Periodicals of Engineering and Natural Sciences
Vol. 11, No. 3, May 2023, pp.176-192

ISSN 2303-4521
Original Research

Enhancing cloud security through the integration of deep learning and data mining techniques: A comprehensive review

Israa ezzat salem *,[1], Karim Hashim Al-Saedi [1]
[1]Computer Science Department, College of Science, Mustansiriyah University, Baghdad, Iraq

ABSTRACT

Cloud computing is crucial in all areas of data storage and online service delivery. It adds various benefits to the conventional storage and sharing system, such as simple access, on-demand storage, scalability, and cost savings. The employment of its rapidly expanding technologies may give several benefits in protecting the Internet of Things (IoT) and physical cyber systems (CPS) from various cyber threats, with IoT and CPS providing facilities for people in their everyday lives. Because malware (malware) is on the rise and there is no well-known strategy for malware detection, leveraging the cloud environment to identify malware might be a viable way forward. To avoid detection, a new kind of malware employs complex jamming and packing methods. Because of this, it is very hard to identify sophisticated malware using typical detection methods. The article presents a detailed assessment of cloud-based malware detection technologies, as well as insight into understanding the cloud's use in protecting the Internet of Things and critical infrastructure from intrusions. This study examines the benefits and drawbacks of cloud environments in malware detection, as well as presents a methodology for detecting cloud-based malware using deep learning and data extraction and highlights new research on the issues of propagating existing malware. Finally, similarities and variations across detection approaches will be exposed, as well as detection technique flaws. The findings of this work may be utilized to highlight the current issue being tackled in malware research in the future.

Keywords: Cloud computing, Malware Worms Detection, deep learning, data mining.

Corresponding Author:

Israa ezzat salem
Computer Science Department,
College of Science, Mustansiriyah University, Baghdad, Iraq.
Email: farg.israa@uomustansiriyah.edu.iq; israa.ezzat@bagdadacollege.edu.iq

1. Introduction

The term "cloud computing" refers to a type of Internet-based computing that provides a shared set of resources, including memory, processing power, network bandwidth, and user applications. With less maintenance and lower infrastructure costs, these resources could be made instantly and on-demand available to end users online [1-3]. Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) are the three categories of cloud computing services [4]. Additionally, it could be made available as a hybrid, private, public, or shared cloud. The fact that so few businesses are currently prepared to fully adopt cloud technology is one of the biggest problems it currently faces. DDoS is a type of violent attack that causes serious problems for cloud servers. Nearly every member of the community now regularly uses the Internet. This is because doing anything without the Internet is practically impossible, including maintaining social connections, conducting online banking transactions, managing one's health, and marketing. As the Internet has gained popularity, criminals have begun to commit crimes online rather than in the real world. Criminals frequently start cyberattacks on the devices of their victims by using malware. Any application that performs malicious loading on victim devices (computers, smartphones, computer networks, etc.) is referred to as malicious. Malware can take many different shapes, such as viruses, worms, trojan horses, hidden roots, and ransomware. Each type of malware and its family is intended to have an impact on the device of the original victim in a variety of ways, including causing harm to the target system, enabling the installation of remote code, stealing personal information, etc. It is becoming more challenging to classify malware because Ali Kashif Bashir was one of the assistant editors who coordinated the review of this article and approved it for publication. Malware instances may display characteristics from several categories at once. [5,] a malware attack that targets the victim in a coordinated manner and affects multiple machines is referred to as a malware attack. Deep learning techniques are used in this study to identify malware threats. Deep learning techniques have historically had trouble identifying assaults. Anomaly detection techniques study network traffic from a baseline profile and identify anomalies as deviations from the baseline profile, in contrast to signature detection methods that identify attacks based on previously acquired attack signatures. Approaches to signature detection are effective against well-known threats, but anomaly detection may spot unforeseen and unheard-of attacks (day zero). It's an unusual situation that the information from the attack is flowing so freely. Because of this, malware attacks become simple to launch, challenging to find and track, etc. [6]. Malware attacks are now one of the most significant threats to network security, as well. Then, to describe our system and gain a better understanding of machine learning and how it develops, we provide deep learning. Computer systems that can learn on their own from data are known as "deep learners" in the field of artificial intelligence. For the most part, deep learning has a significant impact on those industries' employment. Deep learning advances through several significant turning points [7-9] to arrive at this position. We thoroughly examine feature extraction as well as classification and assembly techniques. In addition, we look at additional issues, data mining-based malware detection challenges, and malware development trends.

2. Background theory

The background theory section provides an overview of key concepts and techniques related to data security and intrusion detection using deep learning and data mining. It covers topics such as machine learning algorithms, neural networks, feature extraction, anomaly detection, and classification methods. The section also discusses the importance of data preprocessing and feature selection in improving the accuracy of intrusion detection systems. Overall, this section serves as a foundation for understanding the technical aspects of data security and intrusion detection using advanced machine learning techniques.

2.1 Trends in malware creation and hiding technologies

Malware is defined as non-trustworthy objects that carry out malicious actions. Viruses, worms, back doors, rootkits, and ransomware are just a few of the many types of malware that can be categorized. Malware categories, core characteristics, and well-known malware families Malware is used by hackers to launch Internet-related attacks that take advantage of flaws, vulnerabilities, and failures in current systems, such as buffer overcapacity, insufficient security settings, and flaws in computer network protocols. Since many malware instances exhibit multiple class attributes simultaneously, malware categorization has grown more difficult over time [11]. The virus emerged in the wild for the first time ever. Throughout time, new varieties of malware will appear in computer systems. In the beginning, the malware was created with simple objectives in mind, like breaking into friends' computers or earning some quick cash. However, it was eventually replaced with sophisticated malicious software that damaged important industries, businesses, and government holdings. Classic malware and next-generation malware are the two main categories of malware. Next-generation malware is more dangerous and more difficult to detect and remove from computer systems than traditional malware, which is a common form of malware that is easy to identify and remove from computer systems. In addition, the next generation of malware might just find a way to get around the kernel-mode security program and conceal itself inside of computer systems. The newest malware can be used to launch ongoing and focused cyberattacks. Various malware types are used during attacks. The latest malware frequently uses traditional jamming techniques to avoid detection. Common opacity approaches are displayed and explained in the figure (1). Identifying next-generation malware with a single discovery technique is very challenging. As a result, there is a pressing need to identify using new methodologies and more computing power [12]. Malware analysis is divided into three categories, i. e. hard analysis, dynamic analysis, and hybrid analysis. [13].

Static analysis involves reviewing executable code without executing the original file, using techniques such as operational code frequency distribution, string signature, byte sequence, control flow graph, and n-gram. Assembly language and operating system knowledge are necessary for hard analysis. Dynamic analysis involves executing the malicious code and tracking its activity to analyze how it impacts the host device. This is also known as behavioral analysis and is useful for finding unknown malware. Sandboxes, simulators, virtual machines, and other tools are used for infected code analysis. Hybrid analysis combines both static and dynamic analysis techniques to provide better results but must adhere to the limitations of both methods.

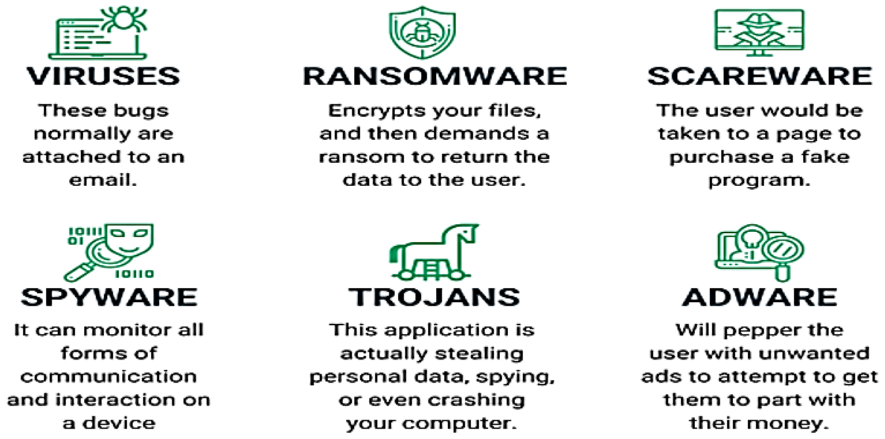

VIRUSES
These bugs normally are attached to an email.
RANSOMWARE
Encrypts your files, and then demands a ransom to return the data to the user.
SCAREWARE
The user would be taken to a page to purchase a fake program.
SPYWARE
It can monitor all forms of communication and interaction on a device
TROJANS
This application is actually stealing personal data, spying, or even crashing your computer.
ADWARE
Will pepper the user with unwanted ads to attempt to get them to part with their money.

