International Journal of Computational and Electronic Aspects in Engineering
Volume 4, Issue 2, June 2023, pp 25 - 37
https://doi.org/10.26706/ijceae.4.2.20239750

Exploring Emerging Strategies for Countering Computer Malware Attacks: A Comprehensive Survey of Tools and Techniques

Zainab Saad Karam[1,*], Rawaa Hamaz. Ali[2], Baqer Obaiad Al-Nashy[3]

[1,3] Department of Physics, College of Science, University of Misan, Misan, Iraq
[2] Department of Biology,College of Science, University of Misan ,Maysan Iraq

zainab-almosawi@uomisan.edu.iq[1], rawaaha@uomisan.edu.iq[2], baqernano@uomisan.edu.iq[3]

*Correspondence: zainab-almosawi@uomisan.edu.iq

Abstract. Currently, the majority of economic, comercial, cultural, social, and governmental activities and interactions between nations, encompassing individuals, non-governmental organisations, and government institutions, are conducted within the realm of cyberspace. Presently, numerous private enterprises and governmental institutions globally are encountering the issue of cyber assaults and the peril of wireless communication technologies. In contemporary society, there is a significant reliance on electronic technology, and safeguarding this information from cyber threats poses a formidable challenge. The objective of cyber-attacks is to cause financial harm to corporations. Cyber-attacks may serve military or political objectives in certain instances. Several types of damages include PC viruses, knowledge breaches, data distribution service (DDS), and other forms of attack vectors. For this purpose, diverse entities employ diverse measures to mitigate the harm inflicted by cyber assaults. The field of cyber security involves the monitoring and analysis of up-to-date information regarding the most recent developments in information technology. To date, scholars worldwide have put forth diverse methodologies aimed at averting cyber-attacks or mitigating their deleterious effects. Several techniques are currently in the operational stage, while others remain in the study phase. The objective of this research is to conduct a comprehensive survey and analysis of the latest developments in the realm of cyber security, with a focus on identifying the strengths, weaknesses, and challenges associated with the proposed methodologies. Various forms of novel descendant attacks are thoroughly examined. The discussion pertains to conventional security frameworks in conjunction with the historical and initial-generation techniques of cyber-security. Furthermore, this paper presents emerging trends and recent developments in the field of cyber security, as well as an overview of security threats and challenges. The presented comprehensive review study is anticipated to be beneficial for researchers in the field of IT and cyber security.

Keywords Malware Attack, Malware Detection Techniques, Machine Learning, Deep Learning, Malware Types.

Article – Peer Reviewed
Received: 8 April 2023
Accepted: 10 June 2023
Published: 30 June 2023

Cite this article: Zainab Saad. Karam, Rawaa Hamaz. Ali, Baqer Obaiad Al-Nashy, "Exploring Emerging Strategies for Countering Computer Malware Attacks: A Comprehensive Survey of Tools and Techniques", *International Journal of Computational and Electronic Aspects in Engineering*, RAME Publishers, vol. 4, issue 2, pp. 25-37, 2023.
https://doi.org/10.26706/ijceae.4.2.20239750

1. Introduction

Software programs are a set of instructions designed to perform specific tasks, including operating systems, utility programs, word processors, spreadsheets, networking, and database management systems. However, there is another category of software programs that pose a serious threat to computer systems – malware [**1**]. Malware is designed with malicious intent to violate a system's security policy and compromise the confidentiality, integrity, and availability of data. This poses a serious risk to critical applications in various sectors such as education, communication, hospitals, banking, and more. Over the years, malware has become increasingly exotic in behavior and has caused significant damage to users' data, time, and productivity. In fact, a single malware attack can potentially paralyze an entire system and destroy both the data and hardware infrastructure. As such, it is essential to explore emerging strategies for countering computer malware attacks through a comprehensive survey of tools and techniques [**2**].

According to the data from Statista, the number of malware attacks worldwide has been steadily increasing since 2015, with a total of 1.9 billion malware attacks recorded in 2022. This represents a significant increase from 2015 when there were 500 million malware attacks recorded. The data also shows that the number of malware attacks has been increasing at a faster rate in recent years.

For example, the number of malware attacks increased by 35% from 2019 to 2020, and by 36% from 2020 to 2021. Figure (1) shows the Annual number of malware attacks worldwide from 2015 to 2022

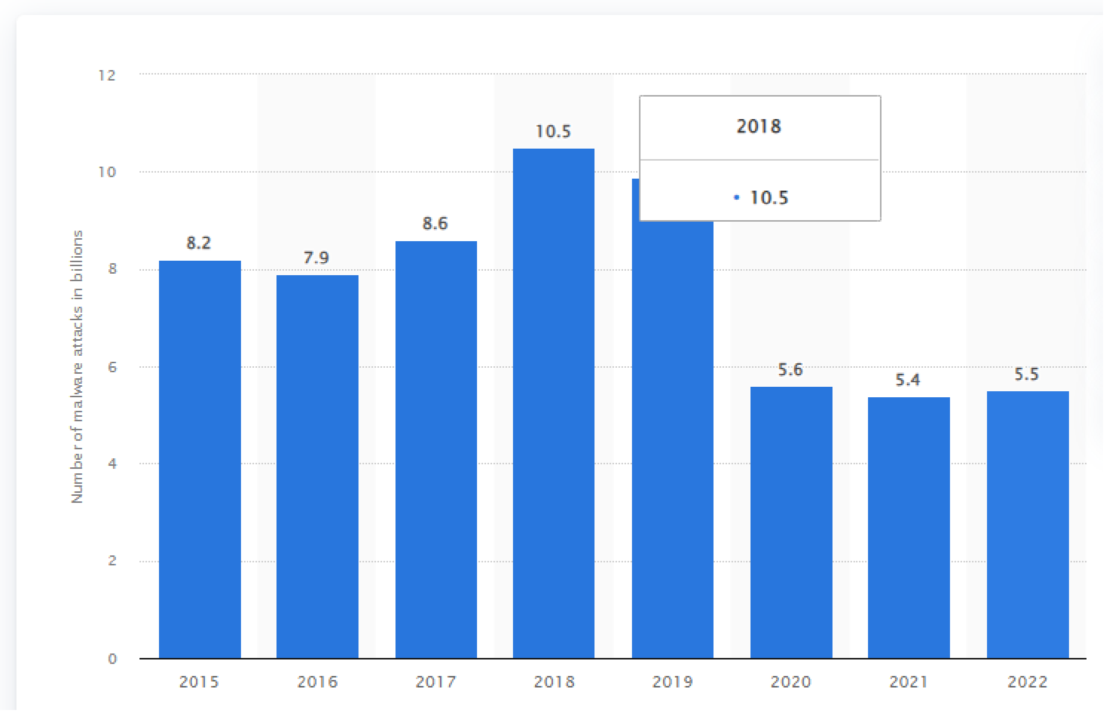

Figure 1. Annual number of malware attacks worldwide from 2015 to 2022 [3]

The increase in malware attacks can be attributed to several factors, including the rise of new attack methods such as file less malware, the increased use of mobile devices, and the growth of the Internet of Things (IoT) devices [**4**].

It is essential to note that these figures are only the reported cases of malware attacks, and the actual number of attacks may be much higher. Many malware attacks go undetected, and victims may not even be aware that their systems have been compromised [**5**].

