Malware Detection Using Machine Learning Techniques: A Review

Mohammed Saadoon*, Suhad Faisal

Department of Computer science, College of Science, University of Baghdad, Baghdad, Iraq.

*Corresponding author E-mail: mohammed.abd2201m@sc.uobaghdad.edu.iq

ARTICLE INFO

ABSTRACT

Keywords

Malware Analysis; Malware Detection; Data Stream; Classification; dataset; Feature section

Cybersecurity has become an important priority requiring immediate response. Threats have become a commonplace phenomenon. This paper will examine malware, a kind of cybersecurity encompassing harmful software that expropriates data and jeopardizes privacy and security. We will elucidate the application of malware analysis and machine learning methodologies for detection. Currently, fraudsters employ polymorphic malware that utilizes strategies challenging for conventional detection technologies to identify. Therefore, this study will utilize a survey on machine learning algorithms that facilitate the detection of different malware types while ensuring optimal detection performance and accuracy.

Received 22 Apr 2024; Received in revised form 14 Jun 2024; Accepted 10 Aug 2024, Published 31 Aug 2024

1. Introduction

Cyberattacks are the most critical matter in the whole world of modern technology. Every day, a wide range of malicious attacks target the world, aiming to steal, manipulate, or destroy information [1]. Malware is considered one of the most frequent cyberattacks [2]. Any binary code or script that aims to harm a computer system or steal private information is known as malware. Malware can occur in diverse patterns and formats, like binary shell code, executable files, and scripts. We classify malware by (1) type, which refers to its use of a general public name; (2) behavior, which signifies its malicious intent for interruption; and (3) privilege, which refers to the intruder's advantage and position. For instance, Figure 1 illustrates the classification of malware [3].classification [3].

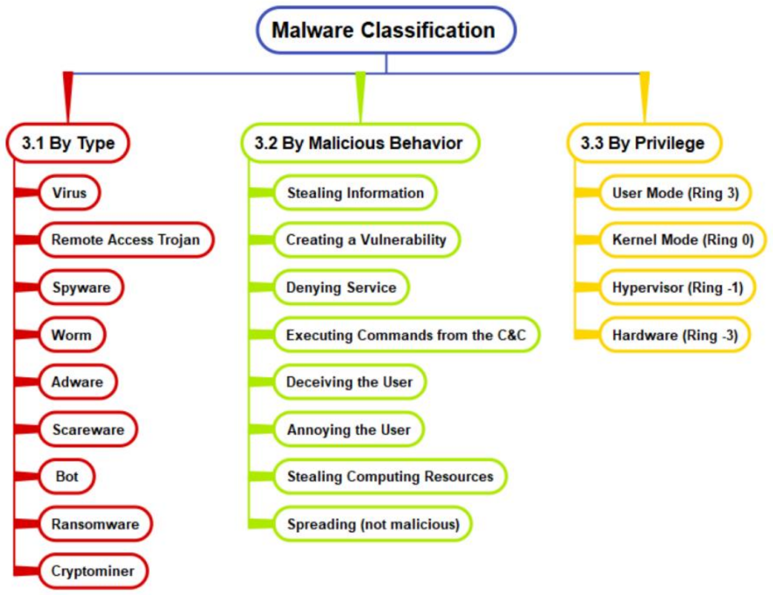

Figure 1: Various malware taxonomies

Most literature refers to the most common types of malware as general public malware, which includes Trojan horses, viruses, spyware, ransomware, rogue software, adware, and more. Each type of malware serves a distinct purpose and contributes to the same overall objective: malicious intent. [3]. malware detection. The traditional technique of malware analysis encompasses both static and dynamic elements. In this literature, data analysis is regarded as a component of data collection as it employs machine learning techniques. It is considered a crucial part of the machine learning process as it involves the entire training dataset. Therefore, it is crucial to manage malware analysis through the use of machine learning algorithms [4].

Malware detection: there are two relevant types of detection techniques: signature-based and behavioral-based. Hence The signature technique is not capable of detecting complex malware, so it cannot detect zero-day attacks. The other behavior technique is considered very hard to detect in the entire behavior found in datasets [4].

We will show and survey machine learning classifier models to solve these problems, and we will use intelligent malware analysis in the data collection phase because of the limitations of detection techniques [4].

2. Background

2.1 Malware types

Understanding malware is crucial, as it aids in analyzing various types of malware and identifying their techniques and behaviors [5],[6] . Some of the most common types of malware are:

Virus: This is a basic type of software. infected user by downloading it from different sources; his intent is to replicate himself by using other programs and modifying it [5].

Trojan: He disguises himself to look like a legitimate program, but in reality, he hides malicious intent—that's to steal and get access to secret layers in computer systems [7].

Worm: It's a type of malicious program that replicates itself without needing user help and uses it to corrupt computer systems, steal personal information, and consume large amounts of memory [8].

Spyware: It's a type of malware used to gather information from infected systems, like personal information, passwords, and keystroke websites, and send that information to a third party without the knowledge of the infected user [9].

Ransomware: is one of the most dangerous and common forms of malware today. It's difficult to detect and stop because of the use of a technique that encrypts all information of the infected system by sending an interface order to the infected by paying fees to get a key to decrypt his information [10].

Rootkit: This type of malware uses high privileges and permission to access sensitive information that was permitted to be accessed and hides its presence; hence, it is very difficult to detect and remove [11].

Backdoor: A backdoor is a malware type that negates normal authentication procedures to access a system. As a result, remote access is granted to resources within an application, such as databases and file servers, giving perpetrators the ability to remotely issue system commands and update malware Backdoor installation is achieved by taking advantage of vulnerable components in a web application. Once installed, detection is difficult as files tend to be highly obfuscated [12].

2.2. Malware analysis

The process that produces the signature for newly discovered malware depends on examining its code or executing it in a safe environment, and that's what gives us knowledge about malicious programs [13].

2.2.1 Static analysis

It's a process of examining executable malicious code without executing it, and it's considered a safe way to examine malware. The Python language offers a portable executable PE file library to facilitate the extraction of features from malicious code. Static analysis also gives the reverse engineer a clue about a specific program goal [14]. Moreover, code analysis, another name for static analysis, typically separates the various operations of a specific malware sample to assess its level of suspicion [15]. Examining file headers, known as static analysis, is crucial as it provides initial general information and initiates the analysis by extracting the executable portion of the portal [16]. Static analysis considers strings, metadata containing suspicious code, and executable files very useful as they provide information about imported functions. Static analysis is safer than dynamic analysis because it does not need to execute a program, so you will not be a victim of the malicious program [17].
Static analysis alone is not enough to detect some type of malware and can't identify zero-day malware, so a company needs dynamic analysis to be more efficient and useful [18].

2.2.2 Dynamic analysis

Dynamic analysis The process of dynamic analysis involves executing and examining malware in a safe environment, closely monitoring the behavior of malicious content, and creating an outcome report that reflects the risk level of the malware. It also employs various techniques, such as using system APIs to monitor various operations and detect different types of malware behavior based on different attributes, such as network calls and registry modifications [19]. Numerous system APIs are called by running software, and these APIs define all software activities, such as network access, file creation and modification, etc [20]. Malware analysis based on APIs is occasionally constrained for two reasons. (1) Cannot fully detect the semantic information in various arguments. (2) It is difficult to detect the relationship between API calls for reporting software behaviour. From one API to the next, the specific arguments and their names will change. To understand the necessary arguments and their definitions for each distinct API call, it's critical to consult the documentation for the API. For your API requests to be handled effectively and produce the appropriate results, these arguments must be configured properly [21]. We put into practice the system. For example, as shown in Figure 2, in order to gather the run-time API requests. The system is divided into three sections: collecting PE files, gathering behavioural data, extracting features, and training models [22]. Tools such as wire shark, process explorer, process monitor, and capture bat are used to keep an eye on the behaviour. Monitoring system calls, injunctive authorization traces, function and API calls, the network, information flow, etc. is the goal of this type of analysis [23].

