Khurram *et al. (IJECI)* 2023

(IJECI)
ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

International Journal for
Electronic Crime Investigation

DOI: https://doi.org/10.54692/ijeci.2023.0703157

Research Article

Vol. 7 issue 3 Jul-Sep 2023

Detection of Malicious software and control using Artificial Intelligence

Syed Khurram Hassan[1] and Muhammad Asif Ibrahim[2]

[1] Institute of Quality and Technology Management, University of the Punjab, Lahore, Pakistan.

[2] Department of Mathematics, FC College University, Lahore.

Corresponding author: khuramshah6515@gmail.com

Received: June 15, 2023; **Accepted**: August 17,2023; **Published**: September 20, 2023

Abstract:

Due to growing pervasiveness of malware in the contemporary digital environment, it is imperative to establish efficient identification and management systems to protect computer systems and networks. Conventional approaches to malware detection frequently encounter difficulties in keeping pace with the swiftly progressing characteristics of malevolent software. However, the emergence of artificial intelligence (AI) has unlocked fresh prospects for augmenting malware detection and control. This article investigates the implementation of AI methodologies in malware detection and mitigation, elucidating the benefits and obstacles connected with this strategy.

Keywords: Malware, virus, worm, trojan, ransomware, spyware.

1. Introduction

In the interconnected digital realm of today, cybersecurity threats are ever-evolving, posing substantial risks to computer systems, networks, and sensitive information. Among these threats, malicious software (malware) stands as a prominent adversary. Malware, commonly referred to as "malicious software," encompasses a diverse category of software programs intentionally designed to cause harm. This article serves as an introduction to malware, providing a comprehensive definition and exploring the various types of malware that exist.

1.1 Defining Malware

Malware comprises a broad range of harmful software programs, each with unique characteristics and objectives. Essentially, malware encompasses any software code or program developed with malicious intent to compro-

mise the integrity, availability, or confidentiality of computer systems and networks. Malware deviates from legitimate software as it aims to infiltrate, disrupt, or damage the target systems it infects [1].

1.2 Types of Malware

1.2.1 Viruses

One of the most renowned forms of malware is the computer virus. Analogous to their biological namesakes, computer viruses propagate and spread by attaching themselves to clean files or programs. Upon execution of an infected file or program, the virus activates and initiates a range of malicious activities, such as data corruption, system impairment, or unauthorized access. Human interaction, such as opening infected email attachments or downloading compromised files from the internet, often serves as the vehicle for virus propagation [2].

1.2.2 Worms

Worms, in contrast to viruses, possess self-replicating capabilities and can spread independently without requiring user interaction. These self-contained programs exploit security vulnerabilities in computer systems or networks, enabling them to propagate from one device to another. Unlike viruses, worms do not necessitate host file attachment. Instead, they replicate and disseminate themselves directly across networks, resulting in widespread damage and excessive consumption of network resources. Worms propagate rapidly, making them particularly perilous to large-scale networks [3].

1.2.3 Trojans

Trojans, also known as Trojan horses, represent deceptive malware programs that masquerade as legitimate software or files, deceiving users into executing them. Unlike viruses or worms, Trojans do not replicate autonomously. Instead, they grant unauthorized access to the user's computer, allowing attackers to engage in various malicious activities. Trojans may establish backdoors, pilfer sensitive information, or install additional malware onto the infected system. They commonly propagate through email attachments, malicious downloads, or compromised websites [4].

1.2.4 Ransomware

In recent years, ransomware has emerged as a prevalent and highly detrimental form of malware. This form of malicious software encrypts the files or entire system of a targeted individual, making them inaccessible until a ransom is provided to the person responsible. Ransomware frequently proliferates through email phishing campaigns, malicious downloads, or exploitation of software vulnerabilities. The financial motivation behind ransomware attacks has turned it into a lucrative tool for cybercriminals, targeting individuals, businesses, and even critical infrastructure [5].

1.2.5 Spyware

Spyware, a clandestine malware variant, covertly collects information about a user's activities without their knowledge or consent. It monitors online behavior, captures keystrokes, records browsing habits, and may even exfiltrate sensitive data such as login

credentials or financial information. Spyware typically operates stealthily, making its detection challenging. Distribution channels for spyware include malicious downloads, infected websites, or bundling with seemingly legitimate software [6].

1.2.6 Adware

While not as malicious as other forms of malware, adware is an intrusive software that inundates users with unwanted advertisements. Adware is often bundled with free software or downloads, with its primary purpose being revenue generation for the creators through targeted advertising. However, adware can consume system resources, impede computer performