Trojan Horse Infection Detection in Cloud Based Environment Using Machine Learning

https://doi.org/10.3991/ijim.v16i24.35763

Hasan Kanaker[1(✉)], Nader Abdel Karim[2], Samer A.B. Awwad[3], Nurul H.A. Ismail[4], Jamal Zraqou[5], Abdulla M. F. Al ali[6]
[1] Department of Cyber Security, Isra University, Amman, Jordan
[2] Department of Intelligent Systems, Al-Balqa Applied University, Al-Salt, Jordan
[3] Department of Quality Assurance, Imam Abdulrahman Bin Fisal University, Dammam, Saudi Arabia
[4] Department of Computer Science and Information technology, Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia
[5] Department of Computer Science, University of Petra, Amman, Jordan
[6] Department of Courses Service, Isra University, Amman, Jordan
hasan.kanaker@iu.edu.jo

Abstract—Cloud computing technology is known as a distributed computing network, which consists of a large number of servers connected via the internet. This technology involves many worthwhile resources, such as applications, services, and large database storage. Users have the ability to access cloud services and resources through web services. Cloud computing provides a considerable number of benefits, such as effective virtualized resources, cost efficiency, self-service access, flexibility, and scalability. However, many security issues are present in cloud computing environment. One of the most common security challenges in the cloud computing environment is the trojan horses. Trojan horses can disrupt cloud computing services and damage the resources, applications, or virtual machines in the cloud structure. Trojan horse attacks are dangerous, complicated and very difficult to be detected. In this research, eight machine learning classifiers for trojan horse detection in a cloud-based environment have been investigated. The accuracy of the cloud trojan horses detection rate has been investigated using dynamic analysis, Cukoo sandbox, and the Weka data mining tool. Based on the conducted experiments, the SMO and Multilayer Perceptron have been found to be the best classifiers for trojan horse detection in a cloud-based environment. Although SMO and Multilayer Perceptron have achieved the highest accuracy rate of 95.86%, Multilayer Perceptron has outperformed SMO in term of Receiver Operating Characteristic (ROC) area.

Keywords—cloud computing, trojan horse attacks, dynamic analysis, detection, machine learning

1 Introduction

Cloud computing is a revolution in IT technology that usually uses remote servers and the internet in order to provide a shared pool of computing resources and applications for its users' requirements. The beginning of this technology was started in the 2000s as the last stage of IT infrastructure development, which indicated a new computing paradigm where individuals gain software solutions and computing power over the networks or Internet [1]. Moreover, cloud computing has appeared as a new model for hosting and distributing services over the Internet [2], [3]. With cloud computing, users have access to the cloud services at their locations so that they can access the data relevant to their tasks without interfering with other people's work and allocated resources. In addition, they have the ability to run their applications on many connected computers at the same time. Furthermore, cloud computing enables users to access resources and applications from anywhere and at any time via internet connectivity, without the need to install applications on their personal systems [2], [4].

Recently, cloud computing has become increasingly attractive and has gained the growing interest of institutions and IT industries all around the world. Hence, they have started to migrate many of their core business functions into cloud platforms [3], [4]. Furthermore, governments have become concerned about the potential of using cloud computing to decrease IT costs and increase the reachability of their delivered services [5], [6]. The cloud technology consists of a full range of IT infrastructures such as servers, applications, email, file storage, large resources, database and services, an available through a network, usually the Internet [7].

Many public-sector organizations in various countries [8], such as the United States (US) government, the United Kingdom (UK), Australia, and other European countries, have utilized cloud computing to gain its benefits [6], [9]. In 2010, the UK national government introduced the Government Cloud (G-Cloud) infrastructure where it was expected that significant savings of around £3.2 billion could be made by transferring to this service [8], [10], [11]. While the US government launched the Cloud Computing Mall in 2009 [11].

Cloud computing technology has many benefits and features, which are: cost saving, shared resource pooling, multi-tenancy, dynamic and massive scalability, elasticity, self-provisioning of resources, and pay-as-you-go [12], [13]. Although cloud computing provides these great features and benefits, it still has many security issues and attacks that could affect it. Security attacks are the major concern for cloud computing. Hence, these attacks trigger concern for both the service providers and the users [14].

There are numerous kinds of possible attacks, such as man-in-the middle attacks, authentication attacks [15], denial-of-service attacks, phishing attacks, and malware injection attacks [2]. Malware attacks (e.g. Trojan horses, viruses, worms, etc.) are considered one of the biggest threats facing the cloud computing environment. Recently, the Trojan horse emerged as a serious issue and increased its harmful power for computer systems and in the cloud computing environment [16], [17].

A Cloud Trojan injection attack is a malicious program that can be uploaded into cloud systems and be able to cause damage. Furthermore, these malicious programs can be embedded in a legitimate command, transmitted to clouds, and executed as legitimate instances [18].

Moreover, a cloud Trojan injection attack is an attack that attempts to inject a spiteful service, virtual machine, or even application into the cloud system and has the ability to affect the cloud services by blocking or altering cloud functionalities [19], [20]. The attacker tries to create his own malicious service, application, or virtual machine instance and then add it to the cloud system. Once the malicious code has been added to the cloud system, the attacker tricks the cloud system into treating the malicious code as a valid instance. Once it is successful, regular users are capable of demanding the malicious service instance, and then the malicious code is executed [19].

Another situation of this attack, an attacker tries to upload a trojan program or virus to the cloud system. Once the cloud system treats it as a legitimate service, the virus and trojan program is automatically executed and infected the cloud system which can cause damage and harm to the cloud system. In the case of the virus and trojan damage the hardware of the cloud system, the other cloud instances which is running on the same hardware may affect to the trojan and virus program because they share the same hardware. As well, the attacker might aim to utilize a Trojan and virus program to attack and assault the further users on the cloud system. Once a client demands the malicious program instance, the cloud system sends the virus through the internet to the client and after that executes on the client's device. The client's computer then is infected by the virus [18] – [20].

Since polymorphism is one of the characteristics of Trojan horses, it is difficult for the signature-based technology to detect them, which is the most commonly used method in existing anti-virus programs [21]. Furthermore, signature-based antivirus is able to detect with extremely high accuracy when the signature is well-known. However, the limitation of this type of detection is that when the malware alters its signature absolutely, Generally, this kind of antivirus would not be able to detect a novel attack.

In this research, trojan horses' attacks implications in the cloud computing environment has been investigated. In-depth study has been conducted to highlight the way that the trojan horse attacks in the cloud computing environment work. Dynamic analysis and Cuckoo Sandbox have been used to analysis the cloud trojan horses. Based on this analysis, the significant patterns are identified and the important features of these cloud trojan horses are extracted. Then, to enhance cloud trojan horses detection, various machine learning algorithms have been evaluated based on their ability to detect and classify the cloud trojan horses in the targeted dataset. This evaluation has been conducted using WEKA tool. This study will be significantly beneficial for other researchers as the foundation for developing effective techniques for detecting trojan horse attacks in the cloud computing environment.

This research paper is presented as follows: initially, research overview including types of malwares, related works and machine learning models have been presented in Section 2 and Section 3 respectively. After that, malware analysis techniques have been explained in Section 4 while research methodology has been described in Section 5. Section 6 has presented experimental results and discussion. Finally, conclusion has been presented in section 7.

2 Research overview

This section describes the various types of malwares, as well as some related research on malware detection in cloud infrastructure using Machine Learning methodologies and multiple Machine Learning models used in our work.

2.1 Types of malwares

Malware is defined as malicious software that is installed without permission on a machine in order to infect and harm that machine, as well as carry itself to legitimate programs and spread. There are many types of malware threats [22], including Trojans, viruses, worms, ransomware, spyware, and others.

Ransomware. Ransomware is malicious software that holds the computer's data and system hostage. It can affect the computer by encrypting the files or data stored on the machine with a key that is already unknown to the user. After paying the ransom, the victim user can resume using his or her system [23]. Ransomware has the ability to infect its victim's targets through Trojans [24].

Adware. Adware is annoying malware that automatically shows, plays, or downloads announcements on a user's computer when he or she is online without their authorization and has the ability to interrupt their existing activity. The primary goal of the adware is to generate financial benefits and revenue for its author [23], [25]. On the other hand, adware is occasionally categorized as spyware due to the severity of the recording. In addition, some adware might come through integrated spyware like keyloggers and other software that violates privacy [23].

Spyware. Spyware is utilized to steal confidential information from a computer system and transmit it to a third party, or preserve a watch on a user's events and have the ability to gather information and send it to the hackers. It is installed without the awareness of the system owner and stealthily collects the information and forwards it back to the hacker [25].