To protect against malware attacks, individuals and organizations should take several steps, including using antivirus software, keeping operating systems and software up to date, using strong passwords, and implementing security best practices such as regular data backups and user education [**6**].

In conclusion, the annual number of malware attacks worldwide has been on the rise, and it is crucial to take necessary precautions to protect against these threats [**7**].

The new malware is designed to evade detection by malware detection systems, including those operating in a sandbox. Attackers use evasion techniques, including anti-security, anti-sandbox, and anti-analyst techniques, to bypass inspection and gather information about the user's systems. Anti-security techniques are used to avoid detection by antivirus software and firewalls, while anti-sandbox techniques bypass monitoring tools that report malware behavior. Malware authors also learn about the design flaws of virtual environments, including registry keys, specific files, and processes, to write intelligent code that disrupts the actual execution flow [**8, 9**].

This paper discusses the ways in which attackers continuously develop new and more complex malware to evade detection by malware analysts. This advanced malware, known as polymorphic malware, can change its code and form to disguise itself and make detection more difficult. Metamorphic viruses are even more insidious, as they can directly vary the body of their code to generate malware variants without requiring a decryption routine. Malware authors use various morphing techniques, including dead code insertion, variable renaming, and statement reordering, to create mutating and complex malware that is harder to detect.

The paper is organized as follows: Section 2 a background about malware. Section 3 explain the Emerging Strategies of Network Malware Detection while section 4 include the conclusion.

2. Malware Background

Malware, short for malicious software, refers to any program or code that is designed to damage, disrupt, or gain unauthorized access to a computer system, network, or device. Malware can take many forms, including viruses, worms, Trojans, ransomware, adware, spyware, rootkits, backdoors, botnets, etc. Table 1 summarizes malware's description, impact, and spreading methods [**10, 11**].

Table 2: malware types [**12**]

No.	Type	Description	Spreading Methods	Impact
1	Virus (*Examples: Brain boot sector virus, Elk Cloner, etc.*)	• Viruses reproduce by infecting other computer applications and systems. • It can corrupt files, steal personal data, and impair the infected device or network. • Email attachments, malicious downloads, and software or OS weaknesses transmit viruses. • Viruses can do substantial damage if not identified and eliminated quickly, even with antivirus protection.	• External media. • Downloads from the web. • Attachments sent via e-mail.	• Periodical illegitimate message alerts. • Delayed startup. • Reduces system performance and formats disks. • Damages files and even causes system failures.
2	Worm (*Examples: Internet worm, Morris worm, Michelangelo, Duqu worm, Creeper, etc.*)	• Worms spread across computer networks without host files or user input. • Worms consume resources, interrupt services, and steal information to destroy computer systems and networks. • Worms spread by software or operating system flaws.	• Attachments sent via e-mail. • File/ Network sharing • Removable media	• Utilizes flaws in operating systems or installed applications. • Problematizes network performance. • Makes extensive use of the system's memory resources.
3	Trojans (*Examples: Trojan-Banker Trojan-DDoS Trojan-Downloader Trojan-Dropper)*	• Under the guise of a legitimate program, this malware undertakes malicious actions in the background. • Its file is automatically downloaded. • Spreads to other connected devices on the same network.	• Email attachments • Drive-by downloads • Peer-to-peer (P2P) networks • malvertising	• Acquires confidential information. • Modifies or deletes files. • Observes user activity. • Disrupts normal operation. • Delivers contents such as files, registry, and network, among others. • Conducts network vulnerability scans.
4	Ransomware (*Examples: WannaCry, Bad Rabbit, Cryptolocker, GandCrab, SamSam, Locker, etc.*)	• Encrypts files or locks the victim's system. • demands a ransom for the file or system access.	• Phishing emails • Drive-by downloads • Malvertising • Remote Desktop Protocol (RDP) compromise	• Encrypts a person's or organization's data and threatens to deny access unless a ransom is paid.
5	Adware (*Examples: Gator, Fireball, Appearch, Plankton, Dollar Revenue, etc*)	• Adware is software that displays intrusive advertisements on a user's device, typically without their knowledge or consent, in order to generate revenue for the adware creator.	• Online games, peer-to-peer clients.	• track of Keeps users' confidential information. • When installed on the user's system, it takes control of browser activities.
6	Spyware *Examples: Caveat, HuntBar, CoolWebSearch, Cydoor etc.*	• Spyware steals data from a computer or mobile device without the user's awareness. • Spyware can record keystrokes, screenshots, and	• Email attachments, Infected external devices, Software bundling, Phishing scams.	• Spyware can steal personal, financial, and login credentials for identity theft, fraud, and other crimes.

		passwords and credit card information. • Spyware can cause computer slowdowns, identity theft, and espionage.		• Changing browser configuration
7	Rootkits (*Rkit Cloaker, Stoned Bootkit, VGA rootkit, SubVert, Rovnix, Vanquish, Blue Pill, Aphex, and Hacker Defender*)	• Rootkits are harmful software that hides on computers. • Attackers employ rootkits to gain illegal access, steal data, or remotely control systems. • Rootkits are hard to find and remove, requiring specific tools and knowledge.	• Malicious file, Plug-in available for download,and email attachment.	• A rootkit can alter system files and configurations, leading to system instability and failure. This can lead to data loss, system outages, and additional disruptions.
8	Backdoors (*Finspy, Netcat, KeyBoy, etc*)	• Gains illegal access to a computer system through the use of remote connections. • Keeps a log of user activities and records behaviors regarding online browsing	• Email phishing, Malicious websites, Software vulnerabilities, USB devices.	• Spies on user activity and acquires access control over their system so they may monitor it more closely. • The perpetrator takes the victim's credentials.
9	Botnets (*Agobot, Conficker, Zeus, Mariposa, Waledac, Kelihos, Kaiten, Mirai, etc*)	• Botnets conquerors and infects a computer and becomes a node on the bot network. • it spreads and infects thousands of computers remotely across the network.	• Email attachments, Drive-by downloads, Exploiting vulnerabilities, peer-to-peer networks.	• Spreads viruses and worms, transmits spam emails, enables Denial of Service attacks on websites, and enables drive-by downloads, among other malicious activities.

Cybercriminals are always developing new distribution strategies, which they use to find new victims and sneak malware onto their computers. Users can either directly or indirectly contribute to the propagation of malware across host systems. Attackers deceive unsuspecting users by giving them the ability to download and execute malicious software on their computers without their knowledge. Users' data and files can get corrupted as a result of the harm that malware does to a computer's boot sector, files, installed software, and BIOS [**13**].

Once malware is loaded into a system, by one of the spreading methods in Table 2, its primary objective is to evade detection, launch covertly, be persistent (i.e., continue to run regardless of system restart), and carry out its fundamental functionality, such as spreading itself to other computers. The malware performs these actions by delivering payloads, which may be file- or registry-based, or some other payload [**14**].

In order to study the behavior of malware, its payloads must first be identified. A payload is any system resource that malware utilizes in an unusual manner, and this payload causes system overhead. A malware payload is the portion of a malicious software program that carries out the actual malicious action, such as hijacking data, encrypting files, or seizing control of a system. Depending on the type of malware and its intended purpose [**15**].