Figure 1. Type of malware detection

2.2 Secure cloud for malware worms detection

Rapidly emerging as a new paradigm for gaining access to a variety of services, including storage, computation, data management, communications, media services, artificial intelligence, machine learning, developer tools, and security. The various cloud publishing models, services, and users are shown in Figure 3. Cloud computing services allow for the anytime, anywhere, and on any device access to data. On the other hand, because it allows easy access to viruses, this access option could be extremely dangerous. Cloud malware may result in several serious security issues, including identity theft, system damage, virtual device hijacking, loss of confidential data, and login credentials, to name a few.

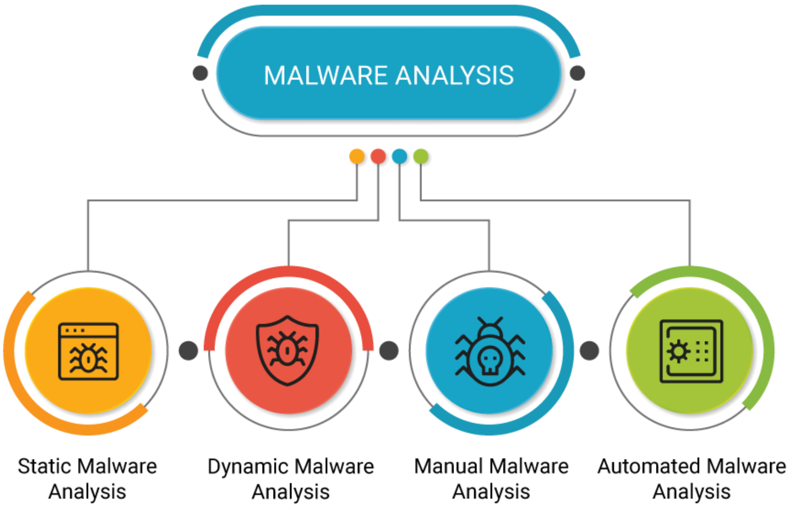


Figure 2. Malware Analysis Categorize

Malware may operate inside virtual devices and steal the user's personal information, according to Intercloud. To identify malware in the cloud, signature, behavioral, and machine learning-based techniques have been suggested. Amazon Web Services (AWS), Azure, and Google [14] are among the leading cloud service providers (CSPs) that provide cloud-based malware detection protection services. Google has unveiled a new tool to identify current threats, and Amazon Guard Duty is a threat detection service that monitors illegal malware detection activities.

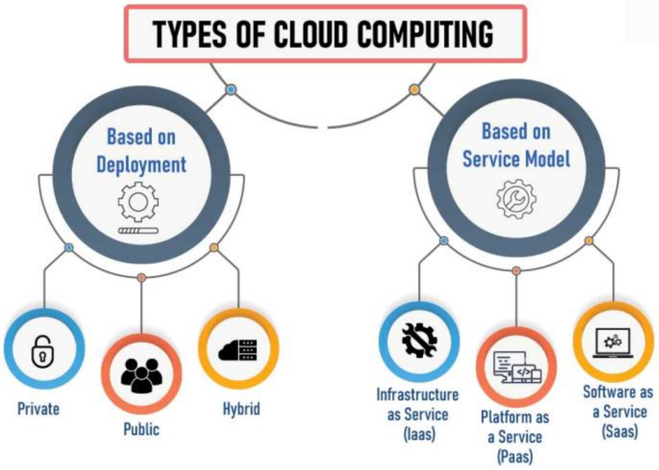

TYPES OF CLOUD COMPUTING
Based on Deployment
Based on Service Model
Private
Public
Hybrid
Infrastructure as Service (Iaas)
Platform as a Service (Paas)
Software as a Service (Saas)

Figure 3.Technologies and deployment methods for cloud computing

Figure 4 shows the malware detection process at the cloud level. Many people get data through email, HTTP, media, instant messaging, P2P, and other methods, including personal computers, mobile phones, and the Internet of Things. These users save their files to the cloud and then obtain a file report. Signature-based detection systems in the cloud spot malware by comparing patterns and these patterns are saved in the cloud. The signature-based method is extremely quick and accurate for detecting known malware, but it is unable to detect new malware. Malware is identified using anomaly-based detection techniques based on its behavior. Although this technique finds new harmful software, it also raises false alarms. Malware detection algorithms based on machine learning have been studied in the past, and these approaches have demonstrated promising performance and efficiency. Resolution tree, support carrier machines, LSTM[15], and other algorithms are used in machine learning-based malware detection approaches. This method is only successful if there is enough data and account strength to train the model. Machine learning-based malware detection technologies, on the other hand, have a scaling issue.

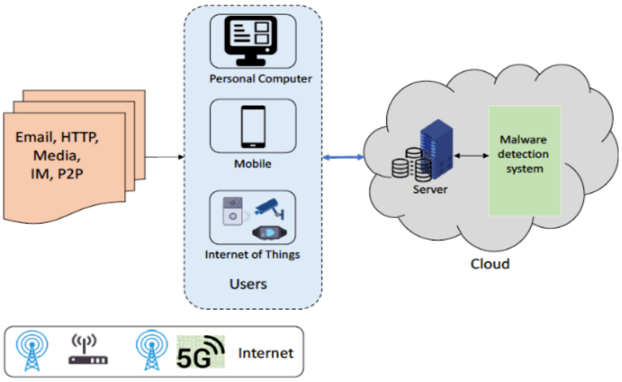


Figure 4. Cloud-based malware detection system

2.3 Deep learning

Because of its capability to handle enormous volumes of data, deep learning has recently proven to be a technology that is particularly useful. In particular, hidden layers have aroused interest in pattern recognition beyond conventional methods. Bypass neural networks are among the most widely used deep neural networks. [16]. Since the 1950s, when artificial intelligence was still in its infancy, researchers have worked to develop a system that can understand visual input. In the years that followed, this area was given the name computer vision. Computer vision advanced dramatically after a team of researchers from the University of Toronto created an AI model that, in 2012, significantly outperformed the leading image recognition algorithms. With an astounding 85 percent accuracy, the AI system, dubbed AlexNet (after its creator, Alex Krizhevsky), won the 2012 ImageNet Computer Vision Competition. The test result for the runner-up was a respectable 74 percent. The brain of AlexNet was a neural network called a neural bypass network, which is a type of neural network that closely resembles human vision. Since CNNs are now a standard component of many computer vision applications, they are covered in every online course on computer vision. Let's investigate the workings of CNNs now. Artificial neural networks (ANN) are methods inspired by the structure of the human brain that learn from a large amount of data in deep learning, a subset of machine learning. When robots can execute activities that would normally need human intellect, this is referred to as artificial intelligence. Machine learning, for example, enables computers to learn from their experiences and build abilities without the need for human interaction. A deep learning algorithm will do a job often, each time it is slightly tweaked to better the output, similar to how you learn from experience. Because neural networks include numerous layers (deep), they are referred regarded as "deep learning." Any issue that involves "thinking" may be learned to solve through deep learning. Deep learning enables computers to tackle complicated issues even when they are given a large, unstructured, and linked data set to work with. The better the learning algorithms do, the higher they are as seen in Figure 5.

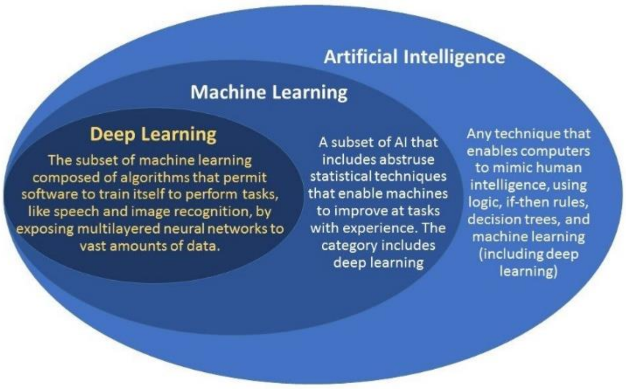

Artificial Intelligence
Machine Learning
Deep Learning
The subset of machine learning composed of algorithms that permit software to train itself to perform tasks, like speech and image recognition, by exposing multilayered neural networks to vast amounts of data.
A subset of AI that includes abstruse statistical techniques that enable machines to improve at tasks with experience. The category includes deep learning
Any technique that enables computers to mimic human intelligence, using logic, if-then rules, decision trees, and machine learning (including deep learning)

Figure 5. AI self-learning

Deep Learning uses four basic algorithms to predict future events and discover patterns based on individual user data [19]. Supervised learning uses previous instances and fresh data sets to predict outcomes, starting with supplied inputs and outputs. Unsupervised learning analyzes data to detect patterns and generate conclusions or predictions without labels or data categories. Semi-supervised learning mixes unlabeled data with human-based training, while reinforcement learning specifies a job or objective for the system to accomplish and receives input from reinforcement signals to learn the required behaviors. The deep learning model aims to provide strong answers for issues involving relationships, classifications, clustering, and prediction. Deep learning models are increasingly being used in e-learning applications. Some of the commonly used models include Convolutional Neural Network (CNN), which is used for evaluating high-dimensional pictures by transforming 2D to 3D using convolutional filters. Recurrent Neural Network (RNN) is a structure that can learn typical sequences and time dependencies as well as connections between hidden states, making it ideal for health problems that require modeling changes in clinical data over time. Deep Belief Network (DBN) has one-way communication on two levels above the layers, with each subnet's hidden levels serving as a visible layer for the following tier. Finally, Deep Neural Network (DNN) contains more than two layers, allowing for a complex nonlinear relationship.