Keyloggers. It is a type of spyware that is utilized to record keystrokes in order to steal credit card details (e.g., ATM card numbers), passwords, and other significant and confidential information. It can be transferred to a computer once the user visits an infected site or through some other malicious program that is installed on the user's computer [25]. The recording is saved in a log file, which is typically encrypted and sent to a particular receiver [23].

Rootkits. This is known as malicious software or a collection of software tools utilized by attackers to gain persistent administrator level access and designed to give unauthorized access to a computer system so as to camouflage the altering of files [23]. Rootkits have the ability to control the operating system and hide themselves in the system, as well as provide a secure environment for other malware to evade antivirus and deliberate them as common applications [25].

Virus. A virus is a kind of malware that can infect computers and files by replicating itself over a network. It can cause serious damage to the computer system, such as degradation of the system performance, modification or deletion of data, and denial of service [23], [25]. A virus is a malicious executable code attached to another executable file. The virus spreads when an infected file is passed from system to system [26]. Once a program virus is active, it will infect other programs on the computer. A virus can be transmitted to other computers in many different ways, such as by inserting copies of infected files into a removable medium such as a USB drive, DVD, or CD, or by sending an infected file as an email attachment [26], [27].

Backdoor. Backdoor is a category of malware that provides an auxiliary stealthy arrival to the system; attackers bypass the habitual authentication used to access the system. The main characteristic of the backdoor is that it opens the door for attackers to cause destruction [23]. In addition, the backdoor also has another feature that grants cybercriminals future entrance to the system even if the organization repairs the original vulnerability used to attack the system.

Sniffers. Sniffers are programs that monitor, analyze, and capture any data passing over a network [25]. Sniffers access information through network interface cards (NICs). After a system has been infiltrated, attackers utilize sniffers to capture passwords and other system information.

Botnet. It is a kind of malware that is also known as a Bot, which is a piece of software that allows the attackers to gain access and control of the infected computer system to do harmful and malicious activities such as steal information, spam messages, denial of service attacks (DoS), and Distributed Denial of Service (DDoS) attacks without the awareness of legitimate users [23], [25]. Bots have the ability to propagate over backdoors that are made available by a virus or worm on the target computer [23]. By using several bots, Distributed Denial of Service (DDoS) attacks, which have the ability to hamper the services of the target computer machine through over-saturating its resources or bandwidth with requests, can be launched [28].

Worm. A worm is a malicious program that can disrupt systems and applications by morphing their primary codes, causing those systems and applications to disintegrate and become unusable [2], [7]. It can multiply, infect, and spread without being attached to a host [8]. The worm is designed to infect another machine by self-replicating and copying itself without the need for human intervention. Worms, unlike viruses, can infect without the need for a pre-existing program [29]. Self-replication refers to the worm's ability to duplicate itself, and its ability to run independently of other software [29]. The worm spreads over a computer network to reach the target system. Many worms have the purpose of stealing data, erasing it, and then spreading to new computers [30]. The worm could have a significant negative impact on network systems, such as using too much system memory or processor (CPU) and causing numerous apps to cease responding [31]. A worm can carry malicious code or be used to install other types of malwares on a computer (e.g., adware).

Trojan Horse. A Trojan is malicious software that disguises itself as beneficial or legal software. Cybercriminals utilize it to get access to users' computers. Users are frequently fooled by social engineering, which results in Trojans being installed unknowingly on their systems [30]. A Trojan Horse appears to be a useful program, yet it serves a malicious goal. They do not duplicate themselves; instead, they are downloaded onto a computer through internet contact [25]. The Trojan horse is installed on the victim's computer and, once installed, it can remotely control the victim's computer, steal confidential information, monitor user activity, and delete, edit, or corrupt files on the system where it is installed [32]. Trojans, unlike computer viruses and worms, must interact with users in order to propagate. Because Trojans are usually found after they have infected a computer system, they are one of the most destructive and dangerous types of malwares [23], [32].

The Trojan horse is divided into two main categories [29], which are: (i) Remote-Access Trojan and (ii) Trojan General.

1. Remote-Access Trojans: The Trojan horse of this sort is the most hazardous type of Trojan horse. They feature a unique capability that allows the attacker to control the victim PC remotely across a LAN or the Internet. An attacker can use this type of Trojan to carry out malicious operations such as stealing confidential information from the victim machine.
2. General Trojans: Trojans of this class engage in a wide range of malicious actions. They can jeopardize the integrity of victim machines' data. They can use system files that include URLs to reroute victims' workstations to a certain web site. They have the ability to install a variety of harmful software on victims' systems. They can even monitor user activity, save the data, and transfer it to the attacker.

3 Related works

This section summarizes some related research on malware detection using Machine Learning methods. There has been a lot of progress in the field of cloud malware detection. Recently, there has been a surge in interest in machine learning-based techniques to developing malware detection models on the cloud [33] – [37].

According to [30] a Convolutional Neural Network-based malware detection solution for cloud platforms (CNN) had been proposed. For malware detection, they used both a 2D CNN model and a 3D CNN model. They experimented with the data they acquired by running various malware on virtual PCs. The accuracy of the 2D CNN model is 79 percent, whereas the accuracy of the 3D CNN model is 90 percent. However, this study just looks at CNN and does not provide a comparison to classic machine learning algorithms [33], which is what we want to do in this research.

According to [38], They explored a machine learning-based malware detection framework and demonstrated the usefulness of a variety of machine learning models, including Decision Trees, Support Vector Machines, and others. They used the Cuckoo sandbox to evaluate malware samples in a simulation environment. The Cuckoo sandbox runs malware samples in a virtual environment and generates an analysis report based on their behavior. Many researchers [38], [39] have used the Cuckoo sandbox for malware investigation in the past.

The usefulness of utilizing CNN, RF, and KNN models for malware detection and relying on features retrieved from API calls was investigated by researchers in [40] – [42]. Additionally, [43] utilizes the random forest classifier to monitor the process activity of a virtual machine.

The effectiveness of SVM and Gaussian-based techniques using cloud performance indicators. and their study focused on cloud anomaly detection in general [44]. For detection purposes, [45] suggested a unique k-means clustering approach. This method proved successful in detecting highly active malware, but not in detecting malware with low activity.

The authors in [46] had suggested a honeypot and machine learning-based architecture for identifying malware. This research had used Support Vector Machines (SVM) and Decision Tree algorithms. In terms of accuracy, decision Tree and SVM algorithms have been demonstrated to be superior.

To detect and categorize malwares, [47] suggested a novel malware analysis methodology. In Weka, they deployed a variety of machine-learning models. They discovered that the J48 Decision Tree has good detection and classification accuracy. They used 220 samples for the experiment, which could be skewed because not all of the attributes were included in that number of samples.

3.1 Machine learning models

In this section, we'll go over the various machine learning models we employed in our research.

Random Forest Classifier (RFC). Random Forest algorithm had been created [48]; it is a supervised machine learning technique. An ensemble of classification trees is used in this technique [49]. The ensemble learning method creates a large number of learners who are then combined into a single result set. Random Forest is based on a variation of the Bagging method [50]. Each classifier in Bagging is constructed individually using a bootstrap sample of the input data. A decision is made at a node split in a normal decision tree classifier based on all feature properties. The optimum parameter at each node of a decision tree in Random Forest, on the other hand, is made up of a randomly selected number of characteristics [43]. This random feature selection aids Random Forest models in not only scaling effectively when there are numerous features per feature vector, but also in reducing feature attribute correlation. As a result, this approach is less susceptible to data noise. Furthermore, this classifier deals with missing values in the data [51].

Naive Bayes. It is a classification technique based on Bayes Theorem. The Bayes Theorem is used to generate classifications via the Naive Bayes classifier. The Bayes Theorem is a method for calculating conditional probability based on a set of attributes, but it takes a lot of computing power. The Bayes Theorem presupposes that all features are interdependent, which is why the theorem is so computationally intensive. To address this, a simpler or Naive technique was developed based on the premise that each of the features is independent; this assumption allows the theorem to be simplified, lowering the processing resources required.

Nearest Neighbor (IBK). It's also known as kNearest Neighbor (KNN), and it's a supervised learning classification method that works by evaluating the distances between samples that are close together. KNN employs the concept that samples with the same classification will be closer in distance to categorize a new sample based on the k closest neighbors.

Support Vector Classifier (SVC). Support Vector Classifiers are classification models that use supervised learning. The ability of SVC to use a non-linear kernel allows it to do non-linear classifications efficiently. This also cuts down on the amount of computing resources needed to calculate relationships in infinite dimensions. Because there isn't always a clear linear categorization between features, SVC uses higher dimensional relationships to make classifications that previous approaches, such as logistic regression, couldn't be able to make.