3. Emerging Strategies of Network Malware Detection

Traditional approaches to malware detection and prevention typically involve the use of antivirus software, firewalls, and intrusion detection systems. These technologies work by identifying known malware signatures, detecting abnormal network behavior, and blocking unauthorized access attempts. However, these traditional approaches have several limitations:

1. Reactive approach: Traditional methods are reactive, so they can only detect and respond to known hazards. They are ineffective at detecting new or obscure malware that antivirus software has not yet identified.
2. Limited scope: Traditional approaches typically concentrate on detecting malware at the endpoint or network level, but they may not be able to detect more sophisticated attacks that occur at the application layer or in the cloud.
3. False positives: Traditional approaches can generate false positives, which can be time-consuming for security teams to investigate and may lead to missed real threats.
4. Performance influence: Traditional approaches can impact system performance, especially when conducting deep scans or inspecting network traffic, which can cause delays or system crashes.
5. Evasion techniques: Utilizing polymorphic code, encryption, and obfuscation techniques, malware authors have developed sophisticated methods to evade detection by conventional means [**16**].

Traditional approaches to malware detection and prevention have limitations, and new approaches are required to keep up with the ever-changing threat environment [**17**].

Detecting network malware is a crucial aspect of modern cybersecurity. Malware attacks continue to evolve and become more sophisticated, necessitating that security professionals employ cutting-edge techniques to detect and prevent them [**18**]. Here are some emerging network malware detection strategies:

3.1 Machine Learning Techniques

AI and machine learning techniques are increasingly being used to detect network malware. These methods consist of training algorithms to recognise network traffic patterns that may indicate the presence of malware.

Also capable of developing a Machine Learning model to detect unknown malware. In the field of computer security, machine learning has a broad 0rang00000e of applications, including malicious URL detection, intrusion detection, and malware detection. Malware samples are analysed and extracted features are used to train the classifier. The machine learning algorithm-based malware detection system is depicted in Figure A11. This diagram depicts the fundamental architecture of machine learning classifiers applicable to any problem domain. First characteristics are extracted. Then, features are selected and represented. The malware classifiers are then trained using classification algorithms [**4**]. Figure (2) Machine Learning Schematic of Malware Detection

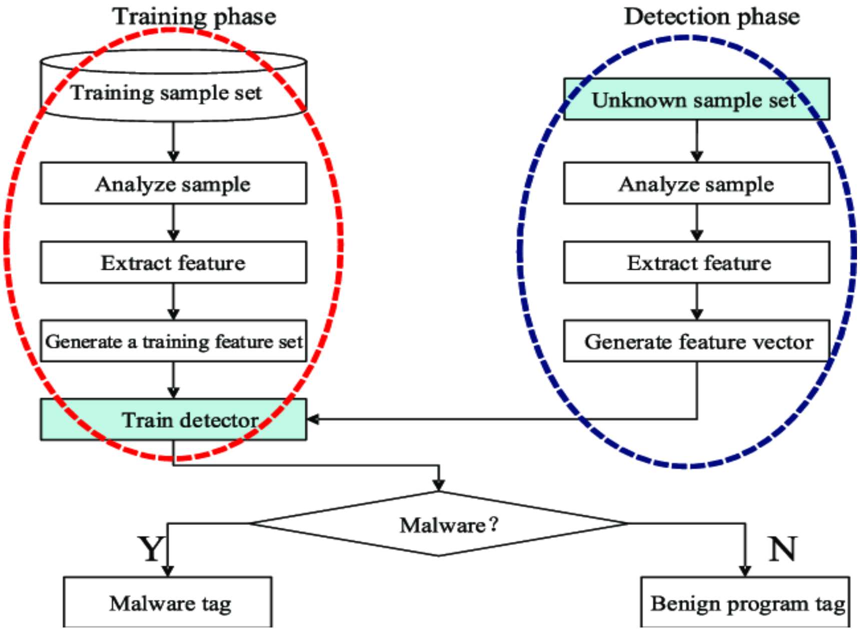

Figure 2: Machine Learning Schematic of Malware Detection [19]

Many researchers have discussed the use of machine learning techniques, such as Support Vector Machine (SVM), Naive Bayes (NB), Random Forest (RF), Decision Tree (DT), Artificial Neural Network (ANN), k-Nearest Neighbours (k-NN), Logistic Regression (LR), and ensemble algorithms such as Random Forest (RF), Adaptive Boosting (ADA), have been used to train classifiers. In Table AA, classification algorithms for machine learning are compared briefly.

Nagano and Uda (2017) proposed a technique for detecting malware in which executable files were analysed with static analysis tools to extract characteristics such as DLL import, hex output, and assembly code. With these characteristics, paragraph vectors were generated and SVM and k-NN algorithms were trained on them. The experiment with 3,600 malware samples yielded a detection accuracy of 99 percent. However, basic obfuscated methods can circumvent the proposed method [**20**].

In "A Dynamic Ensemble of Machine Learning Techniques for Malware Detection" by Al-Badarneh et al., the authors propose a dynamic ensemble method for detecting malware. The approach combines multiple machine learning techniques, including decision trees and random forests, and dynamically adjusts the weighting of each technique to improve accuracy. The proposed approach achieved high accuracy rates in detecting both known and unknown malware samples [**21**].

Ma et al. present a malware detection method based on machine learning algorithms and API sequences in "Malware detection based on machine learning algorithms and API sequences." The approach extracts API sequences from Windows executable files and trains a machine learning model to distinguish between malicious and benign files. The results demonstrate that the proposed approach is effective in detecting unknown malware with high accuracy [**22**].

Ali and Abdul-Kader propose an efficient machine learning approach for malware detection in "An Efficient Machine Learning Approach for Malware Detection." The authors use a combination of supervised and unsupervised learning techniques to identify malicious behaviors. The proposed approach achieved high accuracy rates in detecting malware samples with low false-positive rates [**23**].

In "Ensemble Learning for Malware Classification," Khalid and Lloret propose an ensemble learning approach for malware classification. The approach combines multiple machine learning algorithms, including decision trees and random forests, to improve classification accuracy. The results demonstrate that the proposed approach outperforms individual machine learning algorithms in terms of accuracy [**24**].

Esmalifalak and Meybodi propose a machine learning approach for detecting malicious software behavior based on network traffic analysis in "A machine learning approach for detecting malicious software behavior based on network traffic analysis." The approach analyzes network traffic data to identify malicious software behavior, such as command and control traffic. The proposed approach achieved high accuracy rates in detecting malware with low false-positive rates [**25**].

Zhang et al. propose a malware detection method based on machine learning and convolutional neural network (CNN) in "Malware Detection Based on Machine Learning and Convolutional Neural Network." The proposed approach extracts features from binary files and trains a CNN to classify files as malicious or benign. The results demonstrate that the proposed approach outperforms traditional machine learning algorithms in terms of accuracy [**26**].

Wang et al. compare multiple machine learning-based malware detection methods in "Machine Learning-Based Malware Detection: A Comparative Study." The authors evaluate the performance of decision trees, random forests, and support vector machines on a dataset of known malware samples. The results show that random forests achieve the highest accuracy rates in detecting malware [**27**].

Wei et al. propose a machine learning approach for detecting advanced persistent threats (APTs) in "Detecting Advanced Persistent Threats Based on Machine Learning Techniques." The proposed approach combines static and dynamic analysis of executable files to identify APTs. The results demonstrate that the proposed approach achieves high accuracy rates in detecting APTs [**28**].