2.3 Deep learning in malware detection

The use of deep learning frameworks in cybersecurity is developing, and research in this area is gaining traction. However, in the dynamic context of industry 4.0, high-quality research outputs in this field are impeded by a shortage of categorized data and standard data sets, and DL models may be highly complicated. The primary categories and hierarchy of subcategories used to characterize DL approaches used in cybersecurity are shown in Figure 6. We'll go through how to identify risks in cybersecurity in this part. It is described how to choose from a variety of taking approaches (malware detection) that are divided into different categories. Deep learning methods: types and how to cope with them [22].

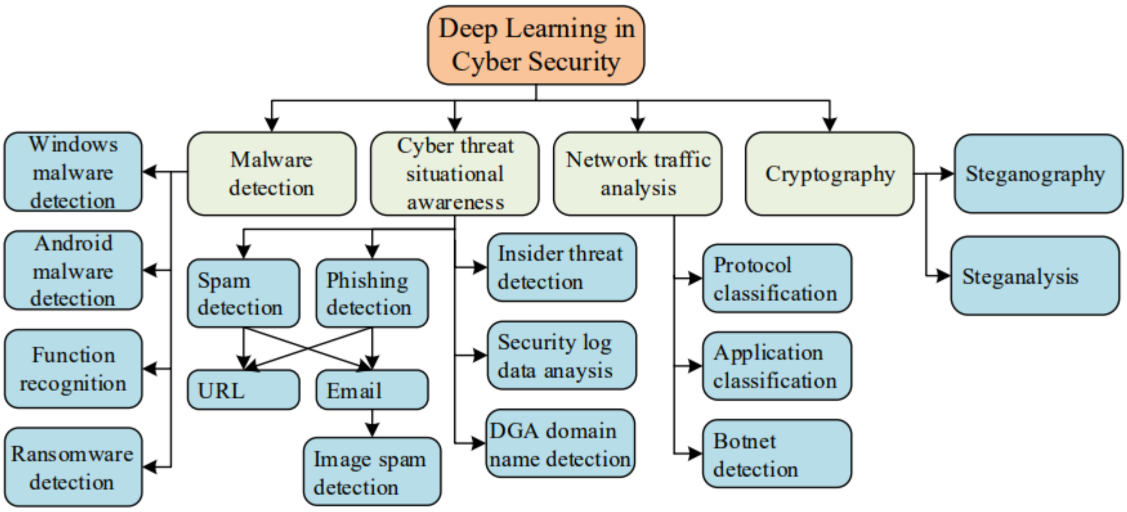


Figure 6. DL Methods in Cyber Security: A Hierarchical Classification

2.4 Deep learning in windows malware analysis

Numerous types of malware, including viruses, trojans, worms, back doors, hidden roots, spyware, ransomware, and panic software, among others, have made Windows-based applications into platforms for cybercrime, espionage, and other illegal activities. To address this, DL designs have recently been used to analyze malware on Windows. Before the decryption of the recovered data and the victim's return, ransomware sets up shop on the victim's computer and transmits an encrypted demand for payment. [23].

A. **Deep Neural Network (DNN):** A multitasking DL framework with 2 million test files and 4 point 5 million training data was proposed to detect binary malware. Leakage also significantly decreased the error rate for both deep and shallow neural frames, and the model's training time was reduced by using the corrected linear activation function. With 97 percent accuracy and a ROC of 0 point 99, the DL model was trained with BD and validated with a challenging data set. On the common EMBER data set, the DNN malware classification structure was contrasted with other shallow models, such as the LR, NB, KNN, DT, AB, and SVM. [24]. The method's primary limitation is that the proposed DNN structure depends on the feature geometry.

B. **Convolutional Neural Network (CNN):** The information contained in PE files was used by FFN bypass to identify and classify malware using the pyramidal feature extraction technique. An AUC of 0.9973 was obtained for malware detection using CNN, which was used to classify and train a DL method using a different kernel and data volume. The infection was represented as an image in CNN. The DL policy suggests classifying malware using two sets of standard data: Malimg and Microsoft malware. The accuracy of gray-gradient image characteristics fed into CNN was 98.5% and 99.7% with the Malimg and Microsoft data sets, respectively. [25]. The ability to detect malware using GoogleNet and ResNet models was also tested using Microsoft's data collection.

C. **Recurrent Structures (RS):** Half-frame models and Max Pooling were used in the projection phase, resulting in 98 point three percent TPR and 0 point one percent FPR. Unsupervised data was used to train a hybrid model made up of Echo and RNN status networks. Additionally, the LSTM and GRU models used maximum time assembly, attention mechanisms, and semi-supervised learning for ransomware detection [26].

D. **Autoencoder (AE) and Deep belief network (DBN):** The malware signature was created and categorized using DBN, and the model used a deep collection of DAEs for a compressed representation of harmful behaviour to achieve a resolution of 98.6 percent. Utilize DL window models to efficiently detect malware while learning an unsupervised feature of four Windows API calls and adjusting them as necessary with the aid of supervised parameters. In a different study, the malware was encoded as opcode sequences and sent to DBNs for classification [27].

E. **Mixed DL architectures:** The DNN model was proposed in two steps to identify malware based on operation behaviour to ascertain whether the terminal was infected. RNN was employed to extract features as images entered CNN for classification, and LSTM was applied to produce features from API call sequences that describe process activity. In a separate study, the NN hybrid was created with two layers of circumvention to extract hierarchical features that include both complete serial modelling and circumvention of n-gram features to classify malware, outperforming other ML approaches like SVMs and hidden Markov models. [28] Malet, an artificial learning model for features that uses CNN and LSTM to identify malware from raw data for 40,000 samples converted to grey pictures, has been suggested with a malware classification accuracy of 99.88 percent. [29] has presented a heterogeneous DL structure made up of AE and Boltzmann multilayered and memory layers related to windows API calls, pre-training, and precise malware detection adjustment.

F. **Hybrid Recurrent Structures (HRS):** The DL window with CNN and BLSTM employed a data-driven technique to classify nine distinct varieties of malware using sophisticated attributes. The system malware model was suggested to produce lower FPR than existing macrocalcification models by using API call characteristics [30].

2.5 Deep learning in android malware detection

Malicious attacks are possible on Android (OS), an open-source operating system with many essential financial and personal apps. To earn a financial advantage, hackers employ malware to steal private sensitive data or delete/change existing data. Many third-party shops offer Android applications, allowing users to unintentionally repackage Android apps with dangerous code. During the installation step, Android provides each app a unique Linux user ID to ensure that each program runs its virtual machine instance. This makes creating a protective mode that separates programs from one another much simpler. Uses Android permissions to provide a delegate mechanism. Android features are gathered from deep-rooted and unseated devices and fed into machine learning models to learn how to discriminate between benign and hazardous apps. Malicious programs concealed from the certified market store that hosts apps like Google Play and permission mechanisms, on the other hand, might deceive a mobile user into granting installation rights. The impact is less commonly recognized to the end-user if naive people take a blind approach to provide rights when installing programs. As a result, Android-based authorization systems must be subjected to risk analysis. As signature and inference-based approaches entirely fail to identify malware on day zero, attacks on the operating system of smart devices such as Android will continue to rise as technology improves. Android malware detection strategies are being investigated more and more. Self-learning systems based on DM, ML, and DL algorithms may give novel sensor capabilities for detecting Android malware that can be scaled up. Furthermore, these approaches may identify pre-existing malware variables as well as wholly new malware. Researchers use two types of methodologies to extract characteristics from Android that are quite comparable to those found in PC environments: static analysis and dynamic analysis.[31] While static analysis examines the operating time execution behavior of applications such as system calls, network connections, memory usage, power consumption, and user interactions, the dynamic analysis examines the operating time execution behavior of applications by unloading or dismantling them without running time. The hybrid analysis is a two-step procedure in which static analysis is conducted first, followed by dynamic analysis, resulting in cheaper computing costs, lower resource use, lighter weight, and shorter processing times. Because it gives greater detection rates, antivirus companies for smartphones are increasingly using the hybrid analytic technique. In Android malware analysis, a summary of the examination is given on DL apps.

