International Journal of Safety and Security Engineering
Vol. 15, No. 2, February, 2025, pp. 197-207
Journal homepage: http://iieta.org/journals/ijsse

Comparative Analysis of Malware Detection Approaches in Cloud Computing

Doaa Abdelrahman[1,2], Mohamed Rasslan[1,3*], Nashwa Abdelbaki[3]

[1] EG-Cert, National Telecommunication Regulatory Authority, Giza 12577, Egypt
[2] Center of Informatics Science, Faculty of Information Technology and Computer Science, Nile University, Giza 12588, Egypt
[3] Electronics Research Institute, Ministry of Higher Education and Scientific Research, Cairo 12622, Egypt

Corresponding Author Email: mohamed@eri.sci.eg

https://doi.org/10.18280/ijsse.150201

Received: 6 November 2024
Revised: 25 January 2025
Accepted: 10 February 2025
Available online: 28 February 2025

Keywords:
malware analysis, cloud computing, malware attacks, malware injection attacks, malware detection approaches

ABSTRACT

The widespread use of cloud computing techniques in many applications renders cloud computing environments extremely vulnerable to malware infections and novel attacks. The flexibility, scalability, and elasticity cloud computing provide add to the difficulty of detecting malicious software in cloud computing environments. In this study, we analyze malware attacks that infect cloud computing environments. Moreover, we elaborate on different malicious software detection approaches in cloud computing environments. Furthermore, we evaluate these approaches by considering other perspectives (i.e., malware detection accuracy and deployed analytical techniques). More than 50% of the approaches of the malware detection papers (in this survey) used deep learning techniques in cloud computing environments. In addition, the majority of authors preferred to use dynamic malware analysis. Deep learning and dynamic analysis are powerful, complementary approaches in malware detection. Dynamic analysis observes the runtime behavior of programs, such as API calls, file operations, and network activity, to detect malicious patterns in controlled environments like sandboxes. When integrated with deep learning, this behavioral data can be analyzed more effectively using advanced models like RNNs or CNNs. Deep learning enhances dynamic analysis by identifying complex, hidden patterns in malware behavior and adapting to zero-day threats. This combination provides a robust defense mechanism, particularly in cloud computing, where large-scale and real-time detection capabilities are critical. The rates of detection are vacillated from 79% to 99%.

1. INTRODUCTION

Cloud computing offers diverse resources and services such as tools, connectivity, servers, and data storage over the Internet. In addition, cloud computing environments provide the capability to remotely use applications and outsource data storage. Consequently, there is no need for a permanent site to gain access to cloud content. In addition, cloud computing can fully or partially offload remote user devices. If an application is fully executed inside the cloud on the remote server side, this is called full offloading. The entire work was moved to powerful remote computing clusters. Otherwise, if an application is executed partially inside the cloud, it is known as partial offloading. In this case, the remaining part runs on the user device. The COVID-19 pandemic (started in 2020) mandates businesses to accelerate the digital transformation process and use cloud capabilities to sustain their business continuity objective. However, with the lack of physical access to applications and data in cloud computing environments, security issues (i.e., data and applications) increase in addition to platform security.

The cloud computing market continues to grow according to Grand View Research [1]. In 2023, the global market size of cloud computing was predictable at USD 602.31 billion. Additionally, the global cloud computing market size is predicted to rise at a CAGR (Compound Annual Growth Rate: the mean annual growth rate of an investment over a specified period longer than one year) of 21.2% from 2024 to 2030 [1]. Owing to various factors, the cloud-computing market is on a fast track.

Digital conversion between industries (i.e., the increasing diffusion of mobile devices and the Internet worldwide and the growing use of big data) are the main factors affecting market growth. Despite the progress in cybersecurity over the last two decades, statistics have shown a significant increase in malicious software activity and sophisticated frequent attacks.

According to a 2023 report [2] by Check Point Research, cloud-based attacks have increased by 45% year-over-year. Cloud misconfigurations, insecure APIs, and exploitation of cloud storage vulnerabilities are common attack vectors. Additionally, McAfee's 2023 Cloud Security Report [3] found that 77% of organizations reported experiencing a cloud security incident last year, with malware being a prominent part of these attacks.

Cloud computing technology is a trending and growing technology in the Information and Communications Technology (ICT) industry. The primary goal of cloud computing is to appropriately deliver services to legitimate

users. In cloud-computing environments, services are established and presented only on the cloud, and users can use them according to their requirements. Cloud computing offers numerous services to users, such as pay-per-use, low costs, and flexibility. Cloud users can use these services without purchasing or storing them in their internal memories.

Cloud computing comprises three basic models. These models are based on the type of service provided by the cloud to the users [4]. There are Platform as a Service (PaaS), Software as a Service (SaaS), and Infrastructure as a Service (IaaS), as illustrated in Figure 1. In SaaS, software or applications are hosted by a third-party provider for on-demand access. The PaaS model provides a platform and an environment for users to develop applications or services. In the IaaS model, resources and a virtualized computing machine are provided to a user to minimize the cost of purchasing their server and data center [5].

The cloud computing architecture comprises of five components: infrastructure, servers, platforms, applications, and clients. The National Institute of Standards and Technology (NIST) [6] defines five important cloud-computing characteristics: on-demand self-service, broad network access, resource pooling, measured service, and rapid elasticity. In addition, cloud computing is designated as a dynamic and easily extended platform that offers users transparent virtualized resources through the internet.

Cloud computing has four deployment models [4]: private, public, hybrid, and community. Public clouds have a server provider that owns and manages physical infrastructure. Private clouds have a specific organization that owns and operates the infrastructure. Community clouds have a consortium of organizations that own and manage their physical infrastructure. Hybrid clouds consist of a mixture of three previous models.

Cloud computing offers many services [6] to its users, such as broad network access, on-demand self-service, resource pooling, scalability, agility, measured service, pay-per-use cost, location, device independence, easy maintenance, efficiency, reliability, application programming interface (API), and productivity.

Productivity increases as many users work on the same data simultaneously instead of waiting for data to be saved and emailed. In addition, time will be saved because information does not need to be re-entered, and users will not need to install applications or software on their computers.