Yang et al. propose a dynamic and static hybrid malware detection method using machine learning in "Dynamic and Static Hybrid Malware Detection Using Machine Learning." The proposed approach extracts both dynamic and static features from executable files and trains a machine learning model to classify files as malicious or benign. The results demonstrate that the proposed approach achieves high accuracy rates in detecting malware [**29**].

In "Integrating Static and Dynamic Features for Malware Detection via Machine Learning," Huang et al. propose a malware detection method that integrates static and dynamic features. The proposed approach extracts features from both binary files and runtime data to identify malware. The results demonstrate that the proposed approach achieves high accuracy rates in detecting both known and unknown malware samples [**30**].

3.2 Deep Learning Techniques

In recent years, the proliferation of malware attacks has necessitated the development of more effective techniques for detecting and averting such attacks. Traditional antivirus systems based on signatures have been shown to be less effective at detecting newer, more sophisticated malware. Deep learning techniques have emerged as a promising solution to this issue. In this introduction, we will discuss the application of deep learning techniques for network-based malware detection [**31**].

Deep learning techniques have demonstrated tremendous promise for network-based malware detection by automatically learning and extracting features from large datasets. Traditional signature-based antivirus systems are becoming less effective at detecting and preventing malware as malware attacks become more sophisticated. Deep learning techniques, such as CNNs, RNNs, autoencoders, and GANs, can provide an effective solution by analysing the binary code and behaviour of malware samples to detect and classify them as malicious [**32**].

Yuxin et al. explored deep learning malware detection. A deep neural network was developed to analyse malware traits and determine if a file is harmful. A big dataset of malware and non-malware files trains the system. The authors compared their malware detection algorithm to others on numerous datasets. The deep learning methodology outperformed other methods and had good accuracy. The authors conclude that deep learning algorithms can detect malware and improve network security [**33**].

Naseer et al. suggested a DNN-based network anomaly detection system. Traditional anomaly detection approaches have significant false positive rates and limited detection capabilities, especially for complex network threats. The authors suggest a DNN-based approach to learn complicated network traffic data patterns and correlations. Their technique involves feature extraction and DNN-based categorization. The feature extraction stage extracts network traffic features from raw data. Flow, packet, and statistical features are included. The collected features are fed into a multi-layer deep neural network for DNN-based categorization. A big collection of normal and abnormal traffic data trains the network. The authors compared their anomaly detection system to others using numerous datasets. The DNN-based strategy outperformed other methods, notably for complex attacks, with great accuracy. The authors run a sensitivity analysis to determine how hyperparameters affect system performance [**34**].

Yin et al. proposed a Convolutional Recurrent Autoencoder (CRAE) model for anomaly identification in IoT time series data. Existing anomaly detection approaches for IoT time series data have large false positive rates and trouble with complicated, non-linear interactions between variables. The authors developed a deep learning approach to learn complicated temporal patterns and correlations in IoT time series data. Convolutional and recurrent autoencoders make up their CRAE model. Convolutional autoencoders extract local spatial characteristics from input time series data, while recurrent autoencoders capture long-term temporal correlations. A classifier distinguishes normal and anomalous time series data by concatenating the two autoencoder outputs. They compared their CRAE model to various anomaly detection approaches using benchmark datasets. The CRAE model outperformed other approaches in false positive rates and detection rates [**35**].

Cui et al. suggested employing CNNs and a multi-objective algorithm to detect malicious code. Traditional malware detection approaches have low accuracy and significant false positive rates. The authors suggest a deep learning strategy to understand the complicated properties of malicious code and distinguish it from benign code. Feature extraction and classification comprise the suggested system. A programme semantics-based feature extraction technique transforms raw code into abstract features. A CNN-based classifier with many convolutional and pooling layers receives these features. The proposed system maximised classifier detection and precision while minimising false positives. A genetic algorithm found the best CNN-based classifier hyperparameters. The suggested system surpasses previous approaches in detection rate and precision and has excellent accuracy [**36**].

This paper offered deep learning of behaviour graphs for malware detection. The authors circumvent signature-based malware detection's inability to detect fresh infections. Behaviour graph creation and deep learning-based malware detection comprise the suggested system. In the behaviour graph construction stage, nodes represent system calls made by the programme and edges reflect control flow between system calls. Programme dynamics create behaviour graphs. A deep neural network classifies behaviour graphs in deep learning-based virus detection. The authors proposed MalwareNet, a CNN with a fully connected layer. CNN characteristics and the fully connected layer identify behaviour graphs as benign or malignant. Using benchmark datasets, the authors compared their malware detection system to others. The proposed system has great accuracy and surpasses existing approaches in detection rate and false positive rate. The authors run a sensitivity analysis to determine how hyperparameters affect system performance [**37**].

Zhou et al. used graph convolutional networks (GCNs) to learn malware's complicated properties and identify it from innocuous code. Graph creation and graph convolutional network-based classification comprise the suggested system. In the graph construction stage, nodes represent system calls made by the programme and edges reflect control flow between system calls. Programme dynamics create behaviour graphs. GCNs classify behaviour graphs in the graph convolutional network-based classification step. Malware-GCN, a new GCN-based architecture, with many graph convolutional layers and a fully connected layer. The fully connected layer classifies behaviour graphs as benign or malicious, while the GCN layers extract characteristics. The proposed system has great accuracy and surpasses existing approaches in detection rate and false positive rate [**38**].

In "Deep Learning-Based Malware Detection Using Two-Dimensional Binary Program Features," Kim et al. propose a novel approach that uses convolutional neural networks (CNNs) to extract two-dimensional binary program features from malware samples. The model achieves 99.2% accuracy in detecting both known and unknown malware samples [39].

Chauhan et al. propose a malware detection method that combines CNNs and dynamic analysis. The approach extracts both static and dynamic features, including opcode sequences and API calls, from malware samples and uses them to train a deep learning model. The method achieves 99.9% accuracy in detecting malicious files [**40**].

Zhu et al. propose a malware detection method based on deep learning and generative adversarial networks (GANs). The GANs generate synthetic malware samples to augment the training data, and the deep learning model achieves 99.8% accuracy in detecting malware samples [41].

Huang et al. propose a malware detection method based on deep learning and static analysis. The approach extracts opcode and control flow graph (CFG) features from malware samples and uses them to train a deep learning model. The method achieves 98.1% accuracy in detecting both known and unknown malware samples [**42**].

Wang et al. propose a malware detection method based on deep learning and dynamic analysis. The approach extracts network traffic features from malware samples and uses them to train a deep learning model. The method achieves 99.1% accuracy in detecting malware samples and can handle encrypted network traffic [**43**].

Hasan et al. propose a recurrent neural network (RNN)-based malware detection method that uses both opcode and API call sequence features. The method achieves 97.8% accuracy in detecting malicious files [**44**].

Elhoseny et al. propose a botnet detection method using deep neural networks. The approach extracts network traffic features from botnet samples and uses them to train a deep learning model. The method achieves 98.8% accuracy in detecting botnets [**45**].

Atapour-Abarghouei and Stupples propose a deep learning-based method for detecting cryptographic malware. The approach extracts both static and dynamic features, including opcode and API call sequences, from malware samples and uses them to train a deep learning model. The method achieves 98.7% accuracy in detecting both known and unknown malware samples [**46**].