The article discusses various frameworks and strategies for detecting Android malware. The first framework, which uses dynamic graphical analysis of system calls, outperforms conventional detectors with a 98.86 percent accuracy rate. The second framework, which combines static and dynamic analysis, gains over 200 features, and achieves a 96 percent accuracy rate, outperforming other models including NB, DT, SVM, MLP, and LR. A new DNN Android malware workbook improves detection accuracy to 94.7 percent by using a multi-core hierarchy that learns from a built-in collection of characteristics. LSTM outperformed RNN in identifying Android malware with a detection accuracy of 93.9 percent and 97.5 percent respectively utilizing dynamic and consistent analysis. Features for the CNN-based Android malware definition framework include API call sequences and protection levels. Overall, consistent analysis has resulted in greater accuracy compared to dynamic analysis for online malware detection.

2.6 Sources of data mining

In this malware detection study, data mining techniques are used. This investigation primarily focuses on the source data used in studies to identify both new and old malware. The PE file, assembly file, and core are the primary sources of data. There are three different ways to extract data from the PE file, so [34]. The PE file is a useful tool for extracting the API call sequence and byte sequences. It consists of two parts, the head and section, with various characteristics that can be utilized as data for analysis. The assembly file contains the operational code, and the frequency of opcode sequences can be extracted as data. The operating system's core is crucial but often targeted by malicious symbols, leading to detectable modifications in characteristics retrieved from it if contaminated. Overall, the PE file provides valuable information for analyzing and detecting potential threats.

API calls are pre-programmed functions that can be used to create new programs for a specific operating system. Byte sequences, obtained from files, are used to collect data and assess frequency using bytecode. Features such as DLL, API function calls, repetitive code instructions, API call sequence, text-based search technology, hash, attribute certificate, date/time seal, file indication, link information, CPU type and PE logical structure can be extracted from a linear data flow of headers and sections that make up a PE file. Opcode sequence frequency is examined by listing the play code in each assembly file. In Sl, features are gathered from the kernel using various data types. To utilize the API, one must plot the byte sequence of the PE file and make observations on the PE head and sections. Additionally, it is necessary to extract the frequency sequence opcode from the instructions in the assembly file. Ishita Basu et al. have retrieved kernel characteristics and terminology system call such as files, records, processes, threads, networks, and memory sections. These functions of the nucleus are often perceived as characteristics.

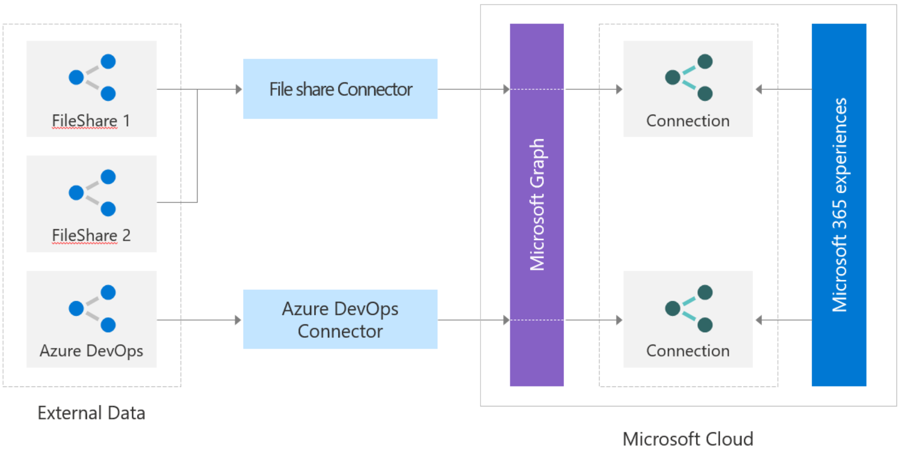


Figure 7. API Graph Construction

2. Related work

Cloud computing technology has revolutionized the way businesses operate, and its impact can be seen across various sectors. One of the most recent applications of cloud computing technology is in malware detection. With the increasing number of cyber threats, it has become imperative for organizations to adopt advanced security measures to protect their sensitive data. Cloud-based detection and cloud-based approaches have emerged as a reliable solution for detecting and preventing malware attacks. Several academic investigations have been conducted on this topic, highlighting the benefits of cloud-based detection methods over other traditional methods. Cloud-based approaches offer scalability, flexibility, and cost-effectiveness, making them an ideal choice for organizations of all sizes. For deep learning and data mining, cloud-based detection approaches are evaluated based on various factors such as the basic concept, feature extraction method, and algorithm types. The use of cloud computing technology in malware detection is expected to grow in the coming years as more organizations realize its potential in enhancing their cybersecurity posture.

3.1 Secure cloud for malware worms detection using deep learning

Gao, Xianwei et al [35], developed a novel malware detection model based on cloud semi-supervised transport (SSTL), which comprises detection, forecasting, and transportation components, and is based on learning. A byte work based on a recurrent neural network (RNN) is built for its detection component to identify malware to safeguard the privacy of tenants in the public cloud. However, because of the shortage of training examples, the byte class's accuracy after supervised learning is only 94.72 percent. The ASM workbook provides a forecast component with a 99.69 percent resolution. The transit component requires a prediction component to classify an unidentified data set by merging predicted labels with bytes attributes from an unclassified data set in a new training data set. The fresh data set is transmitted to the byte workbook for training again, thanks to the benefits of semi-supervised learning. Learning about semi-supervised transport boosted detection component performance from 94.72 percent to 96.9 percent in tests using Kaggle malware data sets.

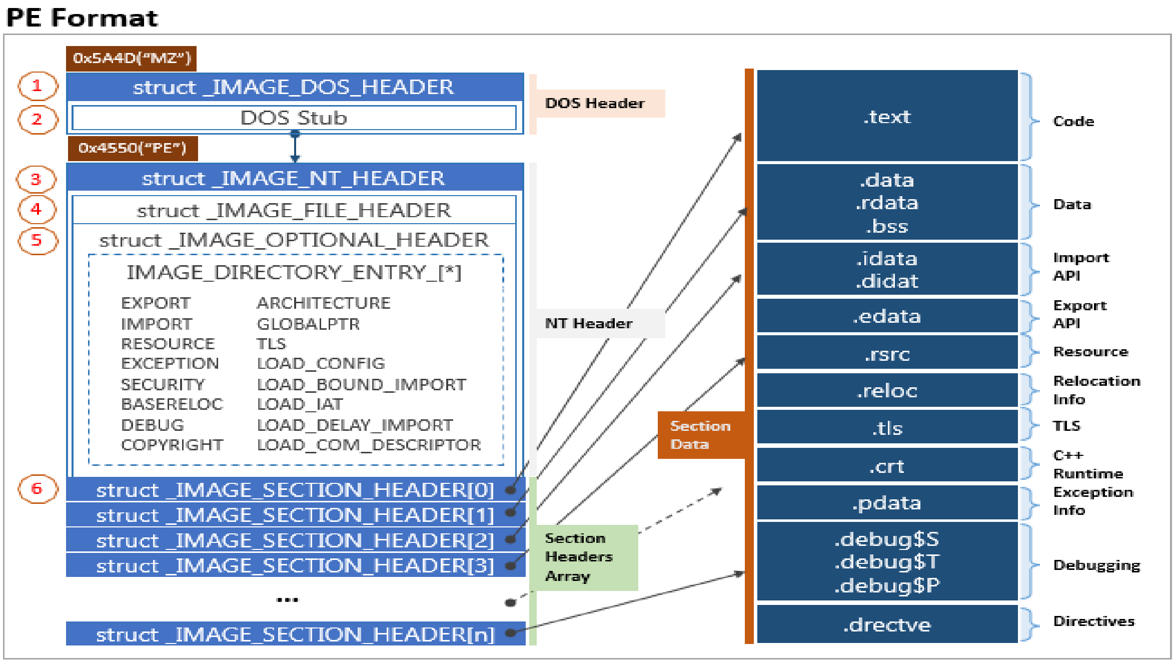


Figure 8. PE File Format

The Deep Wrap Neural Network (DLCNN) is recommended by Venkata Rao, M., et al. [36] for a combined disclosure of both signature-based internet worms and NetFlow in connection to various attacks and may also prevent suspicious attacker behaviors (cybercriminals). Additionally, it protects user data, employs defenses, and halts the spread of internet worms. The effectiveness of the proposed DLCNN model is assessed using the package capture (PCAP) and KDD-CUP-99 data sets. The suggested DLCNN model is then compared to current automated learning to see if it outperforms it, and neural network models are once again used to make this determination.