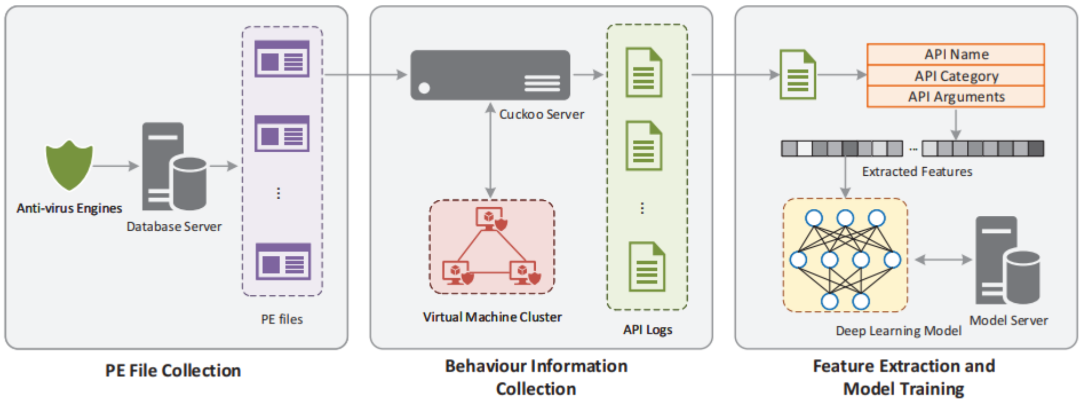

Figure 2: System Architecture

3. Categories of malware classification

We show in this part the categorizations of malware classification, which are divided into two categories. The first one represents a feature, and it’s the base of our work and how we extract it. The second is an algorithm that is acquired for detection and analysis using machine learning techniques. For example, as shown in Figure 3, shows this taxonomy[24]

3.1 Feature extraction

Feature extraction is considered part of malware analysis and is done by using different methods like static analysis and dynamic analysis such as network traffic, API calls, and registry modifications, as explained in paragraphs 2.21 and 2.2.2.[21]. Unwanted elements (missing, redundant, or endless values) should be eliminated or modified from the majority of the datasets that are currently available. Preprocessing is a necessary step in order to obtain an appropriate dataset [25]. feature selection, machine learning data or pattern recognition applications may be understood, computation time can be decreased, and prediction performance can be enhanced [26].

4. Malware classification and detection algorithm

There are two relevant types of detection techniques: signature-based and behavioural-based. Hence The signature technique is not able to detect complex malware, so it cannot detect zero-day attacks. The other behavior technique is considered very hard to detect the entire behavior found in datasets. Because of the limitations of detection techniques, we will use machine learning classifier models to solve these problems and will use intelligent malware analysis in the phase of data collection [27].

4.1 Signature-based

Malware with a signature feature uniquely identifies each piece of malware by encapsulating the program's architecture. Commercial antivirus software frequently employs a signature-based detection approach. This method is fast and useful for detecting known malware. But inadequate to detect an unknown one. Also, obscure techniques are hard to detect by signature-based methods, for example, as shown in Figure 4,[28].

4.2 Behavior-based

In this method, malware is detected based on its behaviour after analysis and feature extraction based on features related to behaviour, which became an indicator used to classify malicious files and also help categorize malware families. Provided long-short-term models (LSTM) based on the typical API call sequences provided by each malware family to create binary and multi-classification models [29].

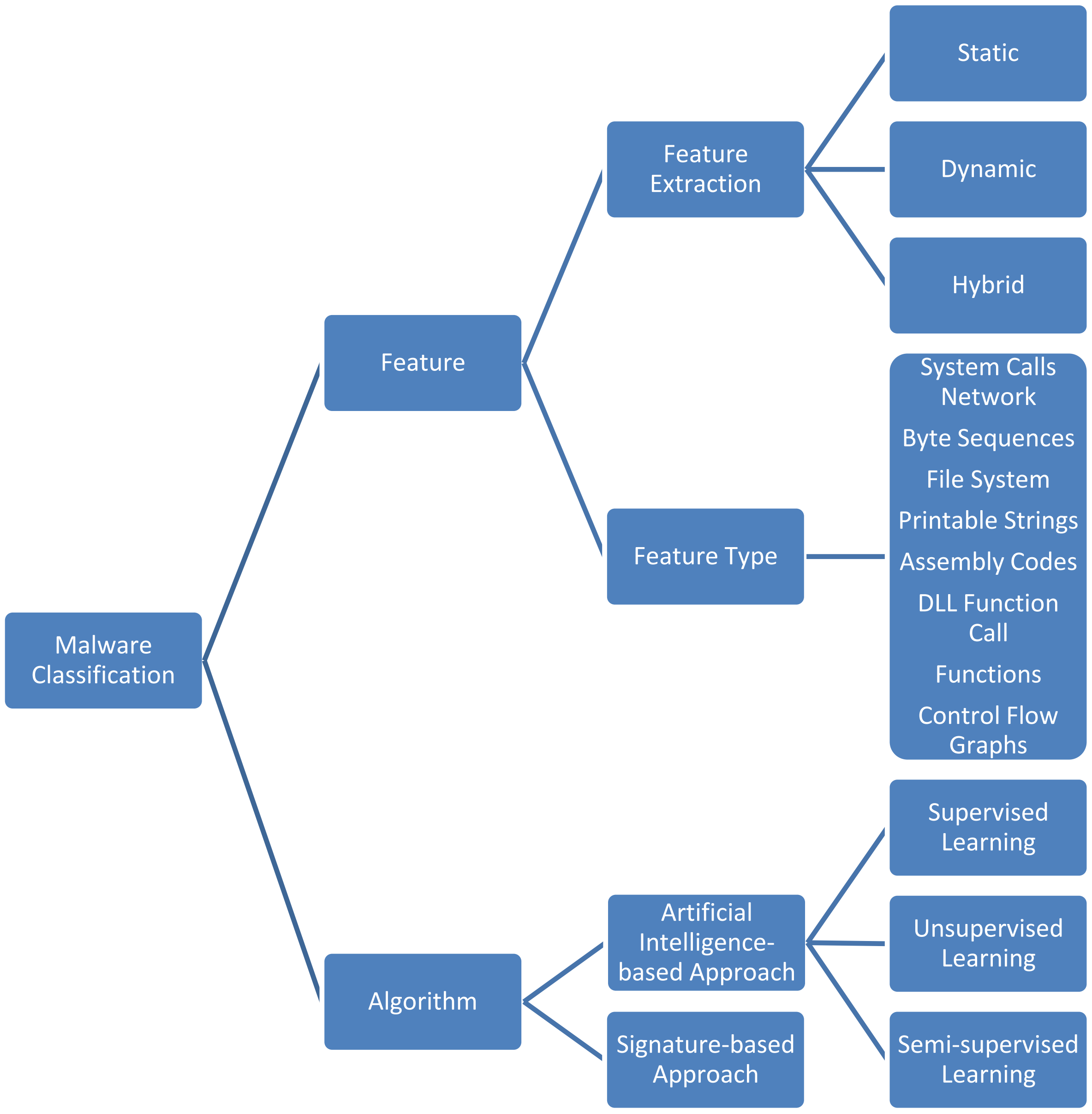

Figure 3: The proposed taxonomy

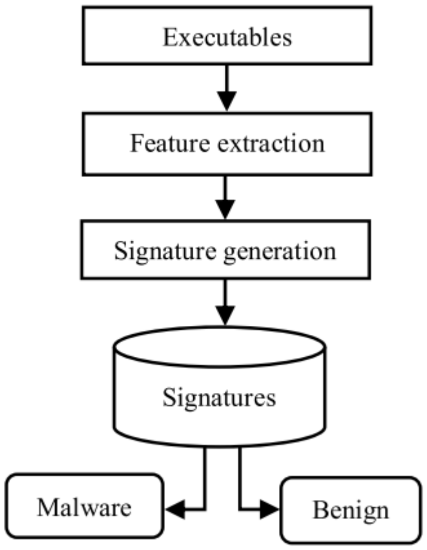

Figure 4: signature-based malware detection schemas

4.3 Machine learning and technique

Machine learning is the process of gathering data from a dataset, cleaning it, and performing feature extraction. These are then split into training data and testing data to train a specific model, which is then evaluated until it is ready for predictions. Hence, there are several methods for feature extraction, for example, embedded-based, which is used to select the most relevant. For example, as shown in Figure. 5, [30].