Malware detection in cloud computing environments is of paramount importance because of the critical role that clouds play in storing, processing, and transmitting vast amounts of sensitive data to businesses and individuals worldwide. The shared and distributed nature of cloud resources makes them attractive targets for cybercriminals seeking to exploit vulnerabilities, steal data, disrupt operations, or launch further attacks. A single compromised cloud instance can potentially affect multiple tenants, leading to data breaches, financial losses, reputational damage, and compliance violations. Cloud environments also host mission-critical applications and services, which, if disrupted, can cause significant operational downtime for businesses. In addition, the scalability and dynamic nature of cloud platforms, with frequent uploads, downloads, and virtual machine migrations, increase the risk of malware spreading rapidly if undetected. Modern malware often employs sophisticated techniques, such as obfuscation, encryption, and polymorphism, making detection challenging without advanced methodologies. Effective malware detection solutions are vital for real-time threat analysis, the detection of zero-day attacks, and the prevention of the lateral spread of malicious entities across cloud networks. These solutions ensure not only the security and privacy of data, but also uphold trust in cloud services, enabling businesses to innovate and operate without fear of cyber threats. As cloud adoption continues to grow, robust malware detection mechanisms are becoming indispensable for maintaining the integrity, availability, and confidentiality of cloud-based systems.

Although malware attacks are a crucial concern in cloud security, no previous study has discussed malware detection approaches in cloud computing environments or compared them to determine the most effective strategy. This study aims to improve current methods for malware detection in cloud computing environments. In this study, we discuss malware attacks in cloud computing environments. In addition, we provide a survey of different malware detection methods used in cloud computing environments. As a result, more than 50% of the authors of the malware detection papers (in this survey) use deep learning methods in cloud computing environments. In addition, dynamic malware analysis is favored by most authors because of its benefits, as mentioned in Section 3. The detection rate fluctuated between 79% and 99%. In the next section, we discuss cloud computing vulnerabilities and malware attacks. In Section 3, we describe different malware detection methods. Section 4 surveys malware detection approaches used in cloud computing environments. Finally, conclusions are presented in Section 5.

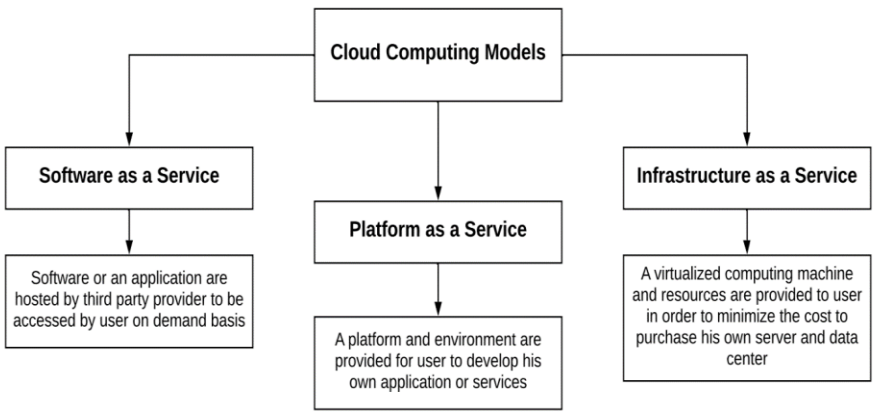

Figure 1. Cloud computing models

2. MALWARE ATTACKS AGAINST CLOUD COMPUTING

Cloud technology has introduced new concepts, such as centralized shared data and resource sharing. This creates new security challenges. In addition, direct access or indirect usage of cloud infrastructure increases cloud threats and vulnerabilities. Some of the new threats to cloud computing environments are as follows:

Vulnerabilities of Exploited Systems: These vulnerabilities are not new; however, they have become a more significant concern, particularly after multi-tenancy in cloud computing. New attack surfaces are created because memory, databases, and other resources are shared among users close to each other. Even if traditional security mechanisms can mitigate these attacks, they can only partially solve them.

Configuration vulnerabilities: Many VMs have the same configuration in a cloud environment; therefore, they have similar misconfigurations and vulnerabilities. This increases the ability of many VMs to become infected using malware and gives attackers a tremendous opportunity to target systems in cloud-computing environments.

Threats by Insiders: This malware infection method. This could occur through multiple methods, such as previous or current employees. The objectives range from data theft to vengeance.

Easily Compromised Credentials: This threat can occur in different ways, such as weak passwords, open passwords, or embedded passwords in source code.

Malware can be considered a high-risk threat when performing cyberattacks. Malware refers to malicious software designed to intentionally harm or disrupt digital assets, stealing, or damaging them. Malware writers use evasive techniques to introduce malware files into a victim's system. There are various types of malware. They can be classified according to their intent and written code, such as viruses, worms, rootkits, backdoors, and ransomware.

Cloud infrastructure [4-11] has become increasingly susceptible to malware and novel attacks. Cloud malware injectionis a threat injected into a victim's Vi [12] ritual Machine (VM) to manipulate it. Many VMs have the same configuration because of automatic provisioning in cloud computing. Thus, if an attacker injects malware into one VMs and compromises it, it is most likely to compromise other VMs with the same configuration. Additionally, botwareware can be injected to create a botnet that benefits from many available VMs. Another scenario of Cloud Malware Injection Attack) [11-13] is when an attacker injects a malicious virtual machine or malicious services into the cloud environment. In this attack, the attacker implements a malicious module, such as SaaS, PaaS, or VM, such as IaaS, and attempts to place it in the cloud system. Then, he/she pretends to be a valid service for cloud systems, and it looks like the deployment of new services, such as current services. If the attacker penetrates the system, the cloud routinely redirects the user request to the malicious service implementation, so that the injected code starts to execute. Attackers often use this method to target the cloud service layers.

Attackers are continuously creating new malicious software such as:

Hypercall Attacks: An attacker's virtual machine exploits the hypercall handler of the victim's Virtual Machine Manager (VMM). This may give the attacker the ability to run arbitrary code.

Man in the Middle (MITM): An attacker overhears the changing messages between two communicators.

Distributed Denial of Service (DDoS)attacks: A large number of queries to a service that can be performed using a botnet to shut down the service or to increase its latency.

Hypervisor denial of service (DoS): An attacker exploits design flaws using numerous hypervisor resources.

Hyperjacking: An attacker attempts to gain control over a VM hypervisor to access an entire machine.

Co-location: An attacker wants to perform cross-side-channel attacks by finding the location of the virtual machine host and the virtual machine alongside it.

Live Migration Attack: As virtual machines can be migrated between cloud services, an attacker can abuse the service to generate several migrations, which may lead to DoS attacks.

Thus, increasing the vulnerability surface can be an entry point for a considerable amount of malware, which can also be the initial step in performing more complicated attacks such as DDoS. Therefore, malware detection methods in the cloud have become essential.

3. MALWARE DETECTION APPROACHES

Malware is one of the main threats to information security. A malware is a type of malicious software. It is any software that executes malicious activities on a victim's machine, with or without the victim's knowledge. Different types of malwares exist, such as viruses, worms, trojans, ransomware, rootkits, and backdoors.

Malware detection identifies malware in a system and then examines the malware file to understand its capabilities and the changes made in the system. Several approaches including traditional and new techniques have been proposed for malware detection.