Hasan Alkahtani et al. [37], "Use machine learning and deep learning approaches to identify harmful attacks targeted at Android.". Support vector machines (SVM), closest neighbors (KNN), linear discriminatory analysis (LDA), long-term memory (LSTM), long-term short-term memory of the neural bypass network (CNN-LSTM), and automated encryption techniques have all been applied to the detection of malware in mobile contexts. To assess the cybersecurity system, two sets of typical Android mobile data were used. The link is determined to locate a high proportion of critical characteristics for these systems in terms of attack defense. On Android applications, machine learning and deep learning algorithms have effectively discovered malware. Using the CICAndMal2017 data set, the SVM method has the greatest accuracy (100%). Using the Drebin data set, the LSTM model similarly obtained excellent percentage accuracy (99.40 percent).

Z. Zahraa. We used a deep bypass neural network (CNN) based on the Xception model to identify malware images. Edie et al. [38] developed a framework for classifying and detecting malware through learning transfer, which is based on existing deep learning models that have previously been trained on enormous picture data sets. Malware samples are represented as gray images of a bath lot in the newly developed CNN special structure known as the Xception model, which is more effective with fewer problems than current common CNN models like VGG16. A deep neural network is trained to freeze the bypass layers of the Xception model that adapts to the last layer of malware family classification, and experimental results on the mailing data set of 9,821 samples from 26 different families show that this method is effective.

Hooman Alavizadeh et al [39] To identify network infiltration, a new generation of network penetration detection approach combines Q-learning-based improved learning with a deep-forward neural network method. Our suggested Q-Learning (DQL) learning model offers continuous automated learning capacity for a network environment, allowing it to identify various forms of network penetrations and improve its continuous detection skills utilizing an automatic pilot error technique. In the DQL model of the most successful self-learning, we outline how to change the many hyperbolic parameters. We confirm that the smaller discount factor, which is denoted as 0.001 under 250 episodes of training, delivers the highest performance outcomes, according to our thorough testing findings based on the NSL-KDD data set. Our experimental findings also reveal that our suggested DQL beats other comparable machine learning algorithms in identifying distinct intrusion types.

3.2 Secure cloud for malware worms detection using data mining

The study [40] Propose an enigmatic neural hybrid (EHNFC) for putting together Android-based malware with consent-based components. The suggested EHNFC can not only identify mystery malware using thin principles, but it can also evolve its structure by adopting new malware identification principles to improve the accuracy of its detection when utilized as a part of a site with additional malware apps. To this purpose, a complex assembly strategy has been modified to adapt and enhance the malware site's gentle principles to promote a varied methodology for radius repair and bulk-based component concentration. This change to the advanced assembly technique increases group integration, results in better-personalized judgments for input data, and hence improves the proposed EHNFC's characterization accuracy. The suggested EHNFC exploratory findings reveal that the concept revolves around some sophisticated mixed malware arrangement techniques, as well as a false positive rate (0.05) and a mistaken negative rate (0.05). (0.05). The findings also suggest that the proposal outperforms existing strange neural frameworks in detecting Android malware (90 percent). The study [41] To connect elements of static and dynamic analyses of Android apps, provide a deep learning method. Additionally, they turned on DroidDetector, an Android malware detection engine based on the Deep Learning Method that can determine whether a file exhibits malicious behavior. With many Android apps, they evaluated DroidDetector and carried out a thorough analysis of the components that deep learning primarily uses to completely film malware. With access to more setup details, deep learning appears to be particularly compelling for describing Android malware. In comparison to traditional machine learning techniques, DroidDetector can detect objects with an accuracy of 96.76 percent.

The study [42] find malware, advise using an affiliation mining strategy based on API invitations. To increase the speed of OOA's mining identification, new methodologies are introduced. To improve governing quality, criteria are suggested to identify the API and to exclude APIs that cannot visit objects in a distinctive way. Experiments demonstrate that the proposed systems can significantly speed up OOA operation. In our studies, the cost of time for information extraction is cut by 32%, and the cost of time for arrangement is cut by 50%. The study [43] A brand-new hybrid methodology called HDM-Analyzer is presented that incorporates points of interest into dynamic and reliable high-frequency investigation techniques while maintaining accuracy at a tolerable level. In this way, they have no overhead performance. HDM-Analyzer can forecast the core leadership's predominance based on the use of empirical data gathered from research elements. The primary responsibility of this paper is to adopt the preferred viewpoint precisely when investigating and standardizing the component in a consistent examination, keeping in mind the goal of increasing the accuracy of the established investigation. The overheads of the execution were, in fact, incurred during the learning phase; as a result, they have no impact on the light extraction phase that is carried out during the examination process. The preliminary findings demonstrate that compared to static investigation strategies and components, HDM-Analyzer achieves better overall accuracy and multifaceted time quality.

The study [44] In light of the semantic evaluation of dynamic API sequences, provide a bi-layer behavior reflection technique. In distinct semantic layers, processes on sensitive frame assets and complicated practices are segregated in an interpretable way. Raw API calls are connected at the lower layer to extract low-layer practices by looking into the dependency on the information. Low-class practices are coupled with extensive interpretation in the upper class to construct more complicated high-class practices. Finally, distinct low-layer procedures are introduced into a high-dimensional vector space. Separate techniques may now be applied explicitly via several well-known machine learning accounts. Furthermore, to solve the issue of not adequately investigating thoughtful projects or significantly imbalanced malware and friendly projects, it is recommended to develop OC-SVM-Neg, a single-class boosting vector machine (OC-SVM) that benefits from negative and accessible instances. According to the erroneous rate of caution and speculative ability, the suggested extraction strategy using OC-SVM-Neg surpasses dual works, according to the trial.

The study [45] developed a graph-based model that identifies whether an unknown software sample is hazardous or benign, and categorizes a malicious program into one of a set of well-known malware family groupings, based on connections between system call pools. Customers more accurately utilized system call dependency charts (or, in short, ScD graphs), which were derived from impacts recorded during dynamic pollution research. The authors designed their model to withstand the drastic changes that occur when we apply our recognition and arrangement systems to a weighted coordinated graph, also known as a Gr-graph, which is a specific graph of a group's relationship that results from scd-graph after collecting separate subsets of its peaks. The authors offered a measure of comparability in Delta for the discovery method, and saMe-similitude and NP-similarity metrics that incorporate saMe-NP convergence for the classification technique. Finally, they assessed their model for detecting and classifying malware, revealing its effectiveness against malware based on detection rates and classification accuracy.

The study [46] The adaptive recognition component will employ the Multi-Feature Collaborative Resolution Integration, and Android-based malware will be used for testing (MCDF). The ultimate objective of setting up a group of workbooks and combining their choices using a group method considering the probability hypothesis is to provide better discoveries by building up a group of workbooks and combining their options using a group approach. The suggested model's implementation is tested using a variety of Android-based malware, including a variety of malware families, and the findings reveal that the technique delivers better execution of the best team plans in their class.