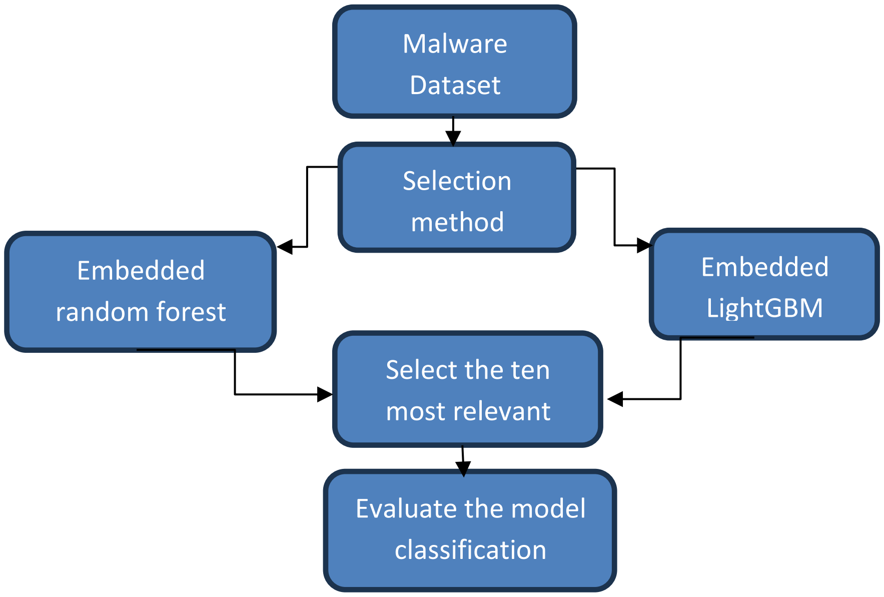

Figure 5: processes of selecting relevant feature

4.3.1 Machine learning algorithm

There are several types of algorithms used by machine learning that's differentiated from each other by performance, time, and accuracy, as shown below [31].

4.3.1.1 Supervised learning

Supervised is a type of machine learning technique that uses labeled data for training models and makes predation or classification, which is considered one of the most relevant of machine learning [32]. Some common supervised learning algorithms include:

a) Support vector machine

This type of supervised machine learning uses classification and regression and is suitable for linear and nonlinear classification [33]. SVM is useful with nonlinear classification and gets very good results with pattern classification because it uses the kernel function [34]. SVM uses hyperplanes to separate multi-dimensional data sets and is called the optimal hyperplane. SVM is useful with binary classification and noisy data [35]. Support vectors are the features of the identified objects that are closest to the parallel hyperplanes. For example, as shown in Figure. 6 [36], depicts an illustration of a separating hyperplane construction in 2D space [36]. An n-dimensional feature space is the representation of the input data. Afterwards, a (n-1) dimensional hyperplane splits space in two. The Yi matrix assigns the labels Yi = 1 for class 1 and Yi = 1 for [27]class 2 to the n-dimensional input data xi (i = 1, 2,..., l). A hyperplane can be defined for data that are linearly separable[37] as in Equ.1.

$$\mathrm{F(x)} = \mathrm{W.X} + \mathrm{p} = \sum_{i=0}^{n} \mathrm{WiXi} + \mathrm{P} = 0 \qquad (1)$$

In Equation (3), W is an n-dimensional vector, p is a scalar, and Sgn (f(x)) is the decision function. These establish the location of the hyperplane that totally divides the area, and it has to respect these restrictions, as in Equ.2 [37].

$$\text{Yi (W. Xi + p)-1} \geq 0 \Rightarrow \begin{pmatrix} f(xi)\text{=Wi,Xi+p}\geq 1 \text{ Yi= +1} \\ f(xi)\text{=Wi,Xi+p}\geq -1 \text{ Yi= }-1 \end{pmatrix} \qquad (2)$$

A hyperplane that generates the maximum limit is considered perfect. In the following equation, Si is the independent variable and C is the error penalty. The minimal solution for the hyperplane represents, as in Equ.3 and Equ.4 respectively [37]:

$$\emptyset(\mathrm{W,S}) = \frac{1}{2}(\mathrm{W.W}) + \mathrm{C}\sum_{i=0}^{1} Si \qquad (3)$$

Based on:

$$Y_i[(\mathrm{W.X}) + \mathrm{p}] \geq 1 - \mathrm{Si}\,, \mathrm{i} = 1,2,3 \ldots\ldots \mathrm{I} \qquad (4)$$

Where, $S_i$is the measurement of the distance between the sample xi and the limit on the opposite side of the limit. Using the following formula will simplify this calculation, as in Equ.5 [37]:

$$\mathrm{V}(\emptyset) = \sum_{i=1}^{1} \propto \mathrm{i} - \frac{1}{2}.\sum_{i,j=1}^{I} \propto \mathrm{i\ jYiY}\, jker(Xi, Xj) \qquad (5)$$

Based on, as in Equ.6:

$$\sum_{i=1}^{1} X_i Y_i = 0 \quad C \geq \propto \geq 0 \; i = 1,2,3 \ldots . \mathrm{I} \qquad (6)$$

The kernel function Ker (XiXj) returns the dot product of the feature space mappings of the original data points [37].

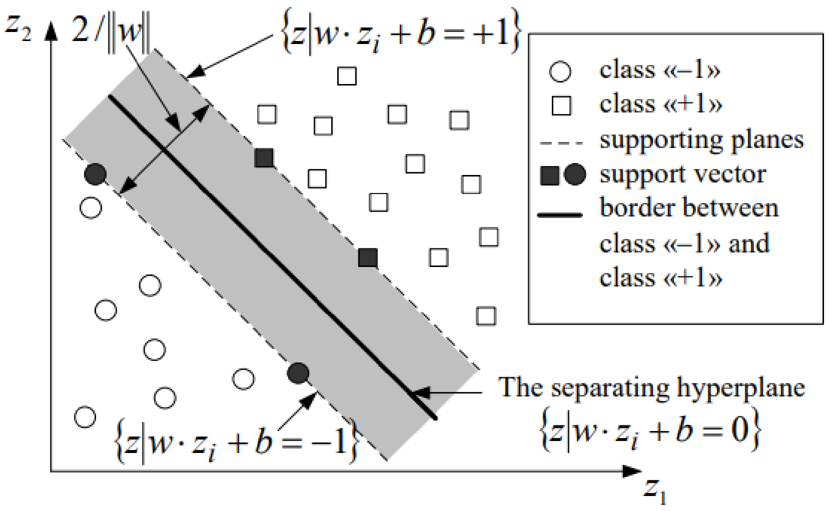
z_2
$2/\|w\|$
$\{z|w \cdot z_i + b = +1\}$
○ class «–1»
□ class «+1»
supporting planes
■● support vector
border between class «–1» and class «+1»
The separating hyperplane
$\{z|w \cdot z_i + b = 0\}$
$\{z|w \cdot z_i + b = -1\}$
z_1