There are several types, behaviors, and levels of malware risk, and evasion techniques have rapidly changed to deceive detection systems. Therefore, modern detection methods and mechanisms must be employed. Having more than one security software to professionally deal with malware is unfeasible.

3.1 Static analysis, dynamic analysis, and hybrid analysis

The proposed malware detection methods use one of three techniques–static, dynamic, or hybrid analysis–to extract features used in different methods. Malware analysis is the process of examining malware files in order to understand their capabilities and system changes.

3.1.1 Static analysis

This involves analyzing samples without executing them [14, 15]. A sample was broken down in a static analysis using reverse engineering tools and techniques to reconstruct the source code. Static analysis can be performed using a program analyzer, debugger, and disassembler.

There are two main approaches to the static analysis. In the first step, an analysis is performed on a binary file, such as extracting features by collecting parts from the binary file (n-grams). In the second approach, the binary file is disassembled or reverse-engineered using disassemblers to acquire actual code. Malware detection occurs in real code using different techniques. Various machine learning methods can then be used.

Static analysis is inexpensive, fast, and very effective; however, it can be avoided using sophisticated malware, for example, by embedding syntactic code errors that could confuse disassemblers. Additionally, polymorphic malware can change and evolve while saving code semantics. Static analysis is complex when malware analysts deal with polymorphisms and encrypted, packed, or obfuscated samples. Many malware programs use obfuscation, where binary code is unreadable or challenging to understand. Packing can be used to avoid static analysis approaches or malware can be modified using a run-time encryption program. A dynamic analysis technique is necessary because of the difficulties associated with statically analyzing sophisticated malware.

3.1.2 Dynamic analysis

The dynamic analysis process involves executing instructions from a malicious sample file in a simulated environment and analyzing the behavior or actions of the application during execution [14, 15]. This can be achieved by monitoring function calls, analyzing function parameters, instruction tracing, and information flow tracking. A virtual machine or sandbox can be used for dynamic analysis.

Unlike static inspection, dynamic analysis approaches are used to overcome static analysis drawbacks, because they depend on behavior monitoring. In dynamic analysis, malware activities are monitored for a few minutes after executing them in a closed environment such as a virtual machine, sandbox, or emulator. The activity was monitored for a few minutes. A clean environment was created for each sample to prevent infection. Although dynamic analysis is used to overcome static analysis challenges, it can also be avoided by other methods such as delayed execution.

Malware uses a delayed execution technique when it does not display its malicious activities for an extended period, rendering the analysis process useless. Increasing the analysis time could be more effective and feasible, and malware may increase its waiting time. Additionally, malware may attempt to discover the presence of a sandbox or an emulator. Once identified, the malicious activity is stopped. In addition, if there is no Internet connection, some malware will stop malicious activities as it attempts to connect to its command and control.

3.1.3 Hybrid analysis

Both static and dynamic analyses have their advantages and disadvantages. Therefore, both were used in the hybrid analysis to obtain improved results.

This technique uses a combination of static and dynamic analyses [15]. Firstly, the malware signature is checked to determine whether it exists in the inspected code, and then, the sample behavior is monitored. Therefore, this technique can benefit from the advantages of both the analytical methods.

4. MALWARE-DETECTION IN CLOUD COMPUTING

Various malware-detection approaches have been proposed. Traditional detection methods are signature-based techniques that function well and quickly with the known malware. This problem arises when dealing with unknown malware such as zero-day malware. This makes researchers use new and different methods such as behavior-based detection, heuristic-based detection, and deep learning.

4.1 Signature-based detection technique

First, malware analysis methods extract features from malware samples to create signature databases. The signature was then generated and stored in a signature database. When a sample must be tested to mark it as malware or benign, the signature of the sample is collected using a technique similar to that of the signature database. This was then compared to the signatures in the database. Depending on the comparison results, it can be classified as sample file malware or benign file, as shown in Figure 2.

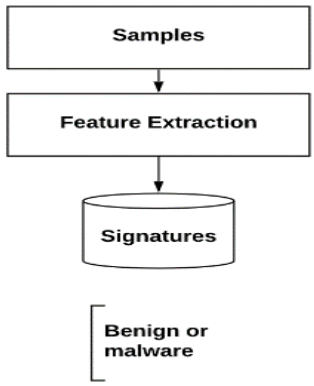

Figure 2. Signature-based detection technique

4.2 Behavior-based detection technique

In this method, the behavior of a sample is observed after it is executed by monitoring the system calls, processes, registry, file changes, and network to detect whether the sample is malware or benign. Thus, there were no problems in detecting new malware using this method. By contrast, a problem arises in malware uses anti-dynamic analysis methods to detect whether it is running in a protected environment, such as virtual machines or sandboxes, to stop malicious behavior. Thus, malware that uses this anti-dynamic analysis method may be incorrectly marked as benign.

Mishra et al. [16] proposed an introspection-based malware detection approach. The proposed out-VM monitoring approach is called vProVal. It is designed to operate in a Kernel VM (KVM)-based cloud environment to identify rootkits and hidden processes. The proposed approach detects malware outside a VM, thereby making it more reliable against attacks. Process logs were extracted from the system after sample execution. It is compared with the process log extracted from the memory from outside the VM to check for the existence of security-critical processes and to detect hidden processes.

Gan et al. [17] discussed malware propagation through different virtual machines in a cloud computing architecture, particularly infrastructure as a service (IaaS). A dynamical propagation model was presented to determine the essential factors that affect malware spread, and the impact of installing antivirus software on VMs was studied. A theoretical analysis of this model is conducted using differential dynamics, from which it is possible to understand the dissemination behavior of malware in an infected cloud environment. In addition, a numerical simulation is performed to validate the applicability and effectiveness of the proposed model.

Al-Khafaji et al. [18] proposed an overview of packet sniffing tools in the IoT and cloud-based environments.

Mishra et al. [19] discussed making a cloud environment more resistant to malware threats. Cloud environments have become more vulnerable to cross-VM attacks because of the fingerprints and artifacts generated by the traditional virtualization software. Thus, current mainstream hardware-assisted virtualization is attempting to enhance transparency. The authors proposed an attack scenario to demonstrate how an attacker can use the following three features to detect hardware-assisted virtualization: saving an extra layer of address translations in the Translation-Lookaside Buffer (TLB), the existence of one more layer of page-table entries in the LLC cache, and instability in the Level-1 Data (L1D) cache. They successfully performed attacks in three native environments: Amazon Elastic Compute Cloud, Google Compute Engine, and Microsoft Azure.

4.3 Heuristic-based detection technique

As shown in Figure 3, both static and dynamic extracted features can be used to benefit from the static and dynamic analysis methods of this technique [20]. Subsequently, the features were input into different machine learning algorithms for training. Machine learning algorithms can then classify the tested sample as malicious or benign.