Bayesian Networks. Bayesian Networks (also known as Bayesian Belief Networks) is a probabilistic directed acyclic graphical model that shows conditional dependencies using directed acyclic graph Network can be used to detect "update knowledge of the state of a subset of variables when other variables (the evidence variables) are observed." In many applications of classification and information retrieval, Bayesian Networks are used [52].

Regression. Regression is a supervised learning method. It can be used to make predictions and model continuous variables. The following are some examples of applications of the linear regression algorithm: real-estate price prediction, sales forecasting, student test score forecasting, and stock exchange price forecasting. We have labeled datasets in regression, and the output variable value is dictated by the input variable values, making it a supervised learning strategy.

Decision Tree. It is a supervised machine learning method for solving classification and regression issues that involves continuously splitting data based on a parameter. The leaves make the decisions, while the nodes partition the data. The decision variable in a classification tree is categorical (the outcome is in the form of Yes/No), whereas the decision variable in a regression tree is continuous. The following are some of the benefits of using a decision tree: It is suitable for both regression and classification problems; it is simple to interpret; it is simple to handle categorical and quantitative values; it can fill missing values in attributes with the most likely value; and it has high performance due to the efficiency of the tree traversal algorithm. The downsides of a decision tree, on the other hand, are that it can be unstable, that it can be difficult to control the size of the tree, that it can be prone to sampling error, and that it provides a locally optimal answer rather than a globally optimal solution [53].

Multilayer perceptron. The Multilayer Perceptron is a nonparametric estimator that may be used to categorize and identify malware and intrusions. It is an artificial neural network structure (Nuanmeesri & Poomhiran, 2022). Using the back-propagation training process, the multilayer perceptron is the most often used model in neural network applications. The definition of architecture in MLP networks is crucial, because a shortage of connections can prevent the network from solving the problem of insufficient customizable parameters, whilst an excess of connections can lead to over-fitting of the training data. Especially when a large number of layers and neurons are used [53], [54].

J48. The J48 algorithm is a C4.5 decision tree learner implementation. Decision tree models are generated as a result of this implementation. It splits a dataset recursively according to attribute value tests in order to distinguish the possible predictions. To generate decision trees for classification, the algorithm employs the greedy technique. The training data is used to build a decision-tree model, which is then used to categorize the trained data. J48 is a program that creates decision trees. The J48 decision tree's node assesses the existence and importance of each unique attribute [55].

Table 1 summarizes the features, scope, malware analysis approach, and the classifiers used in various earlier and related research works. In addition, the last rows in the table summarizes the features, scope, malware analysis approach, and the classifiers used in this research.

Table 1. Distinctions between our work and others

	Features					**Scope**		**Malware Analysis Approach**			**Classifier**										
Paper	***System Calls***	***API Calls***	***Performance Metrics***	***Memory Features***	***Performance Counters***	***Host Environment***	***Cloud Comp. Envir.***	***Online Mal. Detect.***	***Anomaly Detection***	***Dynamic Ma. Detect.***	***KNN***	***Multilayer Perceptron***	***Random Forest***	***J48***	***Naïve Bayes***	***Regression***	***Neural Network***	***Decision Trees***	***SVC***	***Bayesian Networks***	***Clustering***
[56]	✓					✓				✓	✓				✓			✓	✓		
[57]	✓						✓	✓	✓		✓										✓
[58]			✓				✓	✓	✓						✓			✓			
[59]			✓				✓		✓										✓		
[60]					✓	✓		✓			✓		✓					✓			
[41]			✓			✓				✓			✓								
[37]			✓				✓	✓	✓										✓		
[61]	✓					✓				✓							✓				
[42]		✓				✓				✓	✓										
[40]		✓				✓				✓							✓				
[45]			✓				✓	✓	✓												✓
[62]				✓		✓		✓					✓								
[36]			✓				✓	✓									✓				
[63]	✓						✓	✓	✓								✓				
[43]			✓			✓				✓			✓								
[33]			✓				✓	✓			✓		✓		✓		✓		✓		
Own			✓				✓			✓	✓	✓	✓	✓	✓	✓			✓	✓	

As the table depicts, earlier works had concentrated on detecting general malware in the cloud computing environment. Whereas, this research has focused on specifically detecting Trojan horses in the cloud computing environment. Furthermore, this research has investigated wider range of machine learning classifiers such as Multilayer Perceptron, J48, Regression, and Bayesian Networks that have not been used previously in the earlier studies.

4 Malware analysis techniques

Malware analysis is the first step in detecting malware. To detect malware, first, malware behavior must be examined to determine how malware performs its function.

Hence, malware detector developers can easily incorporate defensive features. Based on time and technology used to perform the analysis, malware analysis approaches are categorized into two groups static and dynamic. The following subsections discuss these approaches in detail.

4.1 Static analysis

Static Analysis analyzes executable malware file in a controlled environment without executing it [25], [65], [66]. To assess whether or not the software contains harmful code, static information is collected from the code [25]. Many static features of the executable file are existed, such as memory compactness. Decompiler, dissembler, source code analyzers, and debugger are some of the tools that can be used to perform static analysis [25].

Static analysis can only reliably detect known malware signatures. As a result, it may occasionally fail to analyze unknown malware signatures that are not in its database. Static analysis is usually based on signatures, which must be updated on a regular basis and requires human expertise to produce new signatures [2], [7]. Due to the above reasons, it is not used in this research.

4.2 Dynamic analysis

The behavior of malware is examined in a dynamically controlled environment during dynamic analysis. When the malware runs, it modifies the registry and switches the operating system to privilege mode. Once the malware switches to privilege mode, it will gain the ability to control everything in the operating system.

Dynamic analysis software has a complete control over all resources. Hence, it can run in a safe environment. The software can update computer registry keys and it can execute in debugger mode in a controlled environment. After executing and analyzing a malware sample, the dynamic environment reverts to its prior snapshot which was created at the beginning of environment formation. This ensures that the environment is un-infected before analyzing another malware sample.

In this research, Sandbox, Portmon, Process Explorer, and other tools are used for dynamic analysis. Compared to static analysis, dynamic analysis is more effective [25], and it has significant capability and high reliability [67]. Due to the above reasons, dynamic analysis is adopted in this research.

5 Research methodology

This section contains a full description of the study's planned methodology, including explanation of the controlled lab setup, dataset collecting, analysis, and feature extraction.

5.1 Dataset collection

VirusShare.com [2], [7], [38], [68] and VirusTotal [38], [69] have been used to obtain cloud trojan horse and benign samples. Virusshare is one of the internet's largest openly distributed virus repositories. The VirusTotal site, on the other hand, is a malware samples repository that provides forensic investigators, incident responders, and security researchers with curious access to dangerous code samples [38].

1160 executable Trojan horse and benign samples have been collected from samples available on virushare.com and VirusTotal. These samples were recognized by various large anti-viruses such as Kaspersky, Bit Defender, Avira, Avast, AVG, Comodo, F-Secure, and others. This dataset will then be used to do further malware analysis using machine learning algorithms.

5.2 Controlled lab set up

For Trojan horse analysis, a controlled lab environment is used. Figure 1 illustrates the controlled lab setup for trojan horse malware analysis employed in this study's experiments. To prevent Trojan horse propagation, the physical network connection must be disconnected in order to create a fully regulated isolated environment. However, dynamic analysis of Trojan horse samples is not possible without a network connection. Hence, virtualization technology is employed in this setting to create a virtual cloud environment, and another server (the attacker and monitoring host) is connected to the cloud through Vmnet [2]. Similarly, this controlled lab architecture had been used in [2], [32], and [38].

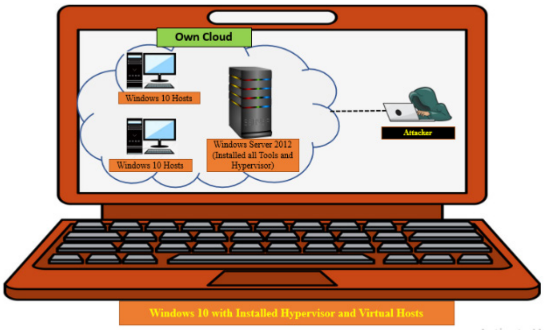

Fig. 1. Controlled laboratory setup

Figure 1 shows a controlled lab architecture that is completely disconnected from the internet. The cloud is deployed in a virtual environment on a physical system. The attacker starts the event in the cloud computing environment from outside of it. Within the cloud environment, all monitoring tools are installed on the server. These technologies are used to keep track of cloud behavior within the cloud. File monitoring, registry monitoring, process monitoring, and network monitoring are some of the techniques available. The cloud Trojan horse identification was tested in a controlled lab environment using dynamic analysis [32], [39] and the Cukoo Sandbox [38] was used for behavior monitoring. The Weka data mining tool was also utilized to determine the detection accuracy rate.