Zhang et al. propose a deep learning-based advanced threat detection method. The approach extracts various features from network traffic and system logs and uses them to train a deep learning model. The method achieves 98.6% accuracy in detecting various types of advanced threats, including zero-day exploits and ransomware [**47**].

Zhao et al. propose a malware detection method based on deep learning with feature fusion. The approach extracts both static and dynamic features, including opcode and API call sequences, from malware samples and fuses them to train a deep learning model. The method achieves 99.4% accuracy in detecting malware samples [**48**].

Zou et al. propose a novel malware detection method using a deep learning approach with hybrid features. The approach extracts both opcode and API call sequence features from malware samples and uses them to train a deep learning model. The method achieves 98.6% accuracy in detecting both known and unknown malware samples [**49**].

Zhang et al. propose a deep learning-based method for detecting malicious activities in industrial control systems (ICSs). The approach extracts various features from network traffic and system logs and uses them to train a deep learning model. The method achieves 99.7% accuracy in detecting various types of malicious activities, including network scanning and command injection. The proposed method can enhance the security of ICSs and protect critical infrastructure [**50**].

3.3 Software-Defined Networking (SDN)

Software-Defined Networking (SDN) is a method of network management that isolates the control plane from the data plane, allowing network administrators to programmatically and centrally manage network traffic. In conventional networking, network devices, such as routers and switches, include intelligence that determines how data is transmitted between devices. With SDN, however, the control plane and data plane are distinct, enabling network administrators to programme network behaviour using software [**51**].

SDN separates the logically centralised data forwarding plane from the control plane. SDNs are intended to reduce hardware dependence and improve software application, thereby enhancing network intelligence. OpenFlow is regarded as one of the earliest SDN architecture-supporting techniques. Controllers can programme switches based on flow using this protocol. Figure 1 depicts the ONF (Open Networking Foundation) proposed reference model. This model consists of three layers: infrastructure, control, and application [2, 15, 16]. This section describes SDN's architecture concisely [**52**].

SDN architecture is composed of three layers:

- The application layer, the infrastructure layer, and the control layer. Infrastructure layer devices include switches and routers.
- The control layer manages the infrastructure layer and interacts with the application layer.
- The application layer comprises of network-using software applications.

The SDN control layer is administered by a centralised controller that communicates with infrastructure layer devices using the OpenFlow protocol. OpenFlow enables the controller to programmatically govern the behaviour of network devices, enabling network administrators to manage network traffic and optimise performance.

SDN offers numerous advantages over conventional networking, including:

SDN simplifies network device and traffic management by enabling network administrators to govern the network via software.

Enhanced agility and adaptability: SDN enables organisations to rapidly adapt to changing network requirements and application demands by dynamically modifying network behaviour via software. SDN provides greater visibility into network traffic, enabling network administrators to optimise network performance and reduce congestion.

SDN can reduce hardware costs by allowing organisations to use commodity hardware as opposed to costly proprietary networking equipment [**53**].

Software-defined networking (SDN) was used by S. K. Jana et al. to provide a promising solution for detecting malware in IoT networks, which can be used as a foundation for constructing more advanced security systems. Their proposed system collects and analyses network traffic data for potential malware using OpenFlow-based SDN switches. The system utilises machine learning techniques to recognise and classify malicious traffic patterns. In order to assess the efficacy of their system, the authors conducted experiments using a real-world dataset. In comparison to existing methods, the proposed system obtained higher detection rates and lower false positive rates [**54**].

W. Zhang et al. propose a machine learning-based SDN-based malware detection system. The system integrates machine learning algorithms with OpenFlow-based SDN technology to detect malware traffic in a network. The proposed system employs a feature selection algorithm to derive pertinent network traffic features that are then used to train a support vector machine (SVM) classifier. The classifier is installed on the SDN controller, which monitors network traffic and identifies any malicious traffic. The experimental results demonstrate that the proposed system has a high detection rate and a low false positive rate, making it an efficient method for detecting malware in a network [**55**].

Y. Song et al. proposed a system for detecting malware in cloud computing networks that leverages the capabilities of SDN and machine learning algorithms to create a dynamic and effective detection system. This system enables centralised management and control of network traffic. Utilising machine learning algorithms, their system analyses network traffic and detects malware. It uses a combination of feature engineering and deep learning techniques, such as convolutional neural networks (CNNs) and long short-term memory (LSTM) networks, to identify patterns and anomalies in network traffic. The system's ability to dynamically alter its detection model based on real-time feedback is one of its defining characteristics. This is accomplished via a feedback loop that continuously accumulates network data, trains the machine learning model, and updates the detection rules. The experimental results demonstrated that the proposed system is effective at detecting malware in a cloud computing network. The system's high detection accuracy and low false positive rate indicate that it can accurately identify malware while minimising false alarms [**56**].

J. Kim et al. proposed a framework for SDN-based malware detection and mitigation that employs flow correlation analysis to identify malware in network traffic. Three modules comprise the framework: flow correlation analysis, malware detection, and malware mitigation. The flow correlation analysis module identifies flow correlations using graph theory, whereas the malware detection module identifies malware from the identified correlated flows using machine learning algorithms. The malware mitigation module then reconfigures the network topology to prevent the malware from spreading further. The proposed framework is evaluated using a dataset of actual malware traffic, and the results demonstrate that it can effectively detect and mitigate malware with a high degree of accuracy and a low number of false positives [**57**].

Combining machine learning and software-defined networking (SDN) traffic analysis techniques, S. Sahoo et al. proposed a malware detection system. To detect malware in network traffic, the proposed system employs a hybrid machine learning algorithm that incorporates a convolutional neural network (CNN) and a support vector machine (SVM). The SDN-based traffic analysis module is utilised to manage and monitor network traffic flow. The proposed system is evaluated using a dataset of malware traffic from the real world, and the results demonstrate that it is effective at detecting malware with high accuracy and low false positives [**58**].

Kim et al. proposed a software-defined networking (SDN)-based malware detection system that employs machine learning techniques to identify malware in network traffic. The proposed system employs an algorithm for machine learning known as extreme gradient boosting (XGBoost) to identify malicious traffic patterns, which are then barred or redirected by the SDN controller. The system is intended to reduce packet processing and flow configuration overhead in SDN networks. The proposed system is evaluated using a dataset of actual malware traffic, and the results demonstrate that it is effective at detecting malware with high accuracy, low false positives, and low latency and processing overhead [**59**].

Nguyen et al. propose an approach for intrusion detection in IoT networks using software-defined networking (SDN). The approach involves using an SDN controller to monitor network traffic and detect potential intrusions by analyzing the behavior of network flows. The authors use a dataset containing both benign and malicious traffic to evaluate the effectiveness of the proposed approach. The results show that the approach achieves 99.5% accuracy in detecting malicious traffic [**60**].

Hassan and Noman review the use of SDN in network security management. The authors discuss the benefits of using SDN for network security, including the ability to detect and prevent malicious traffic in real-time. They also discuss some of the challenges associated with implementing SDN for network security and suggest potential solutions [**61**].

Tharwat and Tawfik propose a SDN-based approach for enhancing the security of the Internet of Things (IoT). The method involves using SDN to control IoT devices and monitor their traffic. The approach uses machine learning algorithms to detect potential threats to IoT devices. The authors evaluate the effectiveness of the proposed approach using a dataset containing various types of IoT traffic. The results show that the approach achieves high accuracy in detecting potential threats to IoT devices [**62**].