Figure 6: Linear separations for two classes by the SVM classifier in the 2D space

b) Decision Tree

Decision trees are supervised machine learning algorithms that are considered the most common techniques used for regression and classification. Their goal is to create a model. Its goal is to learn basic rules derived from the data's properties in order to build a model for predicting a given value [27]. Its use of the if and then rule for prediction by classifying samples like tree structure depends on features. Due to its simple implementation, excellent classification accuracy, and understandable feature description, DT is widely utilized [38]. For example, as shown in Figure. 7, the model was trained using the dataset and can now categorize a decision regarding whether to play tennis as "yes" or "no". The decision nodes and leaf nodes make up this tree. Leaf nodes are connected to decision nodes by a number of branches. The classifications or decisions are represented by leaf nodes. The root node is the first beginning node from the top [39] Consequently, choosing an option from the available alternatives constitutes decision-making. If there are K classes in the classification problem and the sample's probability of belonging to the kth class is pk, the probability distribution's Gini index is defined as in Equations (7) and (8) [38],[40].

$$Gini(P) = \sum_{k=1}^{k} \mathrm{pk}(1 - \mathrm{p}_k) = 1 - \sum_{k=1}^{k} p_k^2 \qquad (7)$$

The Gini index for sample set D in the dichotomy issue is represented as

$$Gini(D) = 1 - \sum_{k=1}^{k} \left(\frac{|C_k|}{D}\right)^2 \qquad (8)$$

In this case, |D| denotes the total number of samples, while |Ck| denotes the number of samples in category k. The sample set's level of uncertainty is indicated by the Gini index. The higher the Gini index, the higher the level of sample uncertainty.

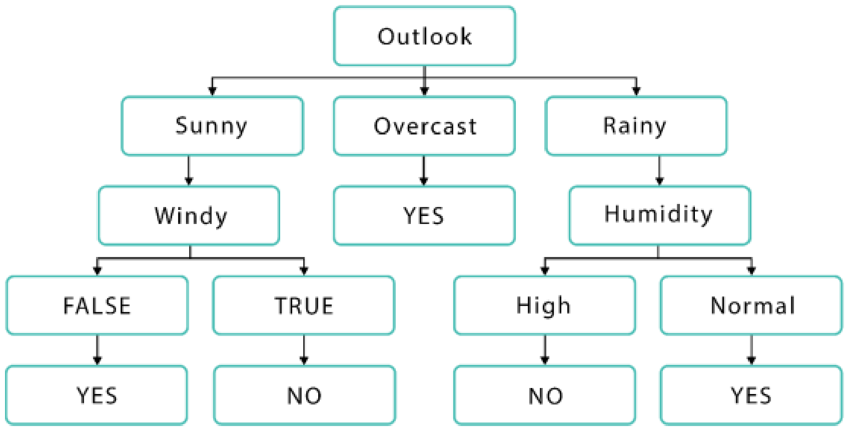

Figure 7: Decision tree example

c) K- Nearest Neighbour Algorithm (KNN)

The KNN is a machine learning algorithm that uses a group of datasets to make predictions and uses them for regression and classification. The algorithm uses a variable parameter called k, uses a voting rule after specifying the variable k, and locates the nearest number of feature neighbors to the k to make a predilection. The easiest and most straightforward classification approach is the nearest neighbor, and it is also the most effective. Comparable objects correspond to comparable layers, which is KNN's primary categorization principle. Choosing a grain using the delicate 1-NN classification rule can be mathematically represented, as in (9) [41]. For example, as shown in Figure 8, Since k for Query B is 3, it looks for the three neighbours who are closest to it. Of the three neighbours, two are of class 1 and one is of class 0, it finds. It then designates its class as 1 using the majority voting rule. Similar to the previous example, since k for Query A is 5 and more of its neighbours are classified as Class 0, it designates its class as 0 [42]. The algorithm has the ability to meet the growing demands of broadband services, optimize spectrum usage, and boost small cell power-transmission efficiency [43].

$$dist(x,y)(a,b) = \sqrt{(x-a)^2 + (y-b)^2} \tag{9}$$

Where, coordinates (x, y) and (a, b) is given by equation

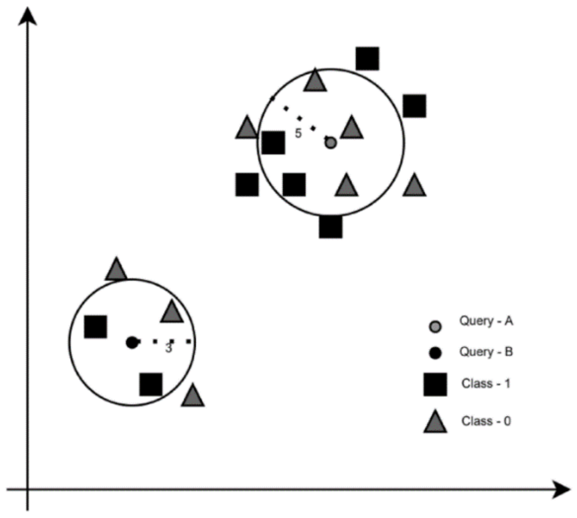

Figure 8: Visual illustration of the KNN algorithm

d) Random forest

is a type of supervised learning consisting of numerous distinct DTs, each trained separately using a different subset of data. RF generates several tree models using the training data set. Later stages can utilize these trees for prediction. It's considered an enhancement for the DT algorithm regarding the overfitting problem [44].

4.3.1.2 Unsupervised learning

A machine learning technique trains a dataset without specifying data for output, and the aim is to get effective information from the input data [45]. Some common unsupervised algorithms include:

a. K-means clustering

K-means is unsupervised technique for clustering data; it's a number of clusters for the input and an initial centroid randomly. The K-mean technique, however, is well-liked because of its efficiency and simplicity; it takes into account the user's input of K and randomly creates K points as initial centers, one for each cluster, before allocating each point to the nearest center based on distance can be mathematically represent , as in (10)[46]. k-means Clustering is the operation of splitting a dataset of features and patterns into different clusters. The patterns within the same cluster exhibit greater similarities than those in other clusters, and the efficiency of k-means primarily depends on the number of clusters. In k-means clustering, the maximum number of clusters is the number of features, and the minimum number is one cluster. and there are unique clusters that have their own features different from other ones, and the k-means is an iteration process that helps to specify the ideal number of clusters and categorize predictions based on cluster numbers [47]. In some cases, use k-means cluster in image's attributes and quality to sperate region of an image [48].

$$X = (x1, x2, x3 \dots xn)$$
$$Y = (y1, y2,33 \dots yn) \qquad (10)$$
$$d(X,Y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 .. + (x_m - y_m)^2}$$

Next, each cluster's center is updated by calculating its mean value; occasionally, points shift from cluster to cluster, but this only occurs at the approach end when nothing changes [46].

b. Hierarchical clustering

Is unsupervised learning algorithm cluster being represented by split data from the bottom to the top or upside down. It splits into large clusters and is divided based on the method used to create clusters. The cluster from top to bottom cluster strategy's break down cluster consists of all individuals into smaller clusters until each individual shapes a cluster on their own [49] . Agglomerative algorithms start with each element as a single cluster and aggregate them into progressively larger clusters. Starting with the complete set, dividing it into progressively smaller clusters that are then utilized to form clusters is the goal of a dividing algorithm [50].