5.3 Analysis phase

In this stage, all files have been assessed using dynamic and Cuckoo Sandbox analysis. In this stage, each file is automatically executed. Then, all of its run-time behaviors are analyzed. After that, thorough analysis data that characterizes the malware's behaviors when it runs inside a newly installed operating system is collected.

For dynamic analysis, each trojan horse sample's test is tried in a controlled environment mode. After testing each Trojan horse sample, the controlled environment is re-established to the uninfected state by utilizing DeepFreeze computer program for analyzing the other samples. Furthermore, various dynamic analysis tools [2], listed in Table 2, have been used to test each sample's features which were installed in the cloud environment.

In this dynamic analysis phase, Trojan samples have been initially injected to the host and the behavior has been observed. Files, tcp, network, registry, all listening ports, processes, dlls and RAM, have been monitored using the aforementioned monitoring tools. If any of these tools' triggers unexpected behavior, the sample is flagged as malicious trojan and its features are thoroughly examined. If the sample is not found to be malicious, the analysis process for this sample is terminated, and the sample is flagged as benign [2], [32].

Table 2. Dynamic analysis tools

Tools	Item2
Process Explorer	To conduct the dynamic analysis
Newt pro	To conduct the dynamic analysis
Promiscdetect.exe	To conduct the dynamic analysis
Process monitoring	To conduct the dynamic analysis
PortMon	To conduct the dynamic analysis
Wireshark	To monitor the network traffic generated from the infected computer

Cuckoo sandbox, on the other hand, is the most extensively used open-source malware analysis system. For each submitted trojan horse sample, Cuckoo sandbox processes the sample file on a clean state virtual computer. It keeps track of all the execution traces that occur in the virtual environment and creates a complete report for each sample. This report describes how the file behaves when run in a realistic yet isolated context. A web interface or API calls are used to retrieve the analysis report.

5.4 Feature selection module

Feature selection is the process of determining the significance of existing features in a dataset. It retains the significant features and discards those that are not relevant [70]. Feature selection helps in getting a better accuracy rate.

For this study, 1160 trojan horse samples have been gathered from Virusshare and Virustotal. Dynamic analysis and the Cuckoo Sandbox analysis report have been used to perform the feature selection process. In this work, feature selection has been accomplished by a process that identified characteristics of each trojan horse through analysis performed in a controlled lab environment and tabulated into a relevant dataset for subsequent investigation.

After completing this phase, a comprehensive output analysis report is created. This report includes information on file and registry key creation, modification, deletion, access, and networking protocols. The output file comprises 25 features and has been converted into arff format (a compatible file format). This file format encodes the output data to be compatible as an input data for the WEKA machine learning simulation environment. J48, IBK, Nave Bayes, Random Forest, Regression and other classifiers and a 10-fold cross-validation have been used in this research to achieve a successful classification of the trojan horse dataset.

6 Experimental results and discussion

Experiments have been carried out with the entire dataset. Weka software is utilized in this experiment and a 10-fold cross-validation method has been adopted in this research. Weka software is utilized to apply the machine learning methods in this experiment. Weka is a Java-based open-source application. It has a set of machine learning methods for dealing with data mining issues [71]. Many scholars, including [2], [23], [38] have employed Weka in their work for clustering, classification, and detection.

The experiment was carried out with the use of a 10-fold cross-validation dataset created with Weka. The 10-fold cross-validation method is the professional standard for determining a learning scheme's error rate on a given dataset. Ten times ten-fold cross-validation was performed for reliable results [72], [75]. Furthermore, the 10-fold cross-validation separates the dataset into ten parts (folds). As a result, for each dataset parts, nine times are used for training and one is used for testing.

Weka uses 10-fold cross-validation by default because extensive studies on a variety of datasets using various learning algorithms have demonstrated that 10 folds is about the right number of folds for getting the best estimate of error [73]. There are two primary causes for using a 10-fold cross-validation test of this type. First, 10-fold cross-validation makes use of as much data as feasible during the training and testing process. Second, its findings are more accurate [74].

Different standard performance indicators, such as True Positive (TP) Rate, False Positive (FP) Rate, Accuracy, Precision, Recall, Kappa Statistics, F-Measure, and Receiver Operating Characteristic (ROC) Area, have been utilized to evaluate the performance of the classifiers. Table 3 shows the experiment's results of this research.

Table 3. Trojan horse detection results in cloud computing environment using various machine learning methods

Algorithm	TP Rate	FP Rate	Precision	Recall	F-Meas-ure	ROC Area	Kappa Statistic	Accuracy (%)
Naïve Bayes	0.957	0.291	0.957	0.957	0.953	0.97	0.768	95.6897
Random Forest	0.957	0.291	0.957	0.957	0.953	0.966	0.768	95.6897
IBK	0.957	0.291	0.957	0.957	0.953	0.966	0.768	95.6897
Regression	0.957	0.291	0.957	0.957	0.953	0.958	0.768	95.6897
J48	0.957	0.291	0.957	0.957	0.953	0.96	0.768	95.6897
Multilayer Perceptron	0.959	0.278	0.959	0.959	0.955	0.971	0.779	95.8621
Bayesian Networks	0.957	0.291	0.957	0.957	0.953	0.97	0.768	95.6897
SMO	0.959	0.278	0.959	0.959	0.955	0.84	0.779	95.8621

6.1 True positive rate results

The number of cloud's trojan horse samples that are accurately classified and labelled as harmful is known as true positive (TP). Equation (1) is used to compute TP.

$$True\ positive\ rate\ (TPR) = \frac{TP}{(TP+FN)} \tag{1}$$

From the experimental results of the classifiers in this study, Multilayer Perceptron and Support Vector classifier (SMO) achieves the highest true positive of 0.959 followed by Naïve Bayes, Random Forest, IBK, Regression, J48 and Bayesian Networks classifiers of 0.957. Figure 2 and Table 3 shows the TP Rate of each classifier.

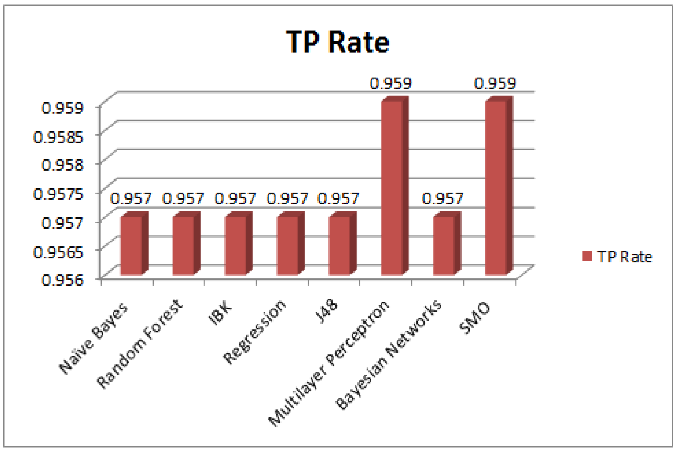

Fig. 2. TP rate of several classification algorithms

6.2 False Positive rate results

The False Positive (FP) indicates that the data has been misclassified, implying that it belongs to a different class. The performance evaluation considers the FP in addition to TP. In a nutshell, FP is the number of cloud's trojan horse samples that are wrongly labeled as harmful. Equation (2) is used to calculate FP.

$$False\ positive\ rate\ (FPR) = \frac{FP}{(FP+TN)} \quad (2)$$

In the case of FP, Multilayer Perceptron and SMO shows the lowest FP rate with 0.278 compared to Naïve Bayes, Random Forest, IBK, Regression, J48 and Bayesian Networks classifiers of 0.291. Figure 3 and Table 3 shows the FP Rate of each classifier.

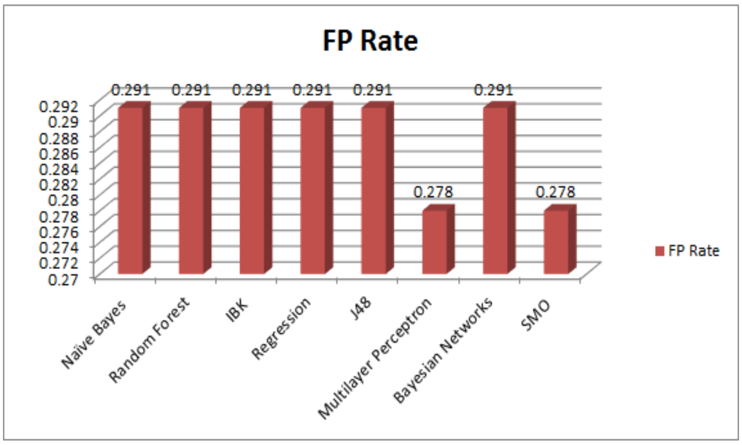

Fig. 3. FP rate of various classification algorithms

6.3 Precision rate results

The proportion of true positive classifications in all positive findings is explained by precision. It refers to the number of samples that are accurately identified and are not false positives. The TP and FP rates are used to calculate precision, as stated in Equation (3). A higher TP indicates greater precision:

$$Precision = \frac{TP}{(TP+FP)} \quad (3)$$

Table 3 and Figure 4 show the precision results for the various classifiers employed in this investigation. Figure 4 shows that the Multilayer Perceptron and SMO have a satisfactory precision value of 0.959. As the figure depicts, Multilayer Perceptron and SMO have a greater precision rate than other classifiers.