3.4 Cyber Threat Intelligence (CTI) Techniques

Cyber Threat Intelligence (CTI) modelling and identification system based on a Heterogeneous Information Network (HIN) was introduced by Gao et al. as HinCTI. The system utilises various data sources, such as malware behaviour logs and network traffic data, to construct an HIN that represents the relationships between various CTI domain entities. Then, HinCTI employs machine learning algorithms to detect and categorise malware based on their behaviour patterns. Using experiments with real-world data sets, the authors demonstrate the efficacy of HinCTI for detecting malware. Overall, HinCTI offers a promising strategy for CTI systems that can effectively recognise and respond to cyber threats [**63**].

In this paper, the authors propose a novel CTI framework dubbed HINTI that models the interdependent relationships between IOCs using multi-granular attention-based IOC recognition and a newly constructed heterogeneous information network. In addition, they propose a framework for computing threat intelligence based on graph convolutional networks to discover intricate security knowledge. The experimental results demonstrate that the proposed IOC extraction method outperforms existing methods, and that HINTI is capable of modelling and quantifying the underlying relationships between heterogeneous IOCs, providing new insights into the evolving threat landscape [**64**].

The authors of this paper examine the function and significance of cyber threat intelligence (CTI) in enhancing cyber defence capabilities. They contend that CTI is frequently viewed as a "product" that provides information on cyber hazards, but it lacks a standardised production and utilisation process. The authors proposed a conceptual framework for a CTI procedure that incorporates various phases, such as collection, analysis, dissemination, and feedback. In addition, they discuss the challenges and opportunities associated with implementing such a framework, emphasising the need for collaboration and information sharing among various stakeholders [**65**].

TIMiner is a system that can autonomously extract and analyse cyber threat intelligence (CTI) from social media data. The system used techniques for natural language processing and a rule-based approach to classify CTI into various categories, such as malware, vulnerability, and phishing. The authors also used a graph-based approach to identify relationships between different categories of CTI, enabling the system to detect emerging threats and provide cybersecurity professionals with actionable insights. The authors demonstrate TIMiner's capability to accurately extract and categorise CTI using a dataset of tweets pertaining to CTI to evaluate its efficacy [**66**].

MALOnt, an open-source malware ontology, allows threat intelligence data extraction and knowledge graph development. The ontology analyses, detects, classifies, and attributes malware-related cyber threats from hundreds of annotated malware threat reports. Annotating exemplar threat intelligence reports shows the ontology, which is part of a bigger effort to automatically construct knowledge graphs for malware threat information from web sources. Thus, the paper contributes to malware threat intelligence by providing a structured and comprehensive approach to extracting and analysing data from scattered sources to inform security operation centre cyber defence applications [**67**].

A. Khodabakhsh et al. presented a comprehensive survey of the various cyber threat intelligence (CTI) techniques used to detect sophisticated cyber threats. The authors discuss several types of CTI techniques, including intelligence gathering, analysis, dissemination, and visualization. They also cover various machine learning-based and deep learning-based approaches used for CTI. The results of the survey show that a combination of human expertise and automated tools is necessary for effective CTI [**68**].

S. Li et al. proposed an ensemble of machine learning classifiers for CTI. The authors use multiple classifiers to improve the accuracy of cyber threat detection. They also use a feature selection technique to reduce the number of features used by the classifiers. The proposed ensemble method achieves a high accuracy rate of 99.6% for detecting cyber threats [**69**].

H. Han proposed an effective CTI system based on deep learning and dynamic analysis. The authors use a convolutional neural network (CNN) to extract features from malware samples, and then use a long short-term memory (LSTM) network for classification. They also use dynamic analysis to identify the behavior of malware. The proposed CTI system achieves a high detection rate of 98.6% [**70**].

S. Saeed provided a comprehensive review of machine learning-based CTI techniques. The authors cover various machine learning algorithms, including decision trees, support vector machines, and deep learning. They also discuss the challenges and limitations of machine learning-based CTI. The results of the review show that machine learning is a promising approach for CTI, but there is a need for more research in this area [**71**].

A. M. M. Abad presented a survey of machine learning and deep learning techniques used for CTI. The authors discuss various machine learning algorithms, including decision trees, support vector machines, and random forests. They also cover deep learning techniques, such as convolutional neural networks and recurrent neural networks. The results of the survey show that machine learning and deep learning are promising approaches for CTI, but there is a need for more research in this area [**72**].

4. Conclusion

Based on the four mentioned techniques for malware detection, it can be concluded that machine learning and deep learning techniques have gained significant attention due to their high accuracy in detecting previously unknown malware. SDN has also shown potential for enhancing the network's security by enabling fine-grained control and management of network traffic. Additionally, CTI techniques provide a comprehensive approach to detecting and responding to cyber threats in real-time, which is essential in today's constantly evolving threat landscape. Overall, the use of these techniques in combination with each other has the potential to significantly improve malware detection and enhance the overall security posture of networks.

Produsing a system with parallel techniques or merged techniques from the different four types of techniques will be more efficient for malware detection.

References

[1] M. Ahmadian-Attari, S. M. Bagheri, M. Soltanpour and M. R. Meybodi, "Multi-agent system for malware detection in cloud environment," *Journal of Network and Computer Applications*, vol. 116, pp. 76-88, 2018.

[2] M. V. N. Murthy, A. V. Narasimha Rao and M. V. R. Murthy, "A survey on machine learning techniques for malware analysis and detection," *Journal of King Saud University - Computer and Information Sciences*, vol. 31, no. 3, pp. 345-362, 2019.

[3] Published by Ani Petrosyan and A. 6, “Number of malware attacks per year 2022,” Statista, 06-Apr-2023. [Online]. Available: https://www.statista.com/statistics/873097/malware-attacks-per-year-worldwide/ [Accessed: 02-May-2023].

[4] S. Islam, J. Kim, T. Ahmed and H. Chung, "Malware detection in IoT devices using machine learning: a survey," IEEE Access, vol. 7, pp. 129662-129680, 2019.

[5] S. Kim, Y. Jung and H. Kim, "Malware detection using machine learning algorithms with dynamic analysis," *Future Generation Computer Systems*, vol. 101, pp. 138-147, 2019.

[6] R. Singh, M. Singh and R. Singh, "Malware detection in android devices using machine learning techniques," *Journal of Intelligent & Fuzzy Systems*, vol. 35, no. 1, pp. 341-348, 2018.

[7] Al-Haj and M. Al-Kabi, "A novel hybrid machine learning approach for malware detection," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 3, pp. 1-9, 2018.

[8] L. N. Anh, T. N. Ha and H. X. Thanh, "A deep learning approach for malware detection using convolutional neural networks," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 2, pp. 354-360, 2018.

[9] D. S. Kim, H. S. Kim and J. H. Park, "A novel approach to malware detection using deep neural networks," *Journal of Intelligent & Fuzzy Systems*, vol. 35, no. 4, pp. 4485-4494, 2018.

[10] N. Ahmed, T. Ahmed, M. R. Islam and M. S. Hossain, "Malware detection using machine learning techniques: a survey," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 6, pp. 1817-1838, 2018.

[11] S. A. Ahmad, A. H. Abdullah and N. A. Basari, "A comparative study of malware detection using machine learning," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 10, no. 1-14, pp. 43-47, 2018.