5. Literature Survey

Indeed, other researchers have proposed various tactics, such as datasets, feature extraction, data preparation, and machine learning algorithms, to address the issue of malware detection. The subsequent paragraphs present an analysis of several of these pieces. The study [2], reviewed different machine learning techniques using datasets obtained from the Canadian Institute for Cyber Security and used these techniques to get and compare the best results of malware detection based on the highest accuracy and TPR and the lowest accuracy that had been obtained from FPR by using a confusion matrix. The classifiers that are used (KNN, CNN, NB, RF, SVM, DT) Hence, this paper demonstrates that the DT classifier gets the highest accuracy (99%), and then the CNN (98%) and SVM (96%) in the given dataset are compared. Also, A survey about dynamic malware analysis in the modern [3], has reviewed and surveyed dynamic analysis techniques, focusing on side channel analysis and volatile memory forensics, and compared these techniques with the early research about function call analysis execution control and flow tracking techniques working on malware behaviours that have evaded analysis and malicious activity. Analysis techniques designed to collect data about malware behaviour from side-channel signals such as electromagnetic emission and power consumption are done using hardware or software. Volatile memory acquisition is a methodology done by taking a copy from the RAM to external storage and then doing analysis on it. Like a memory dump, which is done by software or hardware, software uses bare metal layout, so in this paper survey, they use machine learning to enhance the performance and accuracy of these analysis techniques. Then in [27], They use static and dynamic analysis with different models of machine learning, and they notice that static analysis is more accurate than dynamic. Hence, static gets an accuracy of 99.36%, while dynamic gets 94.64%. The Cukoo sandbox is used by dynamic analysis; some malware has tricky behaviour that is difficult to detect by dynamic analysis, and there are limitations due to controlled network behaviour because these limitations are hard to analyze comprehensively. Moreover [51], uses three different techniques for malware detection. traditional signature-based, SVM by machine learning technique, and image processing through CNN by deep learning. Hence, CNN achieves the best result with an accuracy of 96.59%, followed by SVM with 95%, and the traditional signature-based method gets 94%. Also notice that the traditional signature-based method cannot detect changes in signatures, which means it cannot detect new malware, but the other techniques have the ability to detect these new ones. Whereas [52], In this project, we use different machine learning techniques, both supervised and unsupervised, and then make comparisons between these techniques to get the best accuracy result. Hence, random forest gets the best accuracy from other techniques, followed by XGBoost, decision tree, gradient bossing, and Adboost. There are different challenges faced by this project, including an imbalanced dataset, a lot of attributes, the training of the dataset to achieve best accuracy, and the issue with obtaining the dataset through data mining. Hence, to resolve these issues, we must:

- using several techniques of data preprocessing, like prepping the dataset, exploratory data analysis.
- tuning the models and making various changes to the data set.

In [53], proposed a hybrid IDS for the detection of malicious codes using data mining classification methods. This proposal employed data mining techniques, specifically anomaly detection and misuse detection, using two classifier models, Interactive Dichotomizer3 (ID3) and Naive Bayesain. The system's effectiveness was evaluated based on its accuracy rate. Misuse is similar to signature-based detection, which is considered more accurate than manually signature but not strong against zero-day attacks. On the other hand, anomaly detection has the ability to identify novel attacks known as zero-day attacks or new intrusion attacks. This proposal used three measures of feature selection: (1) association rules, (2) relief measures, and (3) gain ratio. Hence, the NB classifier model uses association rules to get the most accurate result with a 99% false positive rate (FPR) and a 1% FN rate. In [54] , proposes a network intrusion detection system (NIDS) using data mining techniques based on machine learning algorithms that help make predictions using two techniques. (1) Naivy Bayes's (2) multinominal logistic regression, using the KDDcup99 dataset, was recognized as the winner of the third international knowledge discovery and data mining tools competition in 1999. From the result discovery, these two techniques give high accuracy of prediction. Using cross-validation techniques to avoid overfitting, the false alarm rate decreases, and the NB classifier takes less time than multinomial logistic regression. In [55], uses recursive features based on deep learning and neural network algorithms that rely on the NSL-KDD dataset. This paper first uses a deep neural network for binary classification, achieving a 94% accuracy rate, and then employs a neural network for classification using these features (DOS, Normal, Probe, R2L, and U2R). As a result, they achieve good performance overall, particularly in novelty detection or outlier detection. The recommendation for future work is to use another dataset, another selection feature, and implement the system in real time. In [56], using classifier presence in several articles from 2017 to 2022 that are included in this review, they use different algorithms and techniques for smashing detection via machine learning; hence, depending on the result, the RF classifier gets the best result, followed by NB, DT, SVM, Adaboost, etc. It was used with two different datasets in [57], for high-performance detection using dense and LSTM-based deep learning with different versions. The results show that the performance level changes depending on the dataset. For the first dataset, the feature reduction rate is between 18.18 and 42.2%, and for the second dataset, it is between 81.77% and 93.5%. In the paper [58], "Detection of Malware by Deep Learning as CNN-LSTM Machine Learning Techniques in Real Time," the authors employed CNN-LSTM, a hybrid deep learning technique, to identify botnet attacks and software installed without the administrator's permission. The system mixes CNNs and LSTMs based on neural language processing by measuring the correlation of a variable on a dataset; hence, CNN-LSTM has a high accuracy of 99%, compared with the accuracy of DT at 98% and SVM at 95%. In [59] Signature-based ransomware detection based on optimization approaches using Random Classifier and CNN algorithms, machine learning techniques are used to find ransomware by general model. However, the results are not very good, so it is suggested to use a random classifier with the SMOTE optimizer and an ANN with root mean square propagation (Adam) using the Adam optimizer to lower the loss function while neural networks are being trained. So the Adam model reported 5.14 ms of prediction time and got an accuracy of 99.18%. Then in[60], Ransomware Detection Using Dynamic Analysis and Machine Learning: A Survey and Research Directions. This paper conducts a thorough examination of ransomware detection through machine learning, dynamic analysis, and deep learning. It concludes and suggests that for a robust and accurate ransomware detection system, a synergistic relationship between machine learning and deep learning is necessary to achieve optimal accuracy. However [61], conducted a survey on android malware detection techniques, utilizing machine learning algorithms rooted in frameworks and malware detection systems. Based on the survey, the majority of papers employed NB and SVM algorithms for malware detection. The OneR and J48 algorithms achieved the highest accuracy with 100% accuracy, while the average accuracy of the other algorithms ranged from 83% to 90%. In the surveyed papers, Multi-NB, SVM, QDBP, and TSDNN used the Avgprob scheme with a higher rate than 90%. Therefore, we recommend integrating them into a hybrid technique to enhance accuracy and detection rate. In [62] Detection of Malware in Android Phones Using Machine Learning, this paper experiments with neural networks (ANN) and multilayer perceptron (MLP), resulting in the highest accuracy of 92.26% compared to SVM. This paper, in [63], Malware detection using machine learning, presents machine learning on a Brazilian malware dataset with 57 different attributes of PE files and uses different algorithms; hence, from the analysis results, the Random Forest and Decision Tree models get the highest accuracy with 99.7%, followed by gradient boost 98.48%, SVM 96.9%, logistic regression 96.8%, XGBoost 96.7%, and at least AdaBoost 94.3%. This study, [64] Malware Detection Using Honeypot and Machine Learning, employs two distinct techniques. The study employs honeypot as a trap to identify malicious files, and employs machine learning to categorize these malware samples for detection. They employ decision trees and support vector machines as classifiers, and utilize the EMBER dataset, which comprises 300,000 malware samples and 300,000 benign malware samples. To achieve optimal results, they employ support vector machines to generate recommendations in the future, contrasting with unsupervised methods. In [65] From 2014 to 2021, researchers conducted a comprehensive survey on machine learning techniques for Android malware detection, focusing on analysis type, feature extraction method, and ML classification. The results revealed 14 static analysis types, 3 dynamic analysis types, and 2 hybrid analysis types. The feature extraction method consisted of 14 source code analyses, 8 manifest analyses, 2 network traffic analyses, 1 for each code instrumentation, system resource analysis, and user interaction analysis. In terms of machine learning techniques, we employed 15-based models, 7 ensemble learning, 6 feature importance, and 2 dimensionality reduction methods. In [66] The survey focuses on malware analysis and mitigation techniques. This research looks at signature-based and behavior-based approaches, such as dynamic analysis. It shows that signature-based approaches can't find APTs (advanced persistent threats), while dynamic approaches take longer to use and need more testing tools, such as sandboxes. This method also doesn't always find zero-day attacks. In [67] Machine Learning-Aided Static Malware Analysis: A Survey and Tutorial. This study conducted a survey for pe32 windows malware using machine learning techniques using static analysis; as a result, they found K-NN and C4.5 get better resalt and performance than other methods, while ANN and SVM get good performance on some features; furthermore, Nave Bayes and Bayes networks get weak performance compared with other techniques. In Static and Dynamic Malware Analysis Using Machine Learning [68], This survey juxtaposes the two analysis techniques, static and dynamic malware analysis, highlighting the limitations of dynamic analysis due to intelligent malware behaviors and its inability to operate with limited network access. Meanwhile, static analysis boasts an accuracy rate of 99.36%, outperforming dynamic analysis by 94.64%. In [69], this paper made a survey on 77 papers about machine learning algorithms for malware detection, and they found that when using a large dataset, the results were better with the

SVM and DT models than other models. They also noticed that dynamic analysis and hybrid analysis got better results than signature-based analysis. This paper used a behavior-based technique instead of signature-based [70] because the behavior-based approach was more able to detect malware. Hence, first they make dynamic analysis of the dataset inside the virtual environment, capture API calls, trace them, and then generate high-level features. They responded to this by applying machine learning models like DT, SVM, and random forest. This technique's experimental results show high accuracy in detecting variant malware.