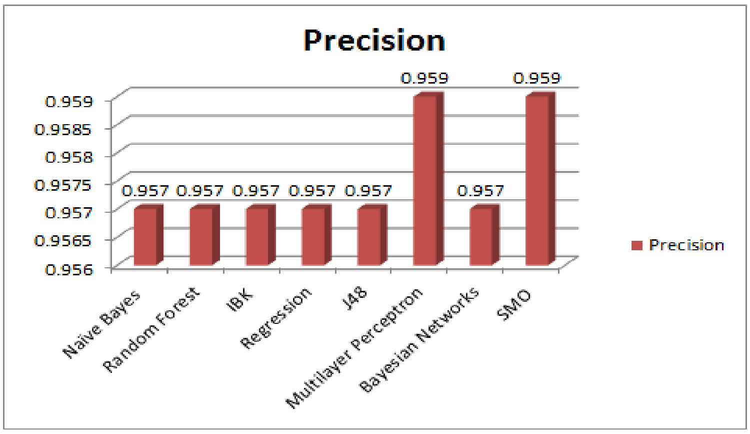

Fig. 4. Precision rate of various classification algorithms

6.4 Recall rate results

The number of projected cloud's trojan horse samples that are accurately classified and labelled as harmful is known as true positive (TP). Equation (1) is used to compute TP.

$$Recall = TPR = \frac{TP}{(TP+FN)} \tag{4}$$

From Table 3 and Figure 5, it is clearly seen that the Multilayer Perceptron and SMO have the highest recall values, which are 0.959%. While all other classifiers recall values are equal and their recall value is equal to 0.957%.

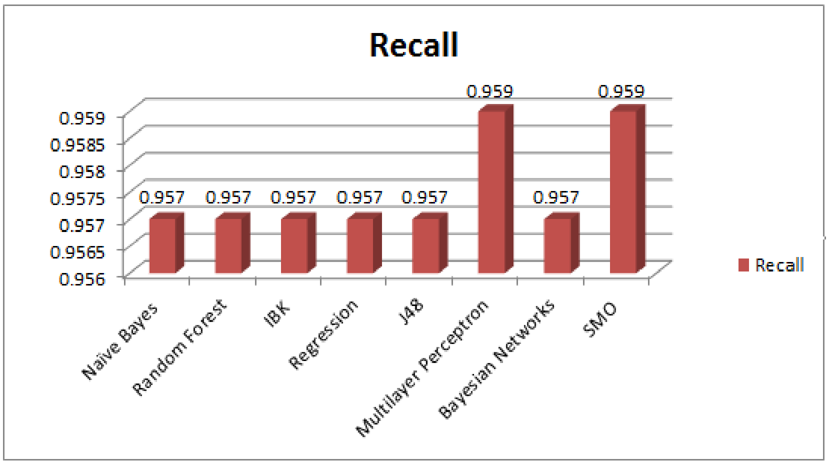

Fig. 5. Recall rate of various classification algorithms

6.5 F-Measure rate results

F-Measure is the value that combines both precision and recall into a single value to measure the system's overall performance. Equation (5) depicts the way F-measure is calculated.

$$F - Measure = \frac{2\ x\ Recall\ x\ Precision}{Recall + Precision} \tag{5}$$

The experimental results in Table 3 and Figure 6 show that SMO and Multilayer Perceptron have the highest F-measure of 0.955.

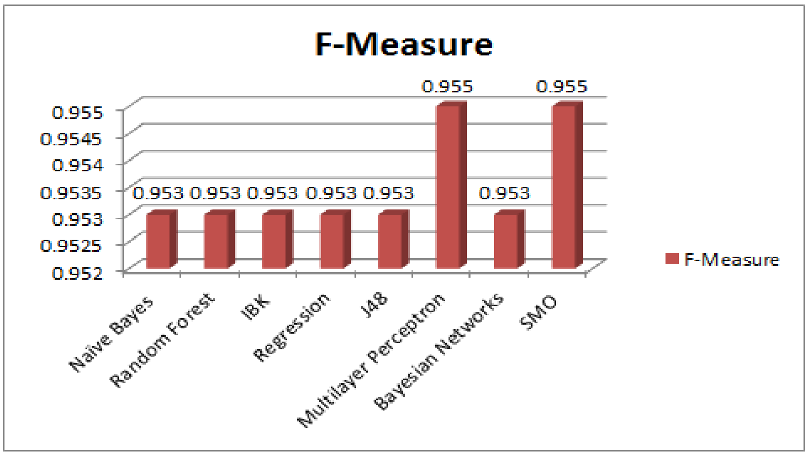

Fig. 6. F-Measure rate of various classification algorithms

6.6 Roc Area rate results

The probability of a classifier ranking a randomly chosen positive instance higher than a randomly chosen negative instance is known as the ROC Area. A perfect prediction is represented by ROC value of 1.0. The experimental results for the various classifiers in this study have shown that Multilayer Perceptron has the best performance in term of ROC Area rate of 0.971. On the other hand, SMO has the worst performance ROC of 0.84. The ROC Area Rate for each classifier is shown in Table 3 and Figure 7.

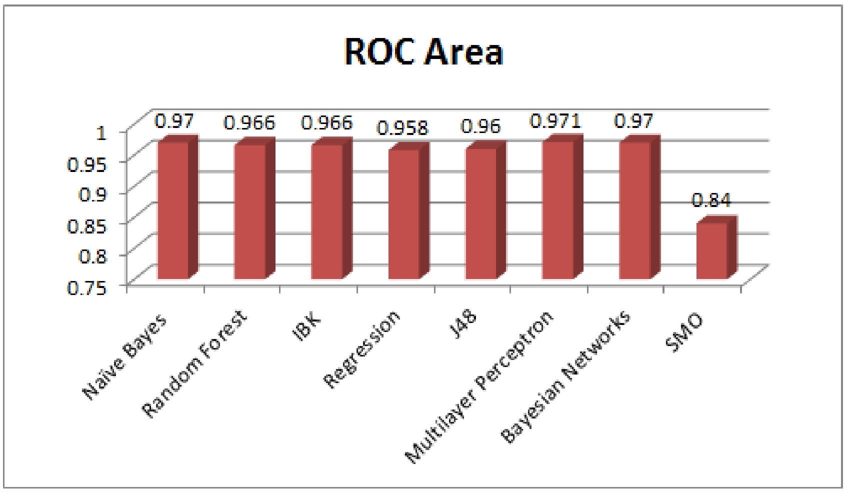

Fig. 7. ROC Area rate of various classification algorithms

6.7 Kappa statistic rate results

The Kappa statistic is a performance indicator that compares observed and expected accuracy (random chance). It expresses the level of agreement between real classes and categories. The maximum kappa statistic value is 0.78, which indicates full agreement.

The experimental results in Table 3 and Figure 8 show that Multilayer Perceptron and SMO have the highest Kappa statistic performance value of 0.779. Other classifiers Kappa statistic performance values are equal and their Kappa value is equal to 0.768.

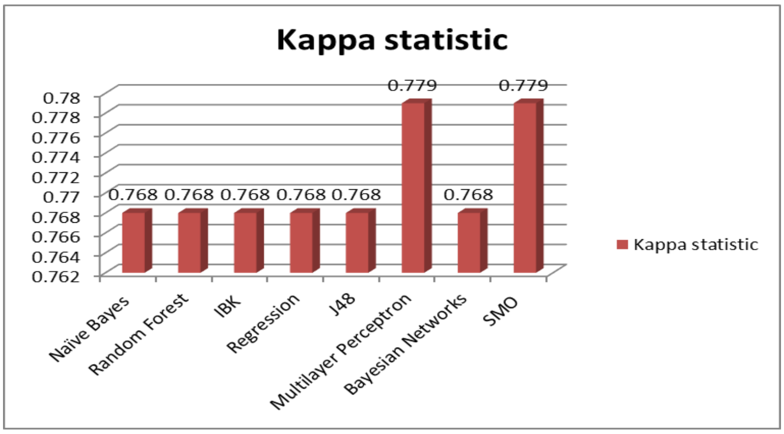

Fig. 8. Kappa statistic rate results of various classification algorithms

6.8 Accuracy rate results

Correctly classification is another term for accuracy. Accuracy is a performance statistic for expressing the percentage of right predictions. The accuracy rates for various classifiers are illustrated in Table 3 and Figure 9.