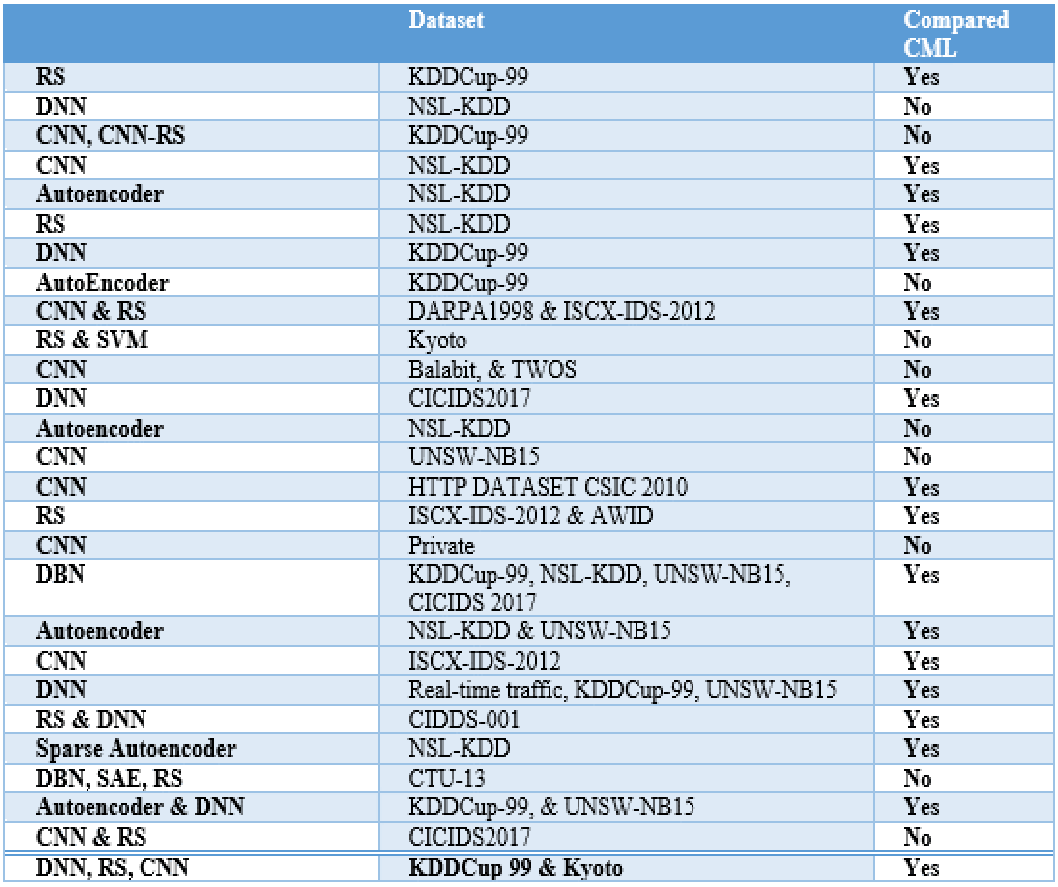
The study [47] Proposed many categorization algorithms with the purpose of identifying malware based on each harmful program's components and behavior. To discover malware traits, dynamic investigative technology has been created. A suggested program for changing the XML document for the Executive Archives of Malware Behavior to enter a suitable WEKA tool has been supplied. The authors use the WEKA tool to depict the efficiency of implementation, information preparation, and testing by applying the suggested methods to a real-world contextual investigation information set. The evaluation's findings described the applicability of the suggested data extraction method. Furthermore, the suggested data extraction approach is more effective at detecting malware, and behavioral malware categorization might be beneficial for detecting malware in behavioral antivirus software.

The study [48] Propose a behavior-based features model that detects the malware example's destructive conduct. The authors initially execute a dynamic assessment of a usually delayed malware data set inside a controlled virtual environment to record the consequences of API calls conjured up by malware samples to eliminate the suggested model. The impacts are subsequently disseminated via high-level characteristics that denote activities. Some well-known classification techniques, such as random forests, resolution trees, and SVM, are used to assess the suggested approach. The results of the experiments reveal that the work achieves high accuracy and good results when it comes to identifying malware variables.

4. Summary of the approaches that have been Examined

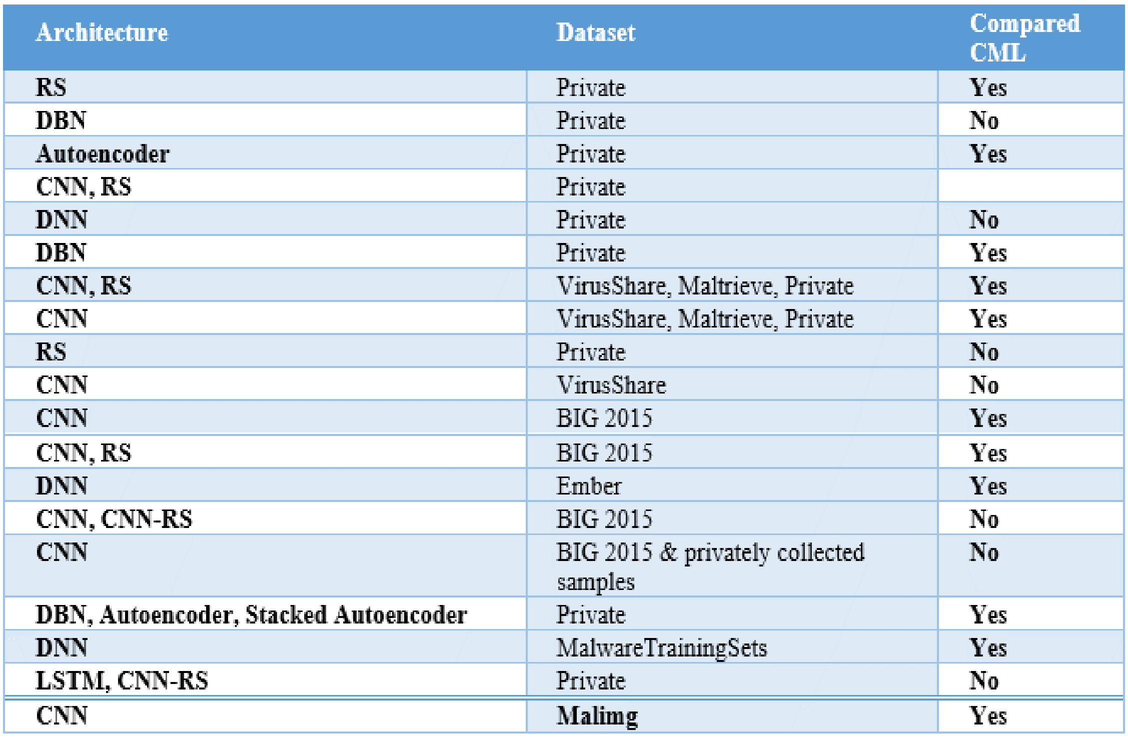
In recent years, there has been a growing interest in the use of deep learning architectures for intrusion detection. Several studies have explored the use of convolutional neural networks (CNNs), recurrent neural networks (RNNs), and deep belief networks (DBNs) for this purpose. These architectures have shown promising results in detecting various types of attacks, including DoS, probing, and malware attacks. Data mining techniques have also been extensively used for intrusion detection. These techniques involve the analysis of large datasets to identify patterns and anomalies that may indicate an attack. Various data mining algorithms such as decision trees, support vector machines (SVMs), and clustering algorithms have been employed for this purpose. The choice of dataset is crucial for the effectiveness of intrusion detection systems. Several datasets have been adopted for evaluating the performance of different approaches. The most used datasets include KDD Cup 99, NSL-KDD, and UNSW-NB15. These datasets contain a wide range of attack scenarios and are widely used by researchers to evaluate the performance of their models. Overall, these approaches have shown promising results in detecting various types of attacks. However, there is still room for improvement in terms of accuracy and efficiency. Further research is needed to develop more robust models that can effectively detect new and emerging threats in real-time environments.

Table 1. A Summary of studies on deep learning architectures for malware analysis



	Dataset	Compared CML
RS	KDDCup-99	Yes
DNN	NSL-KDD	No
CNN, CNN-RS	KDDCup-99	No
CNN	NSL-KDD	Yes
Autoencoder	NSL-KDD	Yes
RS	NSL-KDD	Yes
DNN	KDDCup-99	Yes
AutoEncoder	KDDCup-99	No
CNN & RS	DARPA1998 & ISCX-IDS-2012	Yes
RS & SVM	Kyoto	No
CNN	Balabit, & TWOS	No
DNN	CICIDS2017	Yes
Autoencoder	NSL-KDD	No
CNN	UNSW-NB15	No
CNN	HTTP DATASET CSIC 2010	Yes
RS	ISCX-IDS-2012 & AWID	Yes
CNN	Private	No
DBN	KDDCup-99, NSL-KDD, UNSW-NB15, CICIDS 2017	Yes
Autoencoder	NSL-KDD & UNSW-NB15	Yes
CNN	ISCX-IDS-2012	Yes
DNN	Real-time traffic, KDDCup-99, UNSW-NB15	Yes
RS & DNN	CIDDS-001	Yes
Sparse Autoencoder	NSL-KDD	Yes
DBN, SAE, RS	CTU-13	No
Autoencoder & DNN	KDDCup-99, & UNSW-NB15	Yes
CNN & RS	CICIDS2017	No
DNN, RS, CNN	**KDDCup 99 & Kyoto**	Yes

Table 2. A Summary of Studies on Deep learning Applications For Malware Analysis.