Table.1: Outline of the experiments surveyed works.

work	**year**	**analysis**	**dataset**	**techniques**
[2]	2022	static analysis	Canadian Institute for Cyber Security	KNN, CNN, NB, RF, SVM, DT
[3]	2019	dynamic analysis	collect data about malware behaviour from side-channel signals such as electromagnetic emission and power consumption	side channel analysis and volatile memory forensics
[27]	2018	static and dynamic analysis	N/A	Static and sandbox technique
[51]	2022	Analysis using svm, and signature-based, Dynamic-based method for CNN	Dataset used for CNN method contains around 200 files which are converted binary files of malwares. Dataset total 15036 samples data is available. From that 5560	signature-based, SVM by machine learning technique, and image processing through CNN by deep learning

			are Malware cases and 9476 are benign cases.	
[52]	2023	Feature extraction	The dataset has about 130000 and 57 columns (features).	Random forest gets the best accuracy from other techniques, followed by XGBoost, Decision Tree, Gradient bossing, and Adboost
[53]	2013	Feature Selection	KDD'99 dataset	data mining techniques, Interactive Dichotomizer3 (ID3) and Naïve Bayesain
[54]	2020	Feature Selection	KDD99 dataset	Data mining by Logistic Regression and Naïve bayes
[55]	2021	Normalization and Feature selection	NSL-KDD dataset	Deep Neural Network (DNN) and Recurrent Neural Network (RNN)
[56]	2023	content-based, URL behavior analysis, and heuristic	The model was validated by experiments on both the English and non-English datasets	the RF classifier get best result then flowed by (NB, DT, SVM, Adaboost …etc).
[57]	2023	Feature selection using correlation degree between each attribute and the target column	Unix/Linux-based platform was used to build this dataset and Android Malware Dataset made up of 215 distinct attributes	dense and LSTM-based deep learning
[58]	2022	Feature extraction and selection by measure the	Datasets collected for website Kaggle	CNN-LSTM Machine Learning Techniques

		correlation of a variable on dataset		
[59]	2023	Signature-based analysis	N/A	random classifier with SMOTE optimizer and use ANN using Root Mean Square propagation (Adam)
[60]	2022	Dynamic analysis	N/A	Machine learning and deep learning
[61]	2019	Static and Dynamic analysis	N/A	Multi-NB, SVM, QDBP and TSDNN using Avgprob scheme with higher rate than 90%
[62]	2022	Feature extracted	N/A	neural networks ANN and multilayer perceptron MLP
[63]	2020	Data preprocessing and feature selection	Brazilian malware dataset	Random Forest and Decision tree models get the highest accuracy with 99.7% then sequenced followed by gradient boost 98.48% , SVM 96.9 %, logistic regression 96.8%, XGBoost 96.7% and at least AdaBoost 94.3%.
[64]	2019	Data analysis by ML	dataset used called (EMBER)	Honeypot, Decision Tree and Support vector machine
[65]	2021	Static, dynamic and hybrid analysis	N/A	Base models Ensemble learning Feature importance Dimensionality reduction

[66]	2019	Static and dynamic analysis	N/A	Signature-based and behavior-based
[67]	2018	Static analysis	N/A	K-NN, C4.5, ANN, SVM and ANN and SVM
[68]	2019	Static and Dynamic Malware Analysis	N/A	ML algorithms
[69]	2022	dynamic analysis and hybrid get better result more than signature-based analysis.	N/A	the result by better with SVM and DT model than other models
[70]	2016	Dynamic analysis	N/A	DT, SVM, Random Forest

Conclusion

According to our survey, static analysis is a technique that uses detection patterns to examine source code without executing it, and these patterns include operational code, string signatures, code complexity, etc. Using a debugger, disassembler, or memory dumper tool to reverse compile a Windows executable is necessary, but static analysis alone is insufficient. Therefore, we developed dynamic analysis to supplement static analysis. In this technique, the malware code is executable in a safe environment, and different tools monitor the malicious code's behavior before executing it. depends on file system and registry monitors, process monitoring, and network monitoring, but this technique is not enough for detection, so utilize a technique called machine-learning based on AI and artificial intelligence using different variants of models and classifiers. train and test using several datasets to predict the best results about these malicious programs, and depending on our survey about these models, there is no one special form or another. All is important, but it depends on the case and the function wanted, as well as how to prepare data to predict the best results about these malicious programs, and depending on our survey about these models, there is no one special form or another. All is important, but it depends on the case and the function wanted, as well as how to prepare data for testing. All these factors have high effects, but there are classifiers that are more developed and used to get the best results for others, like KNN, CNN, NB, RF, SVM, DT, etc., so the researchers must keep trying and developing datasets and using different techniques to get the best result.

References

[1] P. Singh, S. Tapaswi, S. Gupta, Malware Detection in PDF and Office Documents: A survey, Inf. Secur. J., 29(2020)134–153, https://doi.org/10.1080/19393555.2020.1723747

[2] M. Akhtar, T. Feng, Malware Analysis and Detection Using Machine Learning Algorithms, Symmetry., 14(2022)2304. https://doi.org/10.3390/sym14112304

[3] O. Or-Meir, N. Nissim, Y. Elovici, L. Rokach, Dynamic Malware Analysis in the Modern Era—A State of the Art Survey, ACM Comput., 52(2020)48, https://doi.org/10.1145/3329786

[4] M. Imran, A. Appice, D. Malerba, Evaluating Realistic Adversarial Attacks against Machine Learning Models for Windows PE Malware Detection, Futur. Internet J., 16(2024)168, https://doi.org/10.3390/fi16050168

[5] A. Ali Almazroi, N. Ayub, Deep learning hybridization for improved malware detection in smart Internet of Things, Sci Rep J., 14(2024)7838, https://doi.org/10.1038/s41598-024-57864-8

[6] S. Shafin, G. Karmakar,I. Mareels, Obfuscated Memory Malware Detection in Resource-Constrained IoT Devices for Smart City Applications., Sensors, 23(2023) 5348, https://doi.org/10.3390/s23115348

[7] A. Ehsan, C. Catal, A. Mishra, Detecting Malware by Analyzing App Permissions on Android Platform: A Systematic Literature Review, Sensors, MDPI., 22(2022)20, https://doi.org/10.3390/s22207928

[8] S. Smmarwar, G. Gupta, S. Kumar, Android malware detection and identification frameworks by leveraging the machine and deep learning techniques: A comprehensive review, Telemat. Inform. Rep., 14(2024)100130 https://doi.org/10.1016/j.teler.2024.100130

[9] H. Bostani V. Moonsamy, EvadeDroid: A practical evasion attack on machine learning for black-box Android malware detection, Comput. Secur. j., 139(2024) 103676, https://doi.org/10.1016/j.cose.2023.103676

[10] B. Khammas, Ransomware Detection using Random Forest Technique, ICT Express., 6(2020) 325-331, https://doi.org/10.1016/j.icte.2020.11.001