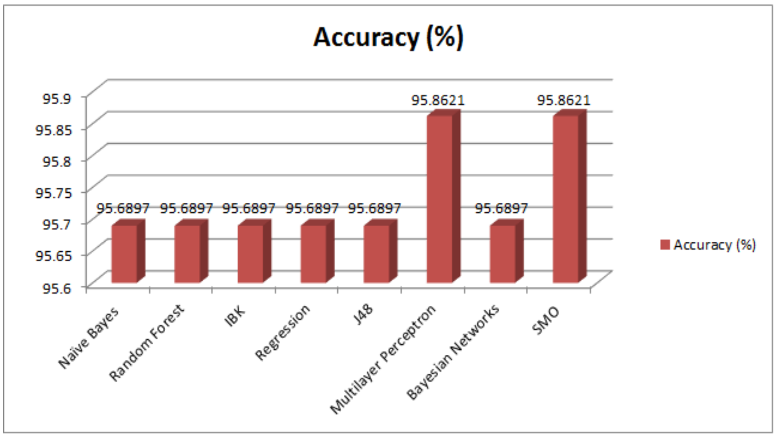

Fig. 9. Accuracy rate of various classification algorithms

The experimental results for the various classifiers in this study have shown that the Multilayer Perceptron and SMO have the highest accuracy rate value of 95.86. In contrast, all other classifiers accuracy rates are equal and their accuracy rates is equal to 95.68. As a result, it can be inferred that, in terms of correctly classified rates, the Multilayer Perceptron and SMO classifiers outperformed other classifiers.

Based on the findings in Table 3 and Figures 2–9, it can be inferred that Multilayer Perceptron and SMO classifiers have outperformed Nave Bayes, Random Forest, IBK, Regression, J48, and Bayesian Networks classifiers for trojan horse detection in a cloud-based environment.

7 Conclusion

This research has compared eight machine learning classifiers for Trojan horse detection in a cloud-based environment. These classifiers include Multilayer Perceptron, SMO, Nave Bayes, Random Forest, IBK, Regression, J48, and Bayesian Networks. The accuracy of the cloud trojan horse detection rate has been investigated using dynamic analysis, Cukoo sandbox, and the Weka data mining tool. Initially, the controlled lab has been set up. Then, trojan horses dataset has been collected. After that, cloud trojan horse dataset has been analyzed in the established controlled lab environment. Then, significant features are extracted. Finally, Weka software is utilized to apply the various machine learning classifiers on cloud trojan horses dataset and compare their detection accuracy rate. 10-fold cross-validation method has been adopted during the experiments. Based on the conducted experiments, the SMO and Multilayer Perceptron have been found to be the best classifiers for trojan horse detection in a cloud-based environment. Although SMO and Multilayer Perceptron have achieved similar results, Multilayer Perceptron has outperformed SMO in term of ROC area. Further research is needed to improve the accuracy rate of Trojan horse detection in the cloud computing environment. The findings of this research will be extremely useful to other researchers to develop an effective and efficient technique for Trojan horse detection in cloud computing environment.

8 Acknowledgment

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

9 References

[1] Bayramusta, M., & Nasir, V. A. (2016). A fad or future of IT?: A comprehensive literature review on the cloud computing research. International Journal of Information Management, 36(4), 635-644. https://doi.org/10.1016/j.ijinfomgt.2016.04.006

[2] Kanaker, H. M., Saudi, M. M., & Marhusin, M. F. (2014, August). Detecting worm attacks in cloud computing environment: Proof of concept. In 2014 IEEE 5th Control and System Graduate Research Colloquium (pp. 253-256). IEEE. https://doi.org/10.1109/ICSGRC.2014.6908732

[3] Avram, M. G. (2014). Advantages and challenges of adopting cloud computing from an enterprise perspective. *Procedia Technology*, *12*, 529-534. https://doi.org/10.1016/j.protcy.2013.12.525

[4] Sun, Y., Zhang, J., Xiong, Y., & Zhu, G. (2014). Data security and privacy in cloud computing. *International Journal of Distributed Sensor Networks*, *10*(7), 190903. https://doi.org/10.1155/2014/190903

[5] Almorsy, M., Grundy, J., & Müller, I. (2016). An analysis of the cloud computing security problem. *arXiv preprint arXiv:1609.01107*.

[6] Jones, S., Irani, Z., Sivarajah, U., & Love, P. E. (2017). Risks and rewards of cloud computing in the UK public sector: A reflection on three Organizational case studies. *Information Systems Frontiers*, 1-24. https://doi.org/10.1007/s10796-017-9756-0

[7] Kanaker, H., Saudi, M. M., & Azman, N. (2017). Evaluation of EWCDMCC Cloud Worm Detection Classification Based on Statistical Analysis. *Advanced Science Letters*, *23*(6), 5365-5369. https://doi.org/10.1166/asl.2017.7377

[8] Kanaker, H. M., Saudi, M. M., & Marhusin, M. F. (2015). A systematic analysis on worm detection in cloud-based systems. *ARPN Journal of Engineering and Applied Sciences*.

[9] Kundra, V. (2011). Federal cloud computing strategy.

[10] Kepes, B. (2015). A Cautionary Government Cloud Story–UK's GCloud. Does The "G" Stand For Gone?. [online] Forbes. Available at: http://www.forbes.com/sites/benkepes/2015/01/27a-cautionarygovernment-cloud-story-uks-g-cloud-does-the-g-stand-for-gone/ [Accessed 1 Jul. 2015].

[11] Bojanova, I., Zhang, J., & Voas, J. (2013). Cloud computing. It Professional, 15(2), 12-14. https://doi.org/10.1109/MITP.2013.26

[12] Bhat, A. H., Patra, S., & Jena, D. (2013). Machine learning approach for intrusion detection on cloudvirtual machines. International Journal of Application or Innovation in Engineering & Management (IJAIEM), 2(6), 56-66.
[13] Carlin, S., & Curran, K. (2013). Cloud computing security. In Pervasive and Ubiquitous Technology Innovations for Ambient Intelligence Environments (pp. 12-17). IGI Global. https://doi.org/10.4018/978-1-4666-2041-4.ch002
[14] Subashini, S., & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. Journal of network and computer applications, 34(1), 1-11. https://doi.org/10.1016/j.jnca.2010.07.006
[15] Karim, N.A., Kanaker, H., Almasadeh, S., & Zarqou, J. (2021). A Robust User Authentication Technique in Online Examination. https://doi.org/10.47839/ijc.20.4.2441
[16] Bhasin, S., & Regazzoni, F. (2015, May). A survey on hardware trojan detection techniques. In 2015 IEEE International Symposium on Circuits and Systems (ISCAS)(pp. 2021-2024). IEEE. https://doi.org/10.1109/ISCAS.2015.7169073
[17] Hashizume, K., Yoshioka, N., & Fernandez, E. B. (2013). Three misuse patterns for cloud computing. In Security engineering for Cloud Computing: approaches and Tools (pp. 36-53). IGI Global. https://doi.org/10.4018/978-1-4666-2125-1.ch003
[18] Chou, T. S. (2013). Security threats on cloud computing vulnerabilities. International Journal of Computer Science & Information Technology, 5(3), 79. https://doi.org/10.5121/ijcsit.2013.5306
[19] Jamil, D., & Zaki, H. (2011). Security issues in cloud computing and countermeasures. International Journal of Engineering Science and Technology (IJEST), 3(4), 2672-2676.
[20] Modi, C., Patel, D., Borisaniya, B., Patel, A., & Rajarajan, M. (2013). A survey on security issues and solutions at different layers of Cloud computing. The journal of supercomputing, 63(2), 561-592. https://doi.org/10.1007/s11227-012-0831-5
[21] Liu, Y. F., Zhang, L. W., Liang, J., Qu, S., & Ni, Z. Q. (2010, July). Detecting Trojan horses based on system behavior using machine learning method. In 2010 International Conference on Machine Learning and Cybernetics (Vol. 2, pp. 855-860). IEEE. https://doi.org/10.1109/ICMLC.2010.5580591
[22] John Love. “Malware Types and Classification.” Lastline.com. March 28, 2018. Url: https://www.lastline.com/blog/malware-types-and-classification
[23] Dada, E. G., Bassi, J. S., Hurcha, Y. J., & Alkali, A. H. (2019). Performance evaluation of machine learning algorithms for detection and prevention of malware attacks. IOSR Journal of Computer Engineering, 21(3), 18-27.
[24] Zimba, A., Wang, Z., & Chen, H. Multi-stage crypto ransomware attacks: A new emerging cyber threat to critical infrastructure and industrial control systems. ICT Express, 2018. https://doi.org/10.1016/j.icte.2017.12.007
[25] Tahir, R. (2018). A study on malware and malware detection techniques. International Journal of Education and Management Engineering, 8(2), 20. https://doi.org/10.5815/ijeme.2018.02.03
[26] Jamil, Q., & Shah, M. A. (2016, August). Analysis of machine learning solutions to detect malware in android. In 2016 Sixth International Conference on Innovative Computing Technology (INTECH) (pp. 226-232). IEEE. https://doi.org/10.1109/INTECH.2016.7845073
[27] Saradha R. Malware Analysis using Profile Hidden Markov Models and Intrusion Detection in a Stream Learning Setting. Master’sThesis. Fac. of Engineering. Indian Institute of Science, 2014.
[28] Cisco (2015). What is the difference: Viruses, worms, trojans, and bots? Online. http://www.cisco.com/web/about/security/intelligence/virus-worm-diffs.html