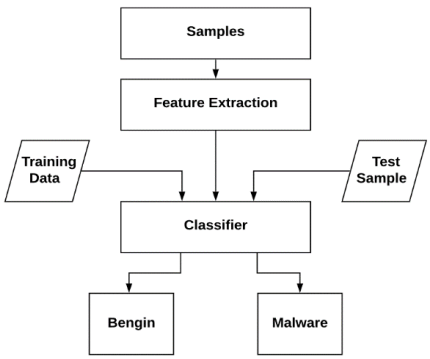

Figure 3. Heuristic-based detection technique

Abdullayeva [21] proposed a malware detection approach based on image similarity. The authors used the Malimg dataset. It contains 9,339 malware-byteplot images from 25 families. The samples were then converted into RGB image representations (red, green, and blue). The images were input into a Gaussian Mixture Model to detect file similarities. The accuracy was 79.21%.

Bedi et al. [22] discussed different types of attacks on the cloud environment, such as breach of confidentiality, denial of service, cloud malware injection, side-channel, man-in-the-middle cryptographic, and authentication attacks. Subsequently, they proposed a new model for overcoming these challenges. The proposed antivirus consisted of three components. The first component is the host agent, which transfers all the new files to the network service, and the second component is the network service. The network service analyzes all received files to classify them as safe or unsafe. The last component is the forensic service, which is responsible for keeping records for all analyzed files and creating an alert interface.

Fui et al. [23] proposed a detection system based on a dynamic malware analysis and machine learning. They used three classifiers: Random Forest, J-48, and naive Bayes. The XEN cloud platform was used as the test cloud. The dataset comprised 9000 samples from the Kaggle database. The samples were executed on a virtual machine and their behavior was monitored using a cuckoo monitoring server to collect the feature vector. The accuracy was found to be 99%.

Kumar et al. [24] proposed a malware-detection approach based on clustering techniques. The samples were tested in an isolated environment. Dynamic analysis was performed using a cuckoo sandbox and API calls from cuckoo sandbox reports to form a feature vector. Feature selection was performed using Principal Component Analysis (PCA), random forest, and chi-square tests. They used three classification algorithms: a Decision Tree, Random Forest, and Logistic Regression.

Abawajy et al. [25] presented a malware detection approach called Hybrid Consensus Pruning (HCP). In this approach, several classifier classes were aggregated into a single scheme. To test the effectiveness of the HCP method, Abawajy et al. [25] conducted experiments to compare its performance with that of Ensemble Pruning via Individual Contribution (EPIC) ordering, Directed Hill-Climbing Ensemble Pruning (DHCEP), and K-Means Pruning approaches for pruning very large ensemble classifiers for malware detection. Byte sequences or n-grams were used in this study. Sequences of n bytes were extracted from executable files to be examined. N-grams yield compelling static features for malware detection. The experimental results show that the HCP achieves better results by producing better ensemble classifiers than those created by EPIC, DHCEP, and K-Means Pruning.

Mishra et al. [19] proposed VMShield, an introspection-based security approach for securing virtual machines and detecting malware in a cloud infrastructure. VMShield introduced virtual memory introspection from a hypervisor to gather runtime process behavior. The proposed approach uses the existing techniques to detect stealthy malware. Random forest was used to classify the samples after extracting the feature vector of the sample.

Kubernetes were used in a previous study [26]. It is an open-source system that automates cloud-application deployment, scaling, and management. The proposed crypto-miner detection system in a cloud environment was based on machine learning. We extracted a sequence of system calls. The feature vectors are input into different machine-learning algorithms, such as decision trees, ensemble learning, feedforward vanilla artificial neural networks, and feedback Recurrent Neural Networks (RNNs).

4.4 Deep learning-based detection technique

This new detection method is used in several malware detection approaches for cloud computing, as discussed in the next section. It is a subset of machine learning methods that uses neural networks with many layers to analyze data and learn from it, as shown in Figure 4. Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) are the most widely used deep learning architectures. These models are particularly effective for various types of data and tasks.

RNNs are designed to handle sequential data in which the order of information is important. Unlike traditional neural networks, RNNs have connections that allow the persistence of information. This memory aspect enables them to make predictions based on both the current and previous inputs in a sequence. It is Effective for tasks that require understanding of the context or previous data points in a sequence. In addition,

it is useful in any application involving time-dependent data, such as audio or text. RNNs can suffer from the vanishing gradient problem, where they struggle to learn long-term dependencies in data. In addition, training can be computationally expensive and time consuming.

CNNs are designed to automatically and efficiently detect patterns in visual data such as images or videos. They use convolutional layers that apply filters to the input data to detect local patterns, such as edges, textures, or specific shapes. CNNs then use pooling layers to reduce the spatial dimensions, allowing them to focus on more important features, which is highly effective for visual data because of its ability to detect hierarchical patterns (e.g., edges, shapes, and textures) at different scales. It automatically learns spatial hierarchies in images, thereby eliminating the need for manual feature engineering. On the other hand, a CNN requires a large amount of labeled data for effective training. Furthermore, it is computationally intensive, particularly for large datasets or deep architectures.

Ergen et al. [27] proposed a malware detection approach based on deep learning using RNNs. They focused on two architectures: long short-term memory RNNs (LSTMs) and Bidirectional RNNs (BIDIs). They used 40,680 samples as datasets. They collected the behavior of each sample after execution in an open online cloud environment with no restrictions. The feature vector contains a sequence of running processes in the VM. The classification accuracy was 99%.

Payne and Kundu [28] presented a hierarchical approach for the development of malware detection systems. They used attention language models to analyze system logs and shape their respective systems in a standardized manner for downstream processing. They used graph and hypergraph learning problems to detect malware in the cloud. They assumed a multi-cloud scenario, in which multiple untrusting clouds cooperate to learn the state of malware without divulging private or sensitive information. In addition, the authors discussed different open problems in cloud-computing environments that defend against malware attacks.

Li et al. [29] discussed the significant problem of malware threats in cloud computing environments, as many hosts are connected with high-risk trust assumptions and security mechanisms that are not difficult to break. Detecting malware propagation is difficult because malware may remain in several components through software or hardware stack. In this case, it is more beneficial to contain malware in the smallest possible number of hosts, and it is also essential for the system administrator to fix the problem promptly. The authors defined the problem and presented their idea of decentralized malware containment as well as the challenges and issues related to this idea. They presented the basic implementation of this approach.

Malvankar et al. [30] presented a malware detection system based on deep learning. In the training stage, CNN was used to build a model of malicious and benign software memory snapshots. This model was used in the testing stage to detect and classify the malware. Images from the virtual machine were extracted after executing malicious and benign software. They converted extracted memory images into grayscale images. Grayscale images were used to train CNN. It operates in the WMM layer to ensure that its system is secure and transparent, and incurs less overhead. More than 10000 malwares were used in the prototype evaluation. The accuracy reached 90.5%.