Architecture	Dataset	Compared CML
RS	Private	Yes
DBN	Private	No
Autoencoder	Private	Yes
CNN, RS	Private	
DNN	Private	No
DBN	Private	Yes
CNN, RS	VirusShare, Maltrieve, Private	Yes
CNN	VirusShare, Maltrieve, Private	Yes
RS	Private	No
CNN	VirusShare	No
CNN	BIG 2015	Yes
CNN, RS	BIG 2015	Yes
DNN	Ember	Yes
CNN, CNN-RS	BIG 2015	No
CNN	BIG 2015 & privately collected samples	No
DBN, Autoencoder, Stacked Autoencoder	Private	Yes
DNN	MalwareTrainingSets	Yes
LSTM, CNN-RS	Private	No
CNN	**Malimg**	Yes

Table 3. A Summary of Studies data mining malware For Malware Analysis

Case study	Classification approach	Data analysis method	dataset	Compared CML
Graph mining in malware detection	Graph search	Dynamic	Windows sandbox malware	Yes
Android malware detection	Random forest	Dynamic	Android applications	Yes
Android malware detection	Multilayer perceptron	Dynamic	Several websites	Yes
Android malware detection	Evolving neuro-fuzzy inference system	Dynamic	Google play and android Malware genome Project	Yes
AMAL: automated malware analysis	Decision trees	Dynamic	Random sample from internal user and external customers such as antivirus companies	Yes
Android malware characterization and detection	Deep belief networks	Hybrid	Google play and android Malware genome project	Yes
Deep Packet Inspection for malware	BoostedJ48, J48, Naïve Bayesian and SVM	Dynamic	Wireless and Secure Networks Research Lab	Yes
Objective Oriented malware	Multiple association rules	Hybrid	Several websites	Yes
Hybrid analysis malware	Bayesian network, Naive Bayes, Lazy K-Stare	Hybrid	Selected randomly from malware repository of APA, the security research laboratory at Shiraz University	Yes
Bilayer behavior abstraction	SMV, Naïve Bayes, decision tree, logistic regression	Dynamic	Open-access malware database such as VXHeaven website	Yes
Malware specifications	System call dependency graph	Dynamic	VXHeavens website	Yes
System-call malware	SaMe-NP	Dynamic	Variety of commodity software types including editors, office suites, media players,	Yes
Android based malware	J48, SVM, IBk, NaiveBayes	Static	Google play and android Malware services	Yes
Behavioral Malware	Regression, SVM, J48	Dynamic	Web data commons library in VirusSign and VXHeaven	Yes
Malicious code based on API	Decision tree, SVM and random forest	Dynamic	API hooking library in VirusSign	Yes
Feature extraction method in cloud	Decision tree, SVM, Boosting	Static	Random dataset of VirusTotal	Yes
Security dependency network for malware detection	No read down and no write up	Dynamic	VXHeavens website	Yes
Data flow android malware detection	KNN, LR, BN	Static	VXHeavens website and Google play	Yes
So-called compression-based malware detection	k-NN, QDA, LDA, SVN, Decision Trees, and random forest	Dynamic	Cuckoo sandbox	Yes
Deep-learning malware detection	Naive Bayes, PART, Logistic Regression, SVM and MLP	Hybrid	Google play, virus share	Yes

5. Conclusion

There is no way to identify every new generation and complex malware, even though several new approaches based on various malware detection methods have been presented. With well-known heuristic malware detection technologies, they work well. On the other hand, model checking, and cloud-based methods work well for complex and unknown malware behavior. To recognize specific malware types, both well-known and novel, data mining, Internet of Things (IoT), and deep learning techniques have been developed. But using these techniques, some malware can't be found. This demonstrates how challenging it is to create an effective malware detection system and how new methodologies and research are required to fill in the gaps. This research continues to be a useful tool for computer scientists and developers working in the field even though malware production and detection technologies are constantly changing. In the future, it should be suggested to use a novel technique and methodology. One of several methods to accomplish this is to combine malware detection technologies. For example, combining behavior-based and model-inspection-based techniques, as well as deep learning, data mining, and cloud computing, will almost surely result in a more effective discovery process.

Declaration of competing interest

The authors declare that they have no known financial or non-financial competing interests in any material discussed in this paper.

Funding information

No funding was received from any financial organization to conduct this research.

References

[1] S. Morgan, "Cybersecurity almanac: 100 facts, figures, predictions and statistics," Cybercrime Magazine Cisco and Cybersecurity Ventures, 2019. doi: 10.13140/RG.2.2.23577.67686.

[2] Y. Ye, T. Li, S. Zhu, W. Zhuang, E. Tas, U. Gupta and M. Abdulhayoglu, "Combining file content and file relations for cloud based malware detection," in Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 222-230, 2011. doi: 10.1145/2020408.2020439.

[3] Ö. Aslan, R.Samet and Ö.O.Tanrıöver, "Using a Subtractive Center Behavioral Model to Detect Malware," Security and Communication Networks 2020, pp.1-12, 2020.doi:10.1155/2020/8897014.

[4] H. R. Abdulshaheed, S. A. Binti, and I. I. Sadiq, "A Review on Smart Solutions Based-On Cloud Computing and Wireless Sensing," Int. J. Pure Appl. Math., vol. 119, no. 18, pp. 461–486, 2018.

[5] Constrained internet of things (IoT)devices,"Software:PracticeandExperience47(3),pp421-441,2017.doi:10 .1002/spe.v47 .3

[6] O. Kayode, D. Gupta, and A. S. Tosun, "Towards a distributed estimator in smart home environment," in 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), IEEE, pp. 1-6, 2020. doi: 10.1109/WF-IoT48130.2020.9220994.

[7] A. Singh and A. Jain, "Study of cyber-attacks on cyber-physical system," in Proceedings of 3rd International Conference on Internet of Things and Connected Technologies (ICIoTCT), pp. 26-27, 2018.

[8] Y. Ye, T. Li, S. Zhu, W. Zhuang, E. Tas, U. Gupta and M. Abdulhayoglu, "Combining file content and file relations for cloud based malware detection," in Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD), pp. 222-230, Aug., 2011.

[9] W. Hardy, L.Chen, S.Hou,Y.Ye,and X.Li,"DL4MD: A deep learning framework for intelligent malware detection," in Proceedings of the International Conference on Data Science (ICDATA), The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), p61.,2016.doi:10/1016/j.procs/2017/01/012

[10] T. Aldwairi, D. Perera, and M. A. Novotny, "An evaluation of the performance of Restricted Boltzmann Machines as a model for anomaly network intrusion detection," Computer Networks, vol. 144, pp. 111-119, 2018. doi: 10.1016/j.comnet.2018.08.012.

[11] Y. Cheng, X. Zhou, S. Wan, and K.-K.R Choo, "Deep Belief Network for Meteorological Time Series Prediction in the Internet of Things," IEEE Internet of Things Journal, vol. 14, no. 8, 2015.

[12] M. Chowdhury, A. Rahman, and R. Islam, "Malware analysis and detection using data mining and machine learning classification," in Proceedings of the International Conference on Applications and Techniques in Cyber Security and Intelligence (ATCI), Edizioni della Normale: Cham, Switzerland, Jun., pp. 266-274.

[13] M.Zekri,S.E.Kafhali,N.Aboutabit,andY.Saadi,"DDoSattackdetectionusingmachinelearningtechniquesincloudcomputingenvironments,"inProceedingsofthe2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech), IEEE: Oct., pp.1-7.doi:10 .1109/CloudTech

.2017 .8095622

[14] Y. Ye, L. Chen, S. Hou, W. Hardy, and X. Li, "DeepAM: a heterogeneous deep learning framework for intelligent malware detection," Knowledge and Information Systems, vol. 54, no. 2, pp. 265-285, 2018. doi: 10.1007/s10115-017-1125-5.

[15] J. Sun, R. Wyss, A. Steinecker, and P. Glocker, "Automated fault detection using deep belief networks for the quality inspection of electromotors," Technisches Messen, vol. 81, no. 5, pp. 255-263, 2014. doi: 10.1515/teme-2014-0009.

[16] S. Tsimenidis, T. Lagkas and K.Rantos,"Deep learning in IoT intrusion detection," Journal of Network and Systems Management , vol .30 , no .1 , pp .1 -40 ,2022 .doi:10 .1007/s10922-021-09638-w .

[17] M.Elsisi et al., "Effective IoT-based Deep Learning Platform for Online Fault Diagnosis of Power Transformers Against Cyberattack and Data Uncertainties," Measurement , vol .189 , no .110686 ,2022 .doi:10 .1016/j.measurement2021 .110686 .