[29] Zeidanloo, H. R., Tabatabaei, F., Amoli, P. V., & Tajpour, A. (2010, July). All About Malwares (Malicious Codes). In Security and Management (pp. 342-348).
[30] Fui, N. L. Y., Asmawi, A., & Hussin, M. (2020). A Dynamic Malware Detection in Cloud Platform. International Journal of Difference Equations (IJDE), 15(2), 243-258. https://doi.org/10.37622/IJDE/15.2.2020.243-258
[31] Damodaran A., Troia F. D., Corrado V. A., Austin T. H. and Stamp M. A Comparison of Static, Dynamic, and Hybrid Analysis for Malware Detection, 2015. https://doi.org/10.1007/s11416-015-0261-z
[32] Abuzaid, A. M., Saudi, M. M., Taib, B. M., & Abdullah, Z. H. (2013). An efficient trojan horse classification (ETC). International Journal of Computer Science Issues (IJCSI), 10(2), 96.
[33] Kimmell, J. C., Abdelsalam, M., & Gupta, M. (2021). Analyzing Machine Learning Approaches for Online Malware Detection in Cloud. arXiv preprint arXiv:2105.09268. https://doi.org/10.1109/SMARTCOMP52413.2021.00046
[34] Kumar, R., Sethi, K., Prajapati, N., Rout, R. R., & Bera, P. (2020, July). Machine Learning based Malware Detection in Cloud Environment using Clustering Approach. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-7). IEEE. https://doi.org/10.1109/ICCCNT49239.2020.9225627
[35] Abdelsalam, M., Krishnan, R., & Sandhu, R. (2019, July). Online malware detection in cloud auto-scaling systems using shallow convolutional neural networks. In IFIP Annual Conference on Data and Applications Security and Privacy (pp. 381-397). Springer, Cham. https://doi.org/10.1007/978-3-030-22479-0_20
[36] Abdelsalam, M., Krishnan, R., Huang, Y., & Sandhu, R. (2018, July). Malware detection in cloud infrastructures using convolutional neural networks. In 2018 IEEE 11th International conference on cloud computing (CLOUD) (pp. 162-169). IEEE. https://doi.org/10.1109/CLOUD.2018.00028
[37] Watson, M. R., Marnerides, A. K., Mauthe, A., & Hutchison, D. (2015). Malware detection in cloud computing infrastructures. IEEE Transactions on Dependable and Secure Computing, 13(2), 192-205. https://doi.org/10.1109/TDSC.2015.2457918
[38] Sethi, K., Kumar, R., Sethi, L., Bera, P., & Patra, P. K. (2019, June). A novel machine learning based malware detection and classification framework. In 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security) (pp. 1-4). IEEE. https://doi.org/10.1109/CyberSecPODS.2019.8885196
[39] Lin, C. T., Wang, N. J., Xiao, H., & Eckert, C. (2015). Feature Selection and Extraction for Malware Classification. J. Inf. Sci. Eng., 31(3), 965-992.
[40] Tobiyama, S., Yamaguchi, Y., Shimada, H., Ikuse, T., & Yagi, T. (2016, June). Malware detection with deep neural network using process behavior. In 2016 IEEE 40th annual computer software and applications conference (COMPSAC) (Vol. 2, pp. 577-582). IEEE. https://doi.org/10.1109/COMPSAC.2016.151
[41] Pirscoveanu, R. S., Hansen, S. S., Larsen, T. M., Stevanovic, M., Pedersen, J. M., & Czech, A. (2015, June). Analysis of malware behavior: Type classification using machine learning. In 2015 International conference on cyber situational awareness, data analytics and assessment (CyberSA) (pp. 1-7). IEEE. https://doi.org/10.1109/CyberSA.2015.7166115
[42] Fan, Y., Ye, Y., & Chen, L. (2016). Malicious sequential pattern mining for automatic malware detection. Expert Systems with Applications, 52, 16-25. https://doi.org/10.1016/j.eswa.2016.01.002
[43] Joshi, S., Upadhyay, H., Lagos, L., Akkipeddi, N. S., & Guerra, V. (2018, April). Machine learning approach for malware detection using random forest classifier on process list data

structure. In Proceedings of the 2nd International Conference on Information System and Data Mining (pp. 98-102). https://doi.org/10.1145/3206098.3206113
[44] Pannu, H. S., Liu, J., & Fu, S. (2012, October). Aad: Adaptive anomaly detection system for cloud computing infrastructures. In 2012 IEEE 31st Symposium on Reliable Distributed Systems (pp. 396-397). IEEE. https://doi.org/10.1109/SRDS.2012.3
[45] Abdelsalam, M., Krishnan, R., & Sandhu, R. (2017, June). Clustering-based IaaS cloud monitoring. In 2017 IEEE 10th International Conference on Cloud Computing (CLOUD) (pp. 672-679). IEEE. https://doi.org/10.1109/CLOUD.2017.90
[46] Matin, I. M. M., & Rahardjo, B. (2019, November). Malware detection using honeypot and machine learning. In 2019 7th International Conference on Cyber and IT Service Management (CITSM) (Vol. 7, pp. 1-4). IEEE. https://doi.org/10.1109/CITSM47753.2019.8965419
[47] Sethi, K., Chaudhary, S. K., Tripathy, B. K., & Bera, P. (2018, January). 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 (pp. 1-4). https://doi.org/10.1145/3154273.3154326
[48] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324
[49] Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J. (2017). Classification and regression trees. Routledge. https://doi.org/10.1201/9781315139470
[50] Breiman, L. (1996). Bagging predictors. Machine learning, 24(2), 123-140. https://doi.org/10.1007/BF00058655
[51] Pirscoveanu, R. S., Hansen, S. S., Larsen, T. M., Stevanovic, M., Pedersen, J. M., & Czech, A. (2015, June). Analysis of malware behavior: Type classification using machine learning. In 2015 International conference on cyber situational awareness, data analytics and assessment (CyberSA) (pp. 1-7). IEEE. https://doi.org/10.1109/CyberSA.2015.7166115
[52] Fan, Y., Ye, Y., & Chen, L. (2016). Malicious sequential pattern mining for automatic malware detection. Expert Systems with Applications, 52, 16-25. https://doi.org/10.1016/j.eswa.2016.01.002
[53] Joshi, S., Upadhyay, H., Lagos, L., Akkipeddi, N. S., & Guerra, V. (2018, April). Machine learning approach for malware detection using random forest classifier on process list data structure. In Proceedings of the 2nd International Conference on Information System and Data Mining (pp. 98-102). https://doi.org/10.1145/3206098.3206113
[54] Pannu, H. S., Liu, J., & Fu, S. (2012, October). Aad: Adaptive anomaly detection system for cloud computing infrastructures. In 2012 IEEE 31st Symposium on Reliable Distributed Systems (pp. 396-397). IEEE. https://doi.org/10.1109/SRDS.2012.3
[55] Abdelsalam, M., Krishnan, R., & Sandhu, R. (2017, June). Clustering-based IaaS cloud monitoring. In 2017 IEEE 10th International Conference on Cloud Computing (CLOUD) (pp. 672-679). IEEE. https://doi.org/10.1109/CLOUD.2017.90
[56] Matin, I. M. M., & Rahardjo, B. (2019, November). Malware detection using honeypot and machine learning. In 2019 7th International Conference on Cyber and IT Service Management (CITSM) (Vol. 7, pp. 1-4). IEEE. https://doi.org/10.1109/CITSM47753.2019.8965419
[57] Sethi, K., Chaudhary, S. K., Tripathy, B. K., & Bera, P. (2018, January). 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 (pp. 1-4). https://doi.org/10.1145/3154273.3154326
[58] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324
[59] Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J. (2017). Classification and regression trees. Routledge. https://doi.org/10.1201/9781315139470