Additionally, Mishra et al. [31] proposed another malware detection system based on deep learning called VMAnalyzer. It uses machine learning algorithms to detect attacks on the VM layer in a cloud environment. The sequence of the system calls was extracted from suspicious programs, and two classification layers were performed. CNN is used in the first layer to select the relevant system call sequence. The output of layer 1 was used as the input for layer 2. Bidirectional long short-term memory (LSTM) is used in layer 2 to detect the behavior of a malicious sequence of system calls. The dataset used in this study is obtained from the University of New Mexico.

Nahmias et al. [32] presented TrustSign, an automatic malware signature generation method based on high-level deep features conveyed by a VGG-19 neural-network model. The proposed system benefits from virtualization, which exists at the core of the cloud architecture. It uses malicious processes present in volatile images to produce signatures. Thus, these systems can detect fileless malware. The classification accuracy reached 99.5%.

Another malware detection approach based on deep learning techniques was presented by Abdelsalam et al. [33]. Malicious files were executed on a virtual machine, and process behavior data were extracted. The extracted data were inputted into a 2D CNN. They improved classification accuracy using a 3D CNN. The classification accuracy was 90%.

Jeon et al. [34] proposed a dynamic analysis system called Dynamic Analysis for IoT Malware Detection (DAIMD). DAIMD dynamically analyzes malware in a nested virtual environment instead of in an IoT device to detect obfuscated and code-changeable files. It extracts memory, networks, processes, system calls, and virtual file systems to identify malicious behavior. The extracted data are then converted to images and applied to the CNN to classify the IoT malware.

Chai et al. [35] proposed a joint framework for malware detection using local and global features. It is called LGMal. It uses a stacked CNN to collect API call sequence information, which helps in local semantic feature collection. In addition, it uses graph convolutional networks to gather API call semantic graph structure information, which helps in global semantic feature collection. They used Alibaba Cloud Security Malware Detection datasets.

Kotian and Sonkusare [36] presented a malware-detection approach with a classification accuracy of 95%. The running information, such as the CPU, memory, and network parameters, was collected and fed into the CNN algorithm. The accuracy was improved by using a 2D CNN.

Samuel et al. [37] presented Intelligent Behavior-Based Malware Detection (BBMD) as a new malware detection framework. It uses artificial neural networks (ANNs) and deep learning. The BBMD analyzes cloud resource behavior to detect abnormal behavior. It uses both system logs and network traffic to provide an overall view of system behavior, and to provide comprehensive solutions for cloud computing environments. BBMD has the ability to integrate with other existing security products such as intrusion detection systems and firewalls.

Mustafa et al. [38] proposed a White Shark Optimization approach. Two datasets (NSL-KDD and Kyoto) were used for the training. A support vector machine was used for classification after the feature extraction. They achieved a detection rate of 99.8%.

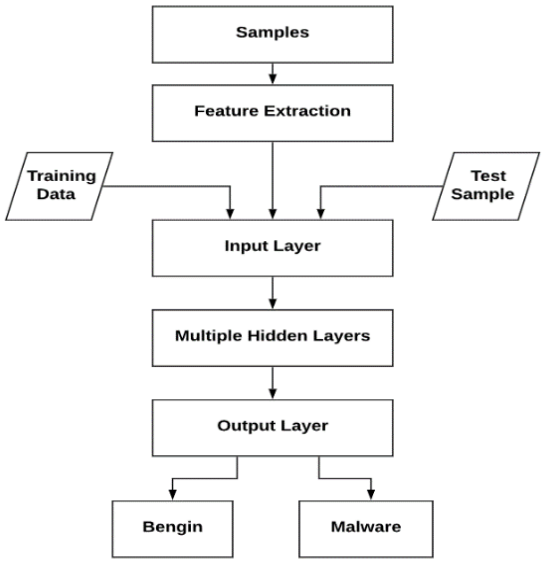

Figure 4. Deep learning-based detection technique

As previously mentioned, detection methods differ in terms of their technicality, such as detection speed, performance overhead, and use cases. Table 1 compares the features of different detection techniques. Signature-based detection is highly effective for known threats, but struggles with new or unknown malware. Behavior-based detection is excellent for detecting unknown threats through analysis of actions, but may have performance and false-positive issues. Deep Learning-Based Detection offers strong detection capabilities for both known and unknown malware; however, it requires significant resources and data to function effectively. Heuristic-Based Detection strikes a balance between detecting new threats and being resource efficient, although it may suffer from false positives.

Malware poses a significant threat to cloud computing. If malware succeeds in compromising a virtual machine, it can steal the critical information or attack other cloud nodes. Several approaches have been proposed to analyze executable binaries to identify malware in the cloud, particularly zero-day malware.

Table 1. Comparison between the different detection techniques features

Feature	Signature-Based Detection	Behavior-Based Detection	Deep Learning-Based Detection	Heuristic-Based Detection
Detection of Known Malware	High (fast matching of signatures)	Low (needs behavior to trigger)	High (trained on large datasets)	Medium (based on characteristic patterns)
Detection of Unknown Malware	Low (can't detect new threats)	High (detected by behavior)	High (learns to detect new threats)	High (identifies anomalous behaviors)
False Positives	Low (if signatures are accurate)	Medium to High (depends on behavior)	Medium (depends on training data)	High (sensitive to benign anomalies)
Performance Impact	Low (fast with minimal overhead)	High (requires constant monitoring)	High (needs computational resources for training and inference)	Medium (depending on the complexity of heuristics)
Adaptability to New Threats	Low (only effective for known threats)	High (can adapt to new types of behavior)	High (learns and adapts from new data)	Medium (depends on heuristics evolution)
Complexity	Low (simple signature matching)	Medium (needs monitoring of behavior)	High (requires machine learning infrastructure)	Medium (requires constant refinement of heuristics)
Resource Consumption	Low (signature database)	High (real-time monitoring)	High (computationally intensive)	Medium (depends on system implementation)

However, traditional malware-detection approaches are inappropriate for cloud computing. Static analysis-based approaches require malware binary files that may not be permanently retained in the file system of a virtual machine. Malware may hide or remove itself from virtual machine disks. Therefore, malware scanners may not be able to locate binary files. However, the runtime overhead introduced by dynamic-analysis-based approaches cannot be accepted in natural cloud computing. In addition, analysis tools in VMs can be detected and exploited as malware runs inside VMs. Recently, several approaches have been proposed for malware detection using cloud computing. Most of the proposed approaches depend on dynamic analysis, as shown in Figure 5. Dynamic analysis detects runtime behaviors and activities such as system calls, network communications, and privilege escalation. It effectively detects packed or encrypted malware that hides its code during runtime. It can identify zero-day threats and advanced malware using behavioral indicators. However, complex malware may detect the analysis environment and alter its behavior to avoid detection. In addition, dynamic analysis may miss dormant functions that are not activated in the monitored environment. Compared to static analysis, it is resource-intensive. It requires execution in isolated environments, leading to higher computational and storage overheads. In addition, this method is time-consuming. The monitoring of runtime behaviors can be slower than that of the static analysis. In addition, the need for controlled execution environments can limit the scalability of large cloud ecosystems.