[11] A. Nasser, A. Hasan, A. Humaidi, DL-AMDet: Deep learning-based malware detector for android, Intell. Syst. Appl. J., 21(2024)200318, https://doi.org/10.1016/j.iswa.2023.200318

[12] S. Poornima, R. Mahalakshmi, Automated malware detection using machine learning and deep learning approaches for android applications, Meas. Sens., 32(2024) 100955, http://dx.doi.org/10.1016/j.measen.2023.100955

[13] E. Gandotra, D. Bansal, S. Sofat, Malware Analysis and Classification: A Survey, J. Inf. Sec., 5(2014)56-64, http://dx.doi.org/10.4236/jis.2014.52006

[14] F. Nawshin , R. Gad , D. Unal , A. Khalid, P. Suganthan, Malware detection for mobile computing using secure and privacy-preserving machine learning approaches: A comprehensive survey, Comput. Electr. Eng. J., 117(2024) 109233 https://doi.org/10.1016/j.compeleceng.2024.109233

[15] N. Usman, S. Usman, F. Khan, M. A. Jan, A. Sajid, M. Alazab, P. Watters, Intelligent Dynamic Malware Detection using Machine Learning in IP Reputation for Forensics Data Analytics, Future Gener, Comput. Syst., 118(2021)124-141, https://doi.org/10.1016/j.future.2021.01.004.

[16] A. Bensaoud, J. Kalita, M. Bensaoud, A survey of malware detection using deep learning, Mach. Learn, Mach. Learn. Appl., 16(2024)100546, https://doi.org/10.1016/j.mlwa.2024.100546

[17] A. Bensaoud, J. Kalita, M. Bensaoud, A survey of malware detection using deep learning, Mach. Learn. Appl., 16(2024)100546, https://doi.org/10.1016/j.mlwa.2024.100546

[18] M. Saqib, S. Mahdavifar, B. C. M. Fung, P. Charland, A Comprehensive Analysis of Explainable AI for Malware Hunting, ACM Comput Surv., (2024) https://doi.org/10.1145/3677374

[19] S. Zhang, J. Wu, M. Zhang, W. Yang, Dynamic malware analysis based on API sequence semantic fusion, Appl. Sci.., 13(2023) 6526, https://doi.org/10.3390/app13116526

[20] S. Xiong and H. Zhang, A Multi-model Fusion Strategy for Android Malware Detection Based on Machine Learning Algorithms, J. Comput. Sci. Res., 6(2024)1-11, https://doi.org/10.30564/jcsr.v6i2.6632

[21] L. Ce, C. Zijun, Z. He, W. Leiqi, L. Qiujian, W. Yan, L. Ning, S. Degang, DMalNet: Dynamic malware analysis based on API feature engineering and graph learning, Computers & Security., 122(2022)102872 https://doi.org/10.1016/j.cose.2022.102872

[22] Z. Zhang, P. Qi,W. Wang, Dynamic malware analysis with feature engineering and feature learning, Cornell University., 5(2019)1-9, https://doi.org/10.48550/arXiv.1907.07352

[23] M. Stamp, M. Alazab, A. Shalaginov, Malware analysis using artificial intelligence and deep learning, Springer International Publishing, 2020, https://doi.org/10.1007/978-3-030-62582-5

[24] A. Abusitta, M. Li, B. Fung, Malware classification and composition analysis: A survey of recent developments, J. Inf. Secur. Appl., 59(2021) 102828. https://doi.org/10.1016/j.jisa.2021.102828

[25] A. Abdulrahman, M. Ibrahem, Intrusion detection system using data stream classification, Iraqi J. Sci., 62(2021)319-328. https://doi.org/10.24996/ijs.2021.62.1.30

[26] R. Najeeb, B. Dhannoon, Improving detection rate of the network intrusion detection system based on wrapper feature selection approach, Iraqi J. Sci., 59(2018)426-433, http://dx.doi.org/10.24996/ijs.2018.59.1B.23

[27] F. Nawshin, D. Unal, M. Hammoudeh, P. Suganthan, AI-powered malware detection with Differential Privacy for zero trust security in Internet of Things networks, Ad Hoc Netw., 161(2024)103523, https://doi.org/10.1016/j.adhoc.2024.103523

[28] O. Aslan, R. Samet, A Comprehensive Review on Malware Detection Approaches, IEEE Access., 8(2020)6249–6271, https://doi.org/10.1109/ACCESS.2019.2963724

[29] A. Aboaoja, A. Zainal, A. Ghaleb, B. Ali, T. Abdalla, and A. Abbas, Malware Detection Issues, Challenges, and Future Directions: A Survey, Appl. Sci., 12 (2022) 8482, https://doi.org/10.3390/app12178482.

[30] M. Chemmakha, O. Habibi, M. Lazaar, Improving Machine Learning Models for Malware Detection Using Embedded Feature Selection Method, IFAC-PapersOnLine., 55(2022)771-776, https://doi.org/10.1016/j.ifacol.2022.07.406 .

[31] K. Bhat, T. Khairnar, S. Phatangare,T. Narkhedkar, Malware detection using machine learning, IJRASET., 11(2023) 2607-2613 https://doi.org/10.22214/ijraset.2023.52217

[32] A. Muzaffar, H. Ragab Hassen, M. A. Lones, H. Zantout, An in-depth review of machine learning based Android malware detection, Comput. Secur., 121(2022) 102833, https://doi.org/10.1016/j.cose.2022.102833

[33] M. Wadkar, F. Di Troia, M. Stamp, Detecting malware evolution using support vector machines, Expert Syst. Appl., 143 (2020) 113022, https://doi.org/10.1016/j.eswa.2019.113022.

[34] C. Pegoraro, A. Savino, S. Di Carlo, A Survey on Hardware-Based Malware Detection Approaches, IEEE Access., 12(2024)54115-54128, https://doi.org/10.1109/ACCESS.2024.3388716

35] A. Savita; C. Amit, Hybrid CNN-SVM classifier for handwritten digit recognition, In Procedia Comput. Sci., 167(2020)2554-2560, https://doi.org/10.1016/j.procs.2020.03.309

[36] L. Demidova, E. Nikulchev, Y. Sokolova, The SVM Classifier Based on the Modified Particle Swarm Optimization, Int. J. Adv. Comput. Sci. Appl., 7(2016)16-24, https://dx.doi.org/10.14569/IJACSA.2016.070203.

[37] A. Hilool, S. Hashem, S. Jafer, Building an efficient system to detect computer worms in websites based on ensemble Ada Boosting and SVM classifiers algorithms, Eng. Technol. J., 40(2022)595-604, https://doi.org/10.30684/etj.v40i4.2148

[38] M. Yang, X. Chen, Y. Luo, H. Zhang, An Android malware detection model based on DT-SVM, Secur. Commun. Netw., (2020)1-11 https://doi.org/10.1155/2020/8841233

[39] M. Azeem, D. Khan, S. Iftikhar, S. Bawazeer, and M. Alzahrani, Analyzing and comparing the effectiveness of malware detection: A study of machine learning approaches, Heliyon., 10(2024) e23574, https://doi.org/10.1016/j.heliyon.2023.e23574

[40] M. Kashmoola, M. Ahmed, A. Alsaleem, Network traffic prediction based on boosting learning. Iraqi J. Sci., 63(2022)4047-4056. https://doi.org/10.24996/ijs.2022.63.9.33

[41] S. Uddin, I. Haque, H. Lu, M. Moni, E. Gide, Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction. Sci. Rep., 12(2022)1-11. https://doi.org/10.1038/s41598-022-10358-x

[42] S. Hassan, S. Rubab, S. Saadat, S. Qasim, M. Attique, A Alasiry, M. MarzA, KNN learning algorithm for collusion-resistant spectrum auction in small cell networks, IEEE Access., 6(2018)45796-45803. http://dx.doi.org/10.1109/ACCESS.2023.3321221

