samples using ensembles of classifiers that use majority voting and evaluate malware and benign samples using an ensemble of features. [25]. We intend to examine the effectiveness of deep learning and machine learning models on evasive samples produced by feature alterations and to provide a defence against adversarial attacks.

In one of the research paper [23] as future work, they would like to modify the dynamic feature such as API calls.

Since internet tools have been made available, the everyday manufacturing of zero-day malware and unidentified malware variants has significantly expanded. These tools allow users to develop new malware or reformat current malware using obfuscation techniques to add new variants. The built models must thus be capable of updating effectively in order to learn the upcoming new behaviours. To do this, deep learning techniques may be created and deployed in combination with unsupervised machine learning approaches for updating learning and constructing models that are adaptively learning new malevolent behaviours.

7 Conclusion

This study project has demonstrated the many tools and approaches that may be used to investigate a specific malware. Numerous studies have demonstrated that a single malware sample cannot be examined by a single instrument. According to experimental findings, each malware analysis tool uses a unique measure and method to examine the harmful code. The articles reviewed were surveyed and classified into two main categories. (1) signature-based and (2) behavior-based approaches. Along with works done in the field of malware analysis bascially in the static and dynamic malware analysis.

References

[1] L. Demetrio, S. E. Coull, B. Biggio, G. Lagorio, A. Armando, and F. Roli. Adversarial exemples: A survey and experimental evaluation of practical attacks on machine learning for windows malware detection. *ACM Transactions on Privacy and Security (TOPS)*, 24(4):1–31, 2021.

[2] H. El Merabet and A. Hajraoui. A survey of malware detection techniques based on machine learning. *International Journal of Advanced Computer Science and Applications*, 10(1), 2019.

[3] D. Gibert, C. Mateu, and J. Planes. The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. *Journal of Network and Computer Applications*, 153:102526, 2020.

[4] M. A. Jerlin and K. Marimuthu. A new malware detection system using machine learning techniques for api call sequences. *Journal of Applied Security Research*, 13(1):45–62, 2018.

[5] T. Kumar, S. Sharma, H. Goel, and M. Memoria. A novel machine learning approach for malware detection. In *International Conference on Advances in Engineering Science Management & Technology (ICAESMT)-2019, Uttaranchal University, Dehradun, India*, 2019.

[6] I. M. M. Matin and B. Rahardjo. Malware detection using honeypot and machine learning. In *2019 7th international conference on cyber and IT service management (CITSM)*, volume 7, pages 1–4. IEEE, 2019.

[7] O. Or-Meir, N. Nissim, Y. Elovici, and L. Rokach. Dynamic malware analysis in the modern era—a state of the art survey. *ACM Computing Surveys (CSUR)*, 52(5):1–48, 2019.

[8] C. Raghuraman, S. Suresh, S. Shivshankar, and R. Chapaneri. Static and dynamic malware analysis using machine learning. In *First International Conference on Sustainable Technologies for Computational Intelligence*, pages 793–806. Springer, 2020.

[9] H. Rathore, S. Agarwal, S. K. Sahay, and M. Sewak. Malware detection using machine learning and deep learning. In *International Conference on Big Data Analytics*, pages 402–411. Springer, 2018.

[10] H. Sayadi, H. M. Makrani, S. M. P. Dinakarrao, T. Mohsenin, A. Sasan, S. Rafatirad, and H. Homayoun. 2smart: A two-stage machine learning-based approach for run-time specialized hardware-assisted malware detection. In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 728–733. IEEE, 2019.

[11] H. Sayadi, H. M. Makrani, O. Randive, S. M. PD, S. Rafatirad, and H. Homayoun. Customized machine learning-based hardware-assisted malware detection in embedded devices. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 1685–1688. IEEE, 2018.

[12] H. Sayadi, N. Patel, A. Sasan, S. Rafatirad, and H. Homayoun. Ensemble learning for effective run-time hardware-based malware detection: A comprehensive analysis and classification. In *Proceedings of the 55th Annual Design Automation Conference*, pages 1–6, 2018.

[13] K. Sethi, S. K. Chaudhary, B. K. Tripathy, and P. Bera. A novel malware analysis framework for malware detection and classification using machine learning approach. In *Proceedings of the 19th International Conference on Distributed Computing and Networking*, pages 1–4, 2018.

[14] M. Sewak, S. K. Sahay, and H. Rathore. Comparison of deep learning and the classical machine learning algorithm for the malware detection. In *2018 19th IEEE/ACIS international conference on software engineering, artificial intelligence, networking and parallel/distributed computing (SNPD)*, pages 293–296. IEEE, 2018.

[15] A. Shalaginov, S. Banin, A. Dehghantanha, and K. Franke. Machine learning aided static malware analysis: A survey and tutorial. *Cyber threat intelligence*, pages 7–45, 2018.

[16] S. Sharma, C. Rama Krishna, and S. K. Sahay. Detection of advanced malware by machine learning techniques. In *Soft computing: Theories and applications*, pages 333–342. Springer, 2019.

[17] J. Singh and J. Singh. A survey on machine learning-based malware detection in executable files. *Journal of Systems Architecture*, 112:101861, 2021.

[18] J. Singh and J. Singh. Assessment of supervised machine learning algorithms using dynamic api calls for malware detection. *International Journal of Computers and Applications*, 44(3):270–277, 2022.

[19] S. Talukder and Z. Talukder. A survey on malware detection and analysis tools. *International Journal of Network Security & Its Applications (IJNSA) Vol*, 12, 2020.

[20] D. Ucci, L. Aniello, and R. Baldoni. Survey of machine learning techniques for malware analysis. *Computers & Security*, 81:123–147, 2019.

[21] N. Usman, S. Usman, F. Khan, M. A. Jan, A. Sajid, M. Alazab, and P. Watters. Intelligent dynamic malware detection using machine learning in ip reputation for forensics data analytics. *Future Generation Computer Systems*, 118:124–141, 2021.

[22] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, and S. Venkatraman. Robust intelligent malware detection using deep learning. *IEEE Access*, 7:46717–46738, 2019.

[23] C. Wu, J. Shi, Y. Yang, and W. Li. Enhancing machine learning based malware detection model by reinforcement learning. In *Proceedings of the 8th International Conference on Communication and Network Security*, pages 74–78, 2018.

[24] D. Yuxin and Z. Siyi. Malware detection based on deep learning algorithm. *Neural Computing and Applications*, 31(2):461–472, 2019.

[25] Y. Zhang and Z. Wang. Hybrid malware detection approach with feedback-directed machine learning. *Information Sciences*, 63(139103):1–139103, 2020.