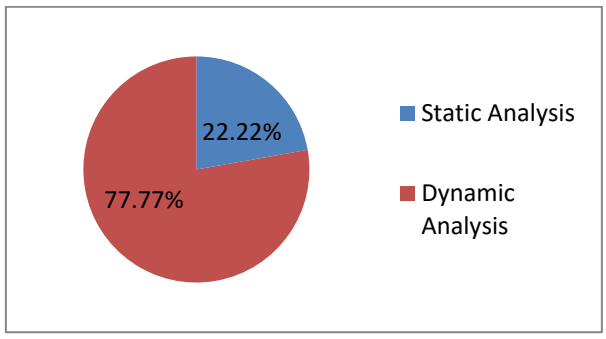

Figure 5. Dynamic analysis vs. static analysis

Most of the presented approaches use deep learning techniques, as shown in Figure 6. Unlike traditional methods, deep-learning models do not require extensive manual feature engineering. They learn features directly from raw data. In addition, it can analyze vast amounts of data to detect malware patterns and behaviors, even those previously unseen.

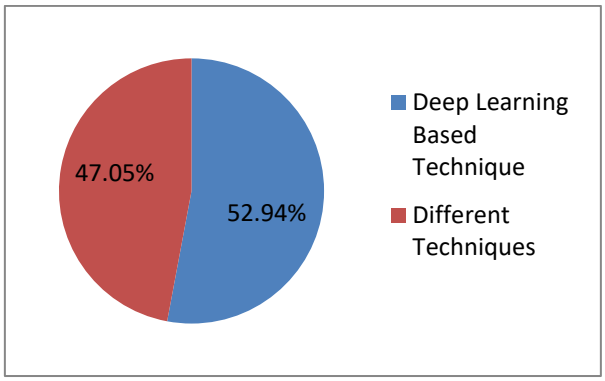

Figure 6. Deep learning-based technique vs. other techniques

Deep learning is highly scalable. It is designed to handle large-scale heterogeneous datasets typical of cloud environments. Frameworks like TensorFlow and PyTorch allow seamless integration with cloud platforms for training and inference. However, traditional detection methods are resource-efficient. These methods are generally lightweight and require less computational resources, making them suitable for less complex environments. However, they struggle to handle the scale and diversity of data in cloud environments effectively. Therefore, they have limited scalability.

Both deep-learning-based detection techniques and traditional detection methods are employed, each with its own strengths and limitations. Deep learning suffers from a computational overhead. Training deep learning models requires significant computational power, which can be resource intensive in a cloud environment. The effectiveness of deep-learning models depends heavily on the quality and diversity of the training data. Thus, deep learning is data-dependent. Deep-learning models are often considered black boxes, making it difficult to interpret the detection results. However, the traditional methods are ineffective against advanced threats. They struggle with advanced persistent threats (APTs) and zero-day attacks. They are inflexible and require frequent updates to be effective against new malware strains.

Accuracy was achieved using different malware detection approaches in a cloud-computing environment, as shown in Figure 7.

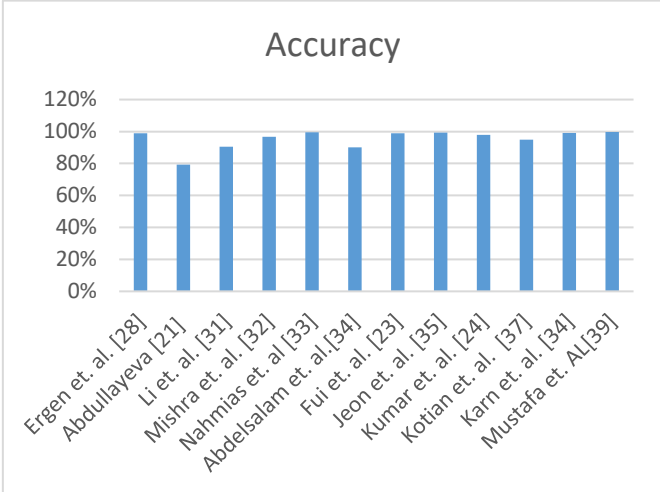

Figure 7. Accuracy

Deep learning approaches have improved detection rates. It can be generalized better to new and unseen malware, thereby reducing false negatives. In addition, they are resilient to evasion techniques. Deep learning models can detect subtle changes in malware behavior that traditional methods may overlook. However, traditional detection approaches have high precision for known malware. Signature-based methods excel at detecting known threats, but suffer from high false negatives for new or obfuscated malware. Rule-based systems may misclassify benign applications if rules are too stringent.

A summary of the proposed malware detection approaches in cloud computing environments is presented in Table 2.

Table 2. Summary of proposed methods

Ref.	Proposed Method	Classification Feature	Used Dataset	Accuracy	Year
[27]	They use RNNs.	Runtime processes system features	40,680 samples	99%.	2018
[21]	The images are inputted into Gaussian Mixture Model.	Features based on the similarity of images	in Malimg dataset	79.21%	2019
[28]	They use inductive graphs and hypergraphs neural network models such as Graph Convolutional Networks, GraphSAGE, Graph Attention Networks, and Deep Hyperedges.	System logs	DARPA IDS evaluation dataset	Not mentioned	2019
[29]	They use graph convolutional networks, graph attention networks.	Using graph analysis	Not mentioned	Not mentioned	2019
[30]	They converted extracted memory images into grayscale images. grayscale photos used to train CNN.	memory images	More than 10000 malware	90.50%	2019
[31]	The sequence of system calls extracted from suspicious programs is then fed into two layers of classification.	The sequence of system calls	University of New Mexico dataset	81.72%-96.67%	2019
[32]	The malicious processes extracted are converted into images and inputted for a deep neural network model.	Malicious process in memory images	The dataset contains samples from both web-based and desktop Monero mining applications	99.5%.	2019
[19]	Processes are extracted from the system after executing the sample and compared with the process log, which is extracted from memory outside the VM, to check the existence of security-critical processes and detect hidden processes.	Processes log	The dataset collected from the University of California	In the future, they will perform a more detailed analysis	2019
[33]	The malicious files are executed on the virtual machine, and process behavior data are extracted. The extracted data are inputted to 2D CNN.	Process behavior data	4500 samples	86%-90%	2018