[12] Jarrod Grasley, Ayman Diyab Alahmar, "Systematic Mapping of Machine Learning–Based Malware Detection Studies", *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET),* pp.1-10, 2022.

[13] A. A. Alazab, R. K. Sharma, S. H. Al-Smadi and S. Z. Al-Sharafat, "A novel framework for cybercriminals detection in online social networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 7, no. 4, pp. 546-558, 2019.

[14] H. T. Huong, T. N. Khoa and N. T. Tung, "A survey of machine learning techniques for malware detection," *Journal of Computer Science and Cybersecurity,* vol. 4, no. 1, pp. 1-12, 2020.

[15] J. Li, D. Huang, Y. Li and Q. Li, "Malware behavior analysis using machine learning techniques: a survey," *IEEE Access*, vol. 8, pp. 34623-34643, 2020.

[16] P. Dutta, A. Das and D. N. K. Jayakody, "Limitations of traditional approaches for malware detection and analysis and their solutions," *in Proceedings of the 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Bangalore, India, 2018, pp. 99-103.

[17] F. Saleem, S. S. Haque, M. A. Tahir and M. U. Ilyas, "Limitations of traditional intrusion detection systems for IoT networks: a comprehensive review*," IEEE Access*, vol. 8, pp. 26265-26281, 2020.

[18] T. Al-Duwairi, R. Jammal and H. Al-Sadi, "Network malware detection using deep learning: a review*," in Proceedings of the 2018 10th International Conference on Computer and Automation Engineering (ICCAE),* Brisbane, QLD, Australia, 2018, pp. 101-106.

[19] Han, Weijie & Xue, Jingfeng & Wang, Yong & Zhu, Shibing & Kong, Zixiao. (2019). Review: Build a Roadmap for Stepping Into the Field of Anti-Malware Research Smoothly. *IEEE Access*. PP. 1-1. https://doi.org/10.1109/ACCESS.2019.2945787

[20] Y. Nagano and R. Uda, "Static analysis with paragraph vector for malware detection," *Proc. 11th Int. Conf. Ubiquitous Inf. Manag. Commun. IMCOM 2017*, 2017, https://doi.org/10.1145/3022227.3022306

[21] M. Al-Badarneh, M. Jarrah, and A. Zeki, "A Dynamic Ensemble of Machine Learning Techniques for Malware Detection," *in IEEE Access,* vol. 6, pp. 22561-22575, 2018.

[22] L. Ma, Y. Huang, and Q. Zhu, "Malware detection based on machine learning algorithms and API sequences," *in IEEE Access, vol. 7, pp. 22915-22923, 2019.*

[23] S. H. Ali and S. M. Abdul-Kader, "An Efficient Machine Learning Approach for Malware Detection," *in IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 36-49, Jan. 2019.

[24] S. Khalid and J. Lloret, "Ensemble Learning for Malware Classification," in IEEE Access, vol. 7, pp. 116170-116184, 2019.

[25] M. Esmalifalak and M. R. Meybodi, "A machine learning approach for detecting malicious software behavior based on network traffic analysis," *in IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 993-1008, Apr. 2019.

[26] X. Zhang, W. Zhang, and Z. Liu, "Malware Detection Based on Machine Learning and Convolutional Neural Network," in IEEE Access, vol. 7, pp. 36526-36537, 2019.

[27] L. Wang, C. Chen, and Y. Chen, "Machine Learning-Based Malware Detection: A Comparative Study," in IEEE Access, vol. 8, pp. 22026-22038, 2020.

[28] X. Wei, Q. Xue, and Y. Qiao, "Detecting Advanced Persistent Threats Based on Machine Learning Techniques," in IEEE Access, vol. 8, pp. 197500-197516, 2020.

[29] F. Yang, Y. Wang, and J. Wang, "Dynamic and Static Hybrid Malware Detection Using Machine Learning," in IEEE Access, vol. 9, pp. 29018-29029, 2021.

[30] J. Huang, W. Shi, and G. Yan, "Integrating Static and Dynamic Features for Malware Detection via Machine Learning," in IEEE Access, vol. 10, pp. 18622-18633, 2022.

[31] M. Almukaynizi, M. Z. Reshi, and N. Alajlan, "FlowDL: A Deep Learning Approach for Malware Detection using Flow-based Traffic Analysis," in IEEE Access, vol. 7, pp. 93123-93134, 2019, https://doi.org/10.1109/ACCESS.2019.2923481

[32] Y. Jiang, X. Lin, and H. Li, "Malware Detection using Deep Learning with Static and Dynamic Feature Engineering," *in 2018 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pp. 103-107, 2018, https://doi.org/10.1109/SmartIoT.2018.00025

[33] Yuxin, D., & Siyi, Z. (2019). Malware detection based on deep learning algorithm. Neural Computing and Applications, 31(2), 461–472. https://doi.org/10.1007/s00521-017-3077-6

[34] Naseer, S., Saleem, Y., Khalid, S., Bashir, M. K., Han, J., Iqbal, M. M., & Han, K. (2018). Enhanced network anomaly detection based on deep neural networks. IEEE Access, 6, 48231–48246. https://doi.org/10.1109/ACCESS.2018.2863036

[35] Yin, C., Zhang, S., Wang, J., & Xiong, N. N. (2022). Anomaly Detection Based on Convolutional Recurrent Autoencoder for IoT Time Series. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 52(1), 112–122. https://doi.org/10.1109/TSMC.2020.2968516

[36] Cui, Z., Du, L., Wang, P., Cai, X., & Zhang, W. (2019). Malicious code detection based on CNNs and multi-objective algorithm. Journal of Parallel and Distributed Computing, 129, 50–58. https://doi.org/10.1016/j.jpdc.2019.03.010

[37] Xiao, F., Lin, Z., Sun, Y., & Ma, Y. (2019). Malware Detection Based on Deep Learning of Behavior Graphs. Mathematical Problems in Engineering, 2019. https://doi.org/10.1155/2019/8195395

[38] Li, S., Zhou, Q., Zhou, R., & Lv, Q. (2022). Intelligent malware detection based on graph convolutional network. Journal of Supercomputing, 78(3), 4182–4198. https://doi.org/10.1007/s11227-021-04020-y

[39] S. Kim, S. Kim, and H. Kim, "Deep Learning-Based Malware Detection Using Two-Dimensional Binary Program Features," in IEEE Access, vol. 6, pp. 38513-38522, 2018.

[40] R. Chauhan, S. Rai, S. S. Bhatia, and S. Singh, "Detecting Malware with Convolutional Neural Networks and Dynamic Analysis," in IEEE Access, vol. 7, pp. 60979-60989, 2019.

[41] J. Zhu, J. Wang, C. Zhang, and M. Liu, "A Malware Detection Method Based on Deep Learning and GAN," in IEEE Access, vol. 7, pp. 163933-163943, 2019.

[42] J. Huang, W. Yang, S. Xie, and Y. Zhang, "Malware Detection Based on Deep Learning and Static Analysis," in IEEE Access, vol. 7, pp. 82567-82574, 2019.

[43] S. Wang, Y. Zhu, W. Zhou, and H. Li, "Malware Detection Based on Deep Learning and Dynamic Analysis," in IEEE Access, vol. 8, pp. 34548-34555, 2020.