[43] F. Zhao, Q. Tang, A KNN learning algorithm for collusion-resistant spectrum auction in small cell networks, IEEE Access., 6(2023)45796-45803. https://doi.org/10.1109/ACCESS.2023.3321221

[44] M. Saadi, B. Dhannoon, Comparing the Random Forest vs. Extreme Gradient Boosting using Cuckoo Search Optimizer for Detecting Arabic Cyberbullying. Iraqi J. Sci, 64(2023)4806-4818. https://doi.org/10.24996/ijs.2023.64.9.40

[45] A. Gliemlo, B. Husic, E. Rodriguez, A. Clementi, C. Noé, A. Laio, Unsupervised learning methods for molecular simulation data. Chemical Reviews., 121(2021)9722-9758. http://dx.doi.org/10.1021/acs.chemrev.0c01195

[46] Y. Hussein, S. A. Jalil, Proposed KDBSCAN algorithm for clustering. Iraqi J. Sci, 59(2018)173-178. http://dx.doi.org/10.24996/ijs.2018.59.1A.18

[47]T. Scharr, R. Pimenta, F. Ceschin, and A. Gregio, Reviewing Clustering Techniques for Android Malware Family Classification, ACM Journals., 5(2024)1-35, https://doi.org/10.1145/3587471

[48] E. Hammadi, A. Hussein, R. Mohammad, W. Ahmad, M. Salih, Using K-mean clustering to classify the kidney images. Iraqi J. Sci., 64(2023)2070-2084. https://doi.org/10.24996/ijs.2023.64.4.41

[49] S. Hera, M. Amjad, Prediction of explicit features for recommendation system using user reviews, Iraqi J. Sci., 63(2022)5015-5023. https://doi.org/10.24996/ijs.2022.63.11.36

[50] R. Sembiring, J. Zain, and A. Embong, comparative agglomerative hierarchical clustering method to cluster implemented course, J. Comput., 2(2010)1101-4270, https://doi.org/10.48550/arXiv.1101.4270

[51] P. Maniriho, A. Naser, M. Jabed, MeMalDet: A memory analysis-based malware detection framework using deep autoencoders and stacked ensemble under temporal evaluations, Comput. Secur. J., 142(2024)103864, https://doi.org/10.1016/j.cose.2024.103864

[52] A. Kamboj, P. Kumar, Bairwa, A. Bairwa, S. Joshi, Detection of malware in downloaded files using various machine learning models, Egypt. Inform. J., 24(2023)81-94. https://doi.org/10.1016/j.eij.2022.12.002

[53] S. H Hashim, I. A Abdulmunem, A proposal to detect computer worms (malicious codes) using data mining classification algorithms, Eng. Technol. J., 31(2013), 142-155, https://doi.org/10.30684/etj.31.2B.3

[54] S. Shareef, S. Hashim, Proposed hybrid classifier to improve network intrusion detection system using data mining techniques, Eng. Technol. J., 38(2020)6-14, https://doi.org/10.30684/etj.v38i1B.149

[55] B. Mohammed, E. Gbashi, Intrusion detection system for NSL-KDD dataset based on deep learning and recursive feature elimination, Eng. Technol. J., 39(2021)1069-1079, https://doi.org/10.30684/etj.v40i1.1933

[56] A. Mahmood, S. Hameed, Review of smishing detection via machine learning. Ijs., 64(2023)4244-4259. https://doi.org/10.24996/ijs.2023.64.8.42

[57] E. Alomari, M. Al-Kabi, S. Al-Jawarneh, M. Al-Ayyoub, and I. Aljarah, Malware detection using deep learning and correlation-based feature selection. Symmetry., 15(2023)123, https://doi.org/10.3390/sym15010123

[58] M. Akhtar, T. Feng, Detection of malware by deep learning as CNN-LSTM machine learning techniques in real time, Symmetry., 14(2022)2308, https://doi.org/10.3390/sym14112308

[59] K. Sangher, A. Singh, H. Pandey, Signature based ransomware detection based on optimizations approaches using RandomClassifier and CNN algorithms, International Journal of System Assurance Engineering and Management., 15(2023)1687-1703, https://doi.org/10.21203/rs.3.rs-2716621/v1

[60] U. Urooj, B. Al-Rimy, A. Zainal, F. Ghaleb, M. Rassam, Ransomware detection using dynamic analysis and machine learning: A survey and research directions, Appl. Sci., 12(2022)172, https://doi.org/10.3390/app12010172

[61] E. Alqahtani, R. Zagrouba, and A. Almuhaideb, A survey on Android malware detection techniques using machine learning algorithms, International Conference SDS., (2019)110-117,. https://doi.org/10.1109/SDS.2019.8768729

[62] K. Poojitha, J. Usha, Detection of malware in Android phones using machine learning, IJRASET., 10(2022)3344-3347, https://doi.org/10.22214/ijraset.2022.45726

[63] A. Kumar, K. Abhishek, K. Shah, D. Patel, Y. Jain, H. Chheda, and P. Nerurkar, Malware detection using machine learning, GmbH., 1232(2020)61-71, http://dx.doi.org/10.1007/978-3-030-65384-2_5

[64] I. Matin, B. Rahardjo, Malware detection using honeypot and machine learning, CITSM., (2019)1-5, https://doi.org/10.1109/CITSM47753.2019.8965419

[65] V. Kouliaridis, G. Kambourakis, A comprehensive survey on machine learning techniques for Android malware detection, Information, 12(2021)185, http://dx.doi.org/10.3390/info12050185

[66] S. Sibi Chakkaravarthy, D. Sangeetha, V. Vaidehi, A Survey on malware analysis and mitigation techniques, Comp. Sci. Rev., 32(2019)1–23, https://doi.org/10.1016/j.cosrev.2019.01.002

[67] A. Shalaginov, S. Banin, A. Dehghantanha, K. Franke, Machine Learning Aided Static Malware Analysis: A Survey and Tutorial, 70(2018)7-45, http://dx.doi.org/10.1007/978-3-319-73951-9_2

[68] M. Ijaz; M. Hanif Durad,M. Ismail, Static and Dynamic Malware Analysis Using Machine Learning, IBCAST., (2019)687-691, https://doi.org/10.1109/IBCAST.2019.8667136

[69] N. Gorment, A. Selamat, L. Cheng, O. Krejcar, Machine learning algorithm for malware detection: taxonomy, current challenges and future directions, IEEE Access., 11(2023)141045-141089, https://doi.org/10.1109/ACCESS.2023.3256979

[70] H. Galal, Y. Mahdy, M. Atiea, Behavior-based features model for malware detection, J. Comput. Virol. Hacking Tech., 12(2016)59–67, http://dx.doi.org/10.1007/s11416-015-0244-0

مراجعة حول عن الكشف البرامج الخبيثة باستخدام تقنيات الذكاء الاصطناعي

, سهاد فيصل شيحان محمد سعدون عبدالزهرة

كلية علوم الحاسوب، جامعة بغداد، بغداد، العراق

المستخلص

في يومنا الحالي، الامن السيبراني يعتبر من الأولويات الأساسية التي يجب ان تأخذ بنظر الاعتبار حيث ان المخاطر الالكترونية اصحبت واقع حال في حياتنا اليومية. في هذا البحث سوف نوضح مفاهيم البرامج الخبيثة والتي هي جزء من الامن السيبراني وملخص حول البرامج الخبيثة التي تستخدم لسرقة البيانات واختراق خصوصية وأمان المستخدم. اليوم الجرائم السيبرانية تستخدم تقنيات وبرامج خبيثة متعددة الاشكال من الصعب كشفها باستخدام تقنيات الكشف التقليدية. بالتالي في هذا البحث سوف نقوم بعمل استبيان حول استخدام خوارزميات الذكاء الاصطناعي والتي سوف تساعدنا في اكتشاف مثل هذا النوع من الهجمات والبرامج الخبيثة للحصول على أفضل دقة وأداء ممكن للكشف عنهم.