[23]	The samples are executed in the virtual machine, and the behavior is monitored using a Cuckoo monitoring server.	Malware behavior such as Kernel Function Calls	9000 samples from the Kaggle database	99%	2020
[34]	They extracted malware behavior data, and then the extracted data were converted to images and applied to CNN to classify IoT malware.	Malware behavior, such as memory, network, and system calls	840 images in the training dataset using the ZFNet model.561 files in the test dataset	90.01-99.28%	2020
[24]	These samples are being compiled in an isolated environment. They performed dynamic analysis using the Cuckoo sandbox, and then API calls from the Cuckoo sandbox reports forming the feature vector.	API calls	2780 sample files	97%	2020
[25]	A new multi-stage hybrid consensus pruning approach to combine several classifier classes into one scheme.	Byte sequences, or n-grams	Collected from the honeynet and VH Heavens	the ensemble that achieved the best value of AUC in comparison to other ensembles, also performed best concerning other metrics.	2020
[35]	They combine the stacked CNN and graph convolutional networks.	API call sequence information	Alibaba Cloud Security Malware Detection Datasets	precision is 87.76%, the recall is 88.08%, and the F1 measure is 87.79%.	2020
[36]	Malware behavior is collected and fed to the CNN algorithm.	Running information such as CPU, memory, and network parameters		95%	2021
[16]	Malware behavior data collected and inputted from random forests.	the run-time processes behavior			2021
[26]	The sequence of system calls is inputted to different machine learning algorithms such as Decision Tree, Ensemble Learning, Feed-Forward Vanilla Artificial Neural Network, and Feedback Recurrent Neural Network.	Syscall and CPU usage		99%	2021

5. CONCLUSION

The rapid growth of cloud computing has attracted various attack types, particularly malware. Additionally, the flexibility, elasticity, and other services provided by the cloud computing architecture render the cloud environment more vulnerable to novel attacks. In this study, we elaborate on malware attacks in cloud computing environments. In addition, we present some recent malware detection approaches that provide malware detection in a cloud computing environment. More approaches are needed to resist the extraordinary increase in malware attacks and countermeasure the sophistication of malware samples designed to deceive the current security systems. These approaches must consider the architecture of the cloud computing environment. The main limitation of our research is that some approaches should have explained their methodology in detail, and some authors should have mentioned the datasets used in their approach. In recent studies, the malware detection rate has reached 99%. Deep learning techniques have become powerful tools in malware detection, leveraging their ability to automatically learn complex patterns and features from large datasets. Deep learning models can analyze static code features, dynamic behavioral data, or network traffic to detect malicious activities with high accuracy using architectures such as CNNs, RNNs, and transformers. These models can be used to identify zero-day threats and polymorphic malware that can evade traditional detection methods. Their scalability and adaptability make them particularly suited for cloud-computing environments, where vast amounts of data and evolving threats require robust real-time analysis. In the future, we would like to develop a hybrid framework for malware detection, particularly in cloud environments. We want to benefit from previous approaches to achieve a higher detection rate with a lower false-positive rate, while considering the processing time.

AUTHORS' CONTRIBUTIONS

Doaa Abdelrahman: Conceptualization (equal); Funding Acquisition (equal); Writing, Review & Editing (equal); Formal Analysis (equal); Methodology (equal). Mohamed Rasslan: Conceptualization (equal); Funding Acquisition (equal); Writing & Editing (equal); Formal Analysis (equal); methodology (equal). Nashwa Abdelbaki: Conceptualization (equal); review (equal); Formal Analysis (equal); methodology (equal). The authors have read and approved the final manuscript.

FUNDING STATEMENT

This work was supported by the National Telecom Regulatory Authority (NTRA).

REFERENCES

[1] Market Analysis Report. (2020). Cloud computing market size, share & trends analysis report by service (SaaS, PaaS, IaaS), by workload, by deployment, by enterprise size, by end-use, by region, and segment forecasts, 2022–2030. Grand View Research, Report ID: GVR-4-68038-210-5, April 2020.