[60] Breiman, L. (1996). Bagging predictors. Machine learning, 24(2), 123-140. https://doi.org/10.1007/BF00058655
[61] Luckett, P., McDonald, J. T., & Dawson, J. (2016, April). Neural network analysis of system call timing for rootkit detection. In 2016 Cybersecurity Symposium (CYBERSEC) (pp. 1-6). IEEE. https://doi.org/10.1109/CYBERSEC.2016.008
[62] Xu, Z., Ray, S., Subramanyan, P., & Malik, S. (2017, March). Malware detection using machine learning based analysis of virtual memory access patterns. In Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017 (pp. 169-174). IEEE. https://doi.org/10.23919/DATE.2017.7926977
[63] Dawson, J. A., McDonald, J. T., Hively, L., Andel,T. R., Yampolskiy, M., & Hubbard, C. (2018,April). Phase space detection of virtual machine cyber events through hypervisor-level system call analysis. In 2018 1st International Conference on Data Intelligence and Security (ICDIS) (pp. 159- 167). IEEE. https://doi.org/10.1109/ICDIS.2018.00034
[64] Nuanmeesri, S., & Poomhiran, L. (2022). Multi-Layer Perceptron Neural Network and Internet of Things for Improving the Realtime Aquatic Ecosystem Quality Monitoring and Analysis. International Journal of Interactive Mobile Technologies, 16(6). https://doi.org/10.3991/ijim.v16i06.28661
[65] Ijaz, M., Durad, M. H., & Ismail, M. (2019, January). Static and dynamic malware analysis using machine learning. In 2019 16th International bhurban conference on applied sciences and technology (IBCAST) (pp. 687-691). IEEE. https://doi.org/10.1109/IBCAST.2019.8667136
[66] Akintola, A. G., Balogun, A. O., Mojeed, H. A., Usman-Hamza, F., Salihu, S. A., Adewole, K. S., ... & Sadiku, P. O. (2022). Performance Analysis of Machine Learning Methods with Class Imbalance Problem in Android Malware Detection. International Journal of Interactive Mobile Technologies, 16, 140-162. https://doi.org/10.3991/ijim.v16i10.29687
[67] Damodaran, A. (2015). Combining dynamic and static analysis for malware detection. https://doi.org/10.1007/s11416-015-0261-z
[68] "VirusShare.com." [Online]. Available: https://virusshare.com/. [Accessed: 03-Jun-2022].
[69] "Virustotal.com." [Online]. Available: https://www.virustotal.com/. [Accessed:03-Jun-2022].
[70] Muhsen, Atheer R., Ghazwh G. Jumaa, Nadia F. AL Bakri, and Ahmed T. Sadiq. "Feature selection strategy for network intrusion detection system (NIDS) using meerkat clan algorithm." International Journal of Interactive Mobile Technologies 15, no. 16 (2021): 158–171. https://doi.org/10.3991/ijim.v15i16.24173
[71] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 11(1), 10-18. https://doi.org/10.1145/1656274.1656278
[72] Bouckaert, R. R., Frank, E., Hall, M., Kirkby, R. Reutemann, P., Seewald, A., & Scuse, D. (2016). WEKA manual for version 3-9-1. University of Waikato, Hamilton, New Zealand.
[73] Witten.I, Frank. E, Hall .M, and Pal . C, Data Mining: Practical machine learning tools and techniques Morgan Kaufmann. 2016.
[74] Saudi.M.M, "A new model for worm detection and response (Doctoral dissertation)," University of Bradford, 2012.
[75] AL-Behadili, H. N. K., & Ku-Mahamud, K. R. (2021). Fuzzy unordered rule using greedy hill climbing feature selection method: An application to diabetes classification. Journal of Information and Communication Technology, 20(3), 391-422. https://doi.org/10.32890/jict2021.20.3.5

10 Authors

Dr. Hasan Kanaker was awarded his B.Sc. in Computer Information System from Al-Zaytoonah University in 2005, master in Computer Science from Al-Balqa Applied University, Jordan in 2007 and Ph.D. in 2018 from Islamic Science University of Malaysia (USIM), Malaysia. Currently, he is an assistant professor and the Head of Cybersecurity and Computer Information System departments at Isra University, Jordan. He has very good experience in the field of cyber security, networking, malware detection and E-learning. Also, he had been engaged in multiple research works such as user authentication technique in online examination, analysis of medical images using neural networks and smart traffic control using internet of things and geographical information system. His research interest includes information Security, Malware and Malware Detection, Cloud Computing Security, Data Mining and Machine learning, Intrusion Detection and Network Security. Dr. Hasan Kanaker can be reached via his email address (hasan.kanaker@iu.edu.jo).

Dr. Nader Salameh was awarded his Ph.D. in 2017 from National University of Malaysia (UKM), Malaysia. His Ph.D. thesis investigated a new User authentication method based on user interface preferences for the account recovery process (UIPA). Currently, he is an assistant professor at faculty of Artificial Intelligence, Al-Balqa Applied University, Jordan. He has very good experience in the field of user authentication, cyber security, Human-Computer Interaction (HCI), and E-learning. Also, he has had been engaged in several research works such as Preferences based authentication, virtual privacy technique. Dr. Nader Salameh can be reached via his email address (nader.salameh@bau.edu.jo).

Dr. Samer A. B. Awwad received his B.Sc. in Engineering Technology with a major in Computer Engineering from Yarmouk University, Irbid, Jordan, in 2004. He worked for Technical and Vocational Training Corporation (Jeddah Military and Vocational Training Institute), Kingdom of Saudi Arabia, as a lecturer for 3 years. He received his M.Sc. in Communications and Network Engineering from University of Putra Malaysia (UPM), Serdang, Selangor, Malaysia, in 2010. He joined Nilai University as a lecturer from 2010 to 2013. He received his Ph.D. in Communications and Network Engineering from University of Putra Malaysia (UPM) in 2016. He was a lecturer in the Department of Computer Engineering and Computer Science, Manipal International University, Malaysia between October 2015 and March 2019. Currently, he is quality assurance manager at the deanship of information and communication technology at Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. His research interests include wireless ad hoc and sensor network specialized in mobility environments, information security, 6LoWPAN, IEEE 802.15.4, IoT, compression and routing. He can be reached via his email (saawad@iau.edu.sa).

Dr. Nurul Halimatul Asmak Ismail received the degree in Computer Science from University of Science Malaysia (USM), in 2000 and master degree from University of Putra Malaysia (UPM), in 2009. She was working with Majlis Amanah Rakyat, Malaysia as a lecturer for Higher National Diploma in Computing an Edexcel program from United Kingdom. She involved with research group, program accreditation and syllabus construction. Years of experiences with Edexcel program encouraging her to finish

her PhD in Computer Science specialization in Networking from National University of Malaysia (UKM), Bangi, Selangor, Malaysia in 2015. Currently she is working at Princess Nourah Bint Abdulrahman University, Kingdom of Saudi Arabia as assistant professor in department of Computer Science and Information Technology, College of Community. Her interests are in 6LoWPAN routing protocol, Internet of Things (IoT), machine learning and information security. Dr. Nurul Halimatul can be reached via her email (nhismail@pnu.edu.sa).

Dr. Jamal Zraqou was awarded his Ph.D. in 2011 from Bradford University, United Kingdom. His Ph.D. thesis investigated the development of new technologies for processing information contained in multiple and overlapping images of the same scene to produce images of improved quality. Currently, he is an associate professor at Computer Science at University of Petra. He has very good experience in the field of image processing such as super-resolution, objects detection and tracking, facial expression tracking and recognition, object character recognition, and 3D image reconstruction from un-calibrated stereo pair of images. Also, he had been engaged in multiple research works such as: building smart cities, tracking systems based on GPS service, and information security. He can be reached via his email address (jamal.zraqou@uop.edu.jo).

Dr.Abdulla Mousa Falah AlAli was awarded his Ph.D. in 2013 from Beijing University of Aeronautics and Astronautics ,China . His Ph.D. investigated Studies on Automatic Segmentation Strategies to Improve Bioluminescent Tomography Performance. Currently, he is an assistant professor at the Department of Computer Science at Isra University, Jordan. He has useful experience in medical image processing algorithms, networking, malware detection and E-learning. Also, he has been engaged in multiple research works such as “Diagnosing Chest X-Rays for Early Detection of COVID-19 And Distinguishing It from Other Pneumonia Using Deep Learning Networks". His research interest includes medical image processing algorithms, networking, malware detection and Education. Dr.Abdulla AlAli can be reached via his email (abdulla.alali@iu.edu.jo).

Article submitted 2022-10-01. Resubmitted 2022-11-13. Final acceptance 2022-11-13. Final version published as submitted by the authors.