[44] A. Hasan, R. Islam, and A. H. M. Zahirul Alam, "Malware Detection Using Recurrent Neural Networks," in IEEE Access, vol. 8, pp. 34313-34320, 2020.

[45] M. Elhoseny, M. M. Hassanien, A. M. Salem, and E. El-Masry, "Deep Neural Networks for Botnet Detection," in IEEE Access, vol. 7, pp. 139648-139659, 2019.

[46] M. Atapour-Abarghouei and D. Stupples, "Deep Learning for Detecting Cryptographic Malware," in IEEE Access, vol. 7, pp. 183347-183357, 2019.

[47] J. Zhang, H. Liu, Z. Jiang, and H. Xie, "Deep Learning for Advanced Threat Detection," in IEEE Access, vol. 7, pp. 145157-145169, 2019.

[48] C. Zhao, W. Zhang, and H. Wang, "Malware Detection Based on Deep Learning with Feature Fusion," in IEEE Access, vol. 7, pp. 97620-97628, 2019.

[49] H. Zou, J. Li, J. Li, and M. Li, "A Novel Method for Detecting Malware Using a Deep Learning Approach with Hybrid Features," in IEEE Access, vol. 8, pp. 109239-109251, 2020.

[50] X. Zhang, L. Ding, X. Feng, and Y. Zhang, "Detecting Malicious Activities through Deep Learning in Industrial Control Systems," in IEEE Access, vol. 7, pp. 105316-105324, 2019.

[51] N. Elhadary, M. Tolba, A. S. Salem and S. B. Elsayed, "SDN-Based Malware Detection in Large-Scale Networks Using Multimodal Deep Learning," in IEEE Access, vol. 8, pp. 206250-206262, 2020, https://doi.org/10.1109/ACCESS.2020.3031378

[52] R. N. Nair, N. R. Prasad and P. V. N. Rao, "An SDN-based Solution for Efficient Malware Mitigation in Cloud Infrastructure," in 2018 IEEE 8th International Conference on Cloud Computing, pp. 1-8, 2018, https://doi.org/10.1109/CLOUD.2018.00010

[53] R. S. Ibrahim, S. F. Hassen, and A. L. Gomaa, "Malware Detection in Software-Defined Networks Using Ensemble Learning," in IEEE Access, vol. 8, pp. 147021-147033, 2020, https://doi.org/10.1109/ACCESS.2020.3014621

[54] S. K. Jana, S. Roy, and D. K. Bhattacharyya, "SDN-Based Malware Detection System for IoT Networks," *in 2020 IEEE International Conference on Communication and Signal Processing (ICCSP)*, Melmaruvathur, India, Apr. 2020, pp. 174-179.

[55] W. Zhang, Y. Li, and W. Zhang, "An SDN-Based Malware Detection System Using Machine Learning," IEEE Access, vol. 8, pp. 227-235, 2020

[56] Y. Song, J. Liu, and Q. Wu, "A Dynamic SDN-Based Malware Detection System for Cloud Computing Networks," *in 2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA),* Hefei, China, Jan. 2021, pp. 272-276.

[57] J. Kim, J. Jeon, and H. Kim, "SDN-Based Malware Detection and Mitigation Framework Using Flow Correlation Analysis," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 333-345, Mar. 2021.

[58] S. Sahoo, A. Misra, and J. Pradhan, "Malware Detection System Using Hybrid Machine Learning and SDN-Based Traffic Analysis," *in 2021 11th International Conference on Cloud Computing*, *Data Science & Engineering (Confluence)*, Noida, India, Mar. 2021, pp. 182-186.

[59] S. H. Kim, S. Y. Ahn, and K. W. Kim, "An Efficient SDN-Based Malware Detection System Using Machine Learning Techniques," *in 2021 IEEE Conference on Dependable and Secure Computing (DSC),* Honolulu, HI, USA, Aug. 2021, pp. 1-8.

[60] P. H. Nguyen, H. P. Nguyen and W. Zhou, "Using Software-Defined Networking for Intrusion Detection in IoT Networks," *in IEEE Internet of Things Journal,* vol. 7, no. 1, pp. 342-351, Jan. 2020, https://doi.org/10.1109/JIOT.2019.2931407

[61] M. N. Hassan and N. Noman, "A Software-Defined Networking (SDN) Approach for Network Security Management: A Review," in 2018 IEEE 5th Intl Conf on Soft Computing & Machine Intelligence (ISCMI), pp. 19-24, 2018, https://doi.org/10.1109/ISCMI.2018.8678621

[62] S. M. Tharwat and H. A. Tawfik, "Software-Defined Networking for Enhancing the Security of the Internet of Things," in IEEE Access, vol. 8, pp. 110547-110559, 2020, https://doi.org/10.1109/ACCESS.2020.3006422

[63] Y. Gao, X. Li, H. Peng, B. Fang, and P. S. Yu, "HinCTI: A Cyber Threat Intelligence Modeling and Identification System Based on Heterogeneous Information Network," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 2, pp. 708–722, 2022, https://doi.org/10.1109/TKDE.2020.2987019

[64] [1] J. Zhao, Q. Yan, X. Liu, B. Li, and G. Zuo, “Cyber threat intelligence modeling based on heterogeneous graph convolutional network,” RAID 2020 Proc. - 23rd Int. Symp. Res. Attacks, Intrusions Defenses, pp. 241–256, 2020.

[65] K. Oosthoek and C. Doerr, "Cyber Threat Intelligence: A Product Without a Process?" *in International Journal of Intelligence and CounterIntelligence,* pp. 1-16, 2020, https://doi.org/10.1080/08850607.2020.1780062

[66] J. Zhao, Q. Yan, J. Li, M. Shao, Z. He, and B. Li, "TIMiner: Automatically Extracting and Analyzing Categorized Cyber Threat Intelligence from Social Data," *Computers and Security*, vol. 95, 2020, https://doi.org/10.1016/j.cose.2020.101867

[67] [1] N. Rastogi, S. Dutta, M. J. Zaki, A. Gittens, and C. Aggarwal, “MALOnt: An Ontology for Malware Threat Intelligence,” Commun. Comput. Inf. Sci., vol. 1271 CCIS, pp. 28–44, 2020, https://doi.org/10.1007/978-3-030-59621-7_2

[68] A. Khodabakhsh, A. Azmoodeh, M. R. Meybodi, and A. Dehghantanha, "Detecting sophisticated cyber threats: A survey on cyber threat intelligence techniques," *Journal of Network and Computer Applications*, vol. 158, p. 102793, 2020.

[69] S. Li, Y. Li, S. Li, and S. Wu, "Ensemble of machine learning classifiers for cyber threat intelligence," *Journal of Ambient Intelligence and Humanized Computing,* vol. 11, no. 2, pp. 461–473, 2020.

[70] H. Han, J. Song, S. Kim, and J. Chung, "An effective cyber threat intelligence system based on deep learning and dynamic analysis," *Future Generation Computer Systems,* vol. 119, pp. 17–25, 2021.

[71] S. Saeed, S. S. Malik, and M. A. Jaffar, "A comprehensive review on machine learning based cyber threat intelligence," *Computers and Security*, vol. 106, p. 102323, 2021.

[72] A. M. M. Abad, H. M. S. Salleh, S. A. S. Suhaimi, and N. A. Zakaria, "Cyber threat intelligence: A survey of machine learning and deep learning techniques," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 9, pp. 8825–8841, 2021.