[2] Check Point Research, 2023 Cloud Security Threats Report.
[3] McAfee, 2023 Cloud Security Report.
[4] Bundela, R., Dhanda, N., Gupta, K.K. (2024). Identification and analysis of security issues in cloud computing. In 2024 2nd International Conference on Disruptive Technologies (ICDT), Greater Noida, India, pp. 1685-1690. https://doi.org/10.1109/ICDT61202.2024.10489443
[5] Goel, P.K., Singhal, A. (2023). Security issues and threats in cloud computing: Problems and solutions. In 2023 3rd International Conference on Advancement in Electronics & Communication Engineering (AECE), Ghaziabad, India, pp. 1019-1023. https://doi.org/10.1109/AECE59614.2023.10428390
[6] The NIST definition of cloud computing. SP 800-145, NIST, 2011.
[7] Mishra, N., Singh, R.K. (2019). Taxonomy & analysis of cloud computing vulnerabilities through attack vector, CVSS and complexity parameter. In 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), Ghaziabad, India, pp. 1-8. https://doi.org/10.1109/ICICT46931.2019.8977667
[8] Nancy, Silakari, S., Chourasia, U. (2016). A survey over the various malware detection techniques used in cloud computing. International Journal of Engineering Research & Technology (IJERT), 5(2): 398-402.
[9] Ranjan, I., Agnihotri, R.B. (2019). Ambiguity in cloud security with malware-injection attack. In 2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, pp. 306-310. https://doi.org/10.1109/ICECA.2019.8821844
[10] Mattoo, A.S., Upadhyay, D., Dubey, A.K., Shukla, M.K. (2020). An approach to analyze and protect data on an untrusted cloud network. In 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Uttar Pradesh, India, pp. 139-144.
[11] Bove, D., Müller, T. (2019). Investigating characteristics of attacks on public cloud systems. In 2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), Paris, France, pp. 89-94. https://doi.org/10.1109/CSCloud/EdgeCom.2019.00-13
[12] Chatterjee, M., Datta, P., Abri, F., Namin, A.S., Jones, K. S. (2020). Cloud: A platform to launch stealth attacks. In 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, pp. 1558-1563. https://doi.org/10.1109/COMPSAC48688.2020.00-33
[13] Chatterjee, M., Datta, P., Abri, F., Namin, A.S., Jones, K.S. (2020). Abuse of the cloud as an attack platform. In 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, pp. 1091-1092. https://doi.org/10.1109/COMPSAC48688.2020.0-125
[14] Kushala, M.V., Shylaja, B.S. (2020). Recent trends on security issues in multi-cloud computing: A survey. In 2020 International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, pp. 777-781. https://doi.org/10.1109/ICOSEC49089.2020.9215303
[15] Aslan, Ö.A., Samet, R. (2020). A comprehensive review on malware detection approaches. IEEE Access, 8: 6249-6271. https://doi.org/10.1109/ACCESS.2019.2963724
[16] Mishra, P., Verma, I., Gupta, S., Rana, V.S., Kadarla, K. (2019). vProVal: Introspection based process validation for detecting malware in KVM-based cloud environment. In 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC), Rome, Italy, pp. 271-277. https://doi.org/10.1109/FMEC.2019.8795365
[17] Gan, C., Feng, Q., Zhang, X., Zhang, Z., Zhu, Q. (2020). Dynamical propagation model of malware for cloud computing security. IEEE Access, 8: 20325-20333. https://doi.org/10.1109/ACCESS.2020.2968916
[18] Al-Khafaji, H.M., Alomari, E.S., Majdi, H.S. (2020). Review of analytics tools on traffic for IoT and cloud based network environment. In 2020 3rd International Conference on Engineering Technology and its Applications (IICETA), Najaf, Iraq, pp. 73-77. https://doi.org/10.1109/IICETA50496.2020.9318914
[19] Mishra, P., Aggarwal, P., Vidyarthi, A., Singh, P., Khan, B., Alhelou, H.H., Siano, P. (2021). VMShield: Memory introspection-based malware detection to secure cloud-based services against stealthy attacks. IEEE Transactions on Industrial Informatics, 17(10): 6754-6764. https://doi.org/10.1109/TII.2020.3048791
[20] Ye, Y., Li, T., Adjeroh, D., Iyengar, S.S. (2017). A survey on malware detection using data mining techniques. ACM Computing Surveys (CSUR), 50(3): 1-40. https://doi.org/10.1145/3073559
[21] Abdullayeva, F. (2019). Malware detection in cloud computing using an image visualization technique. In 2019 IEEE 13th International Conference on Application of Information and Communication Technologies (AICT), Baku, Azerbaijan, pp. 1-5. https://doi.org/10.1109/AICT47866.2019.8981727
[22] Bedi, A., Pandey, N., Khatri, S.K. (2019). Analysis of detection and prevention of malware in cloud computing environment. In 2019 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, pp. 918-921. https://doi.org/10.1109/AICAI.2019.8701418
[23] 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
[24] Kumar, R., Sethi, K., Prajapati, N., Rout, R.R., Bera, P. (2020). Machine learning based malware detection in cloud environment using clustering approach. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, pp. 1-7. https://doi.org/10.1109/ICCCNT49239.2020.9225627
[25] Abawajy, J.H., Chowdhury, M., Kelarev, A. (2015). Hybrid consensus pruning of ensemble classifiers for big data malware detection. IEEE Transactions on Cloud Computing, 8(2): 398-407. https://doi.org/10.1109/TCC.2015.2481378
[26] Karn, R.R., Kudva, P., Huang, H., Suneja, S., Elfadel, I.M. (2020). Cryptomining detection in container clouds using system calls and explainable machine learning. IEEE Transactions on Parallel and Distributed Systems, 32(3): 674-691. https://doi.org/10.1109/TPDS.2020.3029088
[27] Ergen, T., Sahin, S.O., Kozat, S.S. (2018). Recurrent

neural networks based online learning algorithms for distributed systems. In 2018 26th Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey, pp. 1-4. https://doi.org/10.1109/SIU.2018.8404806

[28] Payne, J., Kundu, A. (2019). Towards deep federated defenses against malware in cloud ecosystems. In 2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), Los Angeles, CA, USA, pp. 92-100. https://doi.org/10.1109/TPS-ISA48467.2019.00020

[29] Li, H., Zhan, D., Liu, T., Ye, L. (2019). Using deep-learning-based memory analysis for malware detection in cloud. In 2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW), Monterey, CA, USA, pp. 1-6. https://doi.org/10.1109/MASSW.2019.00008

[30] Malvankar, A., Payne, J., Budhraja, K.K., Kundu, A., Chari, S., Mohania, M. (2019). Malware containment in cloud. In 2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), Los Angeles, CA, USA, pp. 221-227. https://doi.org/10.1109/TPS-ISA48467.2019.00036

[31] Mishra, P., Khurana, K., Gupta, S., Sharma, M.K. (2019). VMAnalyzer: Malware semantic analysis using integrated CNN and bi-directional LSTM for detecting VM-level attacks in cloud. In 2019 Twelfth International Conference on Contemporary Computing (IC3), Noida, India, pp. 1-6. https://doi.org/10.1109/IC3.2019.8844877

[32] Nahmias, D., Cohen, A., Nissim, N., Elovici, Y. (2019). Trustsign: Trusted malware signature generation in private clouds using deep feature transfer learning. In 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, pp. 1-8. https://doi.org/10.1109/IJCNN.2019.8851841

[33] Abdelsalam, M., Krishnan, R., Huang, Y., Sandhu, R. (2018). Malware detection in cloud infrastructures using convolutional neural networks. In 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), San Francisco, CA, USA, pp. 162-169. https://doi.org/10.1109/CLOUD.2018.00028

[34] Jeon, J., Park, J.H., Jeong, Y.S. (2020). Dynamic analysis for IoT malware detection with convolution neural network model. IEEE Access, 8: 96899-96911. https://doi.org/10.1109/ACCESS.2020.2995887

[35] Chai, Y., Qiu, J., Su, S., Zhu, C., Yin, L., Tian, Z. (2020). LGMal: A joint framework based on local and global features for malware detection. In 2020 International Wireless Communications and Mobile Computing (IWCMC), Limassol, Cyprus, pp. 463-468. https://doi.org/10.1109/IWCMC48107.2020.9148289

[36] Kotian, P., Sonkusare, R. (2021). Detection of malware in cloud environment using deep neural network. In 2021 6th International Conference for Convergence in Technology (I2CT), Maharashtra, India, pp. 1-5. https://doi.org/10.1109/I2CT51068.2021.9417901

[37] Samuel, J.K., Jacob, M.T., Roy, M., Sayoojya, P.M., Joy, A.R. (2023). Intelligent malware detection system based on behavior analysis in cloud computing environment. In 2023 International Conference on Circuit Power and Computing Technologies (ICCPCT), Kollam, India, pp. 109-113. https://doi.org/10.1109/ICCPCT58313.2023.10245065

[38] Mustafa, H.M., Al-Zyod, M.H. (2024). Cloud computing malware detection using feature selection based on optimized White Shark Algorithm (WSO). In 2024 2nd International Conference on Cyber Resilience (ICCR), Dubai, United Arab Emirates, pp. 1-6. https://doi.org/10.1109/ICCR61006.2024.10533155