[18] S. Q. Salih and A. R. A. Alsewari, “A new algorithm for normal and large-scale optimization problems: Nomadic People Optimizer,” Neural Comput. Appl., vol. 32, no. 14, pp. 10359–10386, 2020, doi: 10.1007/s00521-019-04575-1.

[19] S. Malik, A. K. Tyagi, and S. Mahajan, "Architecture, Generative Model, and Deep Reinforcement Learning for IoT Applications: Deep Learning Perspective," in Artificial Intelligence-based Internet of Things Systems, Springer, Cham, 2022, pp. 243-265. doi: 10.1007/978-3-030-93613-1_12.

[20] D. Kajaree and R. Behera, "A Survey on Healthcare Monitoring System Using Body Sensor Network," Int. J. Innov. Res. Comput. Commun. Eng., vol. 5, no. 2, pp. 1302–1309, 2017.

[21] F.C.C Garcia, C.M.C Creayla and E.Q.B Macabebe, "Development of an Intelligent System for Smart Home Energy Disaggregation Using Stacked Denoising Autoencoders," in International Symposium on Robotics and Intelligent Sensors (IRIS), IEEE Japan, 2016.

[22] T.J Saleem and M.A Chishti, "Deep Learning for Internet of Things Data Analytics," Procedia Computer Science, vol.163 , pp .381–390 ,2019 . doi:10.1016/j.procs.2019.12

[23] S.M Tabish , M.Z Shafiq and M Farooq , "Malware detection using statistical analysis of byte-level file content," in Proceedings of the ACM SIGKDD Workshop on CyberSecurity and Intelligence Informatics (CSI-KDD), pp .23-31 ,2009 .

[24] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," arXiv preprint arXiv:1901.03407, 2019. doi: 10.1145/3359996.

[25] Z. Cui, F. Xue, X. Cai, Y. Cao, G.G. Wang and J. Chen, "Detection of malicious code variants based on deep learning," IEEE Transactions on Industrial Informatics, vol. 14, no. 7, pp. 3187-3196, Jul. 2018. doi: 10.1109/TII.2018.2835063.

[26] A. Azmoodeh, A. Dehghantanha and K.K.R Choo, "Robust malware detection for internet of (battlefield) things devices using deep eigenspace learning," IEEE Transactions on Sustainable Computing, vol. 4, no. 1, pp. 88-95, Jan.-Mar., 2019.doi:10 .1109/TSUSC .2018 .2886192.

[27] A.Azmoodeh,A.Dehghantanha,andK.K.R.Choo,"Robustmalwaredetectionforinternetof(battlefield)thin gsdevicesusingdeepeigenspacelearning,"IEEETransactions on Sustainable Computing , vol .4 , no .1 , pp .88-95 , Jan.-Mar.,2019.doi:10 .1109/TSUSC .2018 .2886192.

[28] D. Ucci, L. Aniello, and R. Baldoni, "Survey of machine learning techniques for malware analysis," Computers & Security, vol. 81, pp. 123-147, May 2020. doi: 10.1016/j.cose.2018.11.010.

[29] D. Gibert, C. Mateu, and J. Planes, "The rise of machine learning for detection and classification of malware: Research developments, trends and challenges," Journal of Network and Computer Applications, vol. 153, p. 102526, 2020. doi: 10.1016/j.jnca.2020.102526.

[30] Z. Cui, L. Du, P.Wang, X.Cai,and W.Zhang,"Malicious code detection based on CNNs and multi-objective algorithm," Journal of Parallel and Distributed Computing , vol .129 ,pp .50-58 ,2019.doi:10 .1016/j.jpdc .2019 .02 .007.

[31] O.Or-Meir,N.Nissim,Y.Elovici,andL.Rokach,"Dynamic malware analysis in the modern era—A state of the art survey," ACM Computing Surveys (CSUR), vol .52,no .5,pages1-48 ,2019.doi:10 .1145/3341713.

[32] I.Santos,F.Brezo,X.Ugarte-Pedrero,andP.G.Bringas,"Opcode sequences as representation of executables for data-mining-based unknown malware detection," Information Sciences ,vol .231 ,pp .64-82 ,2013.doi:10 .1016/j.ins .2012 .12 .038

[33] A. Bhattacharya and R. T. Goswami, "Comparative analysis of different feature ranking techniques in data mining-based android malware detection," in Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications, Singapore, 2017, pp. 39-49, doi: 10.1007/978-981-10-6623-3_5.

[34] R. Sihwail, K. Omar, and K. Z. Ariffin, "A survey on malware analysis techniques: Static, dynamic, hybrid and memory analysis," Int. J. Adv. Sci. Eng. Inf. Technol., vol. 8, no. 4-2, pp. 1662-1671, 2018.

[35] Xianwei Gao et al., "Malware classification for the cloud via semi-supervised transfer learning," Journal of Information Security and Applications, vol. 55, p.102661-, 2020, doi:10.1016/j.jisa.2020.102661.

[36] M.Venkata Rao et al., "Deep Learning CNN Framework for Detection and Classification of Internet Worms," Journal of Interconnection Networks , vol .21 , no .4 , p .2144024-,2022.

[37] H.Alkahtani and T.Aldhyani,"Artificial Intelligence Algorithms for Malware Detection in Android-Operated Mobile Devices," Sensors (Basel Switzerland), vol .22 , no .6 , p .2268-,2022 ,doi:10 .3390/s22062268.

[38] Z.Z.Edie,"MALWARE DETECTION SYSTEM BASED ON DEEP LEARNING TECHNIQUE," Iraqi Journal of Information and Communications Technology ,vol .1,no .1,p .33-44-,2021Or-Meir, O., Nissim, N., Elovici, Y., & Rokach, L. (2019). Dynamic malware analysis in the modern era—A state of the art survey. ACM Computing Surveys (CSUR), 52(5), 1-48.

[39] A. Altaher, "An improved Android malware detection scheme based on an evolving hybrid neuro-fuzzy classifier (EHNFC) and permission-based features," Neural Comput. Appl., vol. 28, pp. 4147-4157, 2016. doi: 10.1007/s00521-016-2383-8.

[40] Z. Yuan, Y. Lu, and Y. Xue, "Droiddetector: android malware characterization and detection using deep learning," Tsinghua Sci Technol, vol. 21, pp. 114-123, 2016. doi: 10.1109/TST.2016.7444962.

[41] A. Boukhtouta et al., "Network malware classification comparison using DPI and flow packet headers," J Comput Virol Hacking Tech, vol. 12, pp. 69-100, 2016. doi: 10.1007/s11416-015-0264-x.

[42] Y. Ding et al., "A fast malware detection algorithm based on objective-oriented association mining," Comput Secur, vol. 39, no.B, pp.315-324, 2013.doi:10 .1016/j.cose .2013 .08 .003

[43] M. Eskandari, Z. Khorshidpour, and S. Hashemi, "HDM-Analyser: a hybrid analysis approach based on data mining techniques for malware detection," J. Comput. Virol. Hacking Tech., vol. 9, pp. 77-93, 2013. doi: 10.1007/s11416-013-0185-5.

[44] Q. Miao, J. Liu, Y. Cao, and J. Song, "Malware detection using bilayer behavior abstraction and improved one-class support vector machines," Int. J. Inf. Secur., vol. 15, pp. 361-379, 2016. doi: 10.1007/s10207-015-0298-y.

[45] S.D Nikolopoulos and I Polenakis, "A graph-based model for malware detection and classification using system-call groups," J Comput Virol Hacking Tech., vol. 13, pp. 29-46, 2016.doi:10 .1007/s11416-016-0271-x.

[46] S.Sheen,R.Anitha,andV.Natarajan,"Androidbasedmalwaredetectionusingamultifeaturecollaborativedecisionfusionapproach,"Neurocomputing ,vol .151 ,pp .905 -912 ,2015.doi :10 .1016/j.neucom .2014 .08 .089 .

[47] M.Norouzi,A.Souri,andM.SamadZamini,"Adataminingclassificationapproachforbehavioralmalwaredetection"J.Comput.Netw.Communications ,vol .2016 ,pp .9 ,2016.doi :10 .1155/2016/2690592 .

[48] S. Q. Salih, “A New Training Method based on Black Hole Algorithm for Convolutional Neural Network,” J. Southwest Jiaotong Univ., vol. 54, no. 3, Jun. 2019, doi: 10.35741/issn.0258-2724.54.3.22.

[49] H. Tao et al., “A Newly Developed Integrative Bio-Inspired Artificial Intelligence Model for Wind Speed Prediction,” IEEE Access, vol. 8, pp. 83347–83358, 2020, doi: 10.1109/ACCESS.2020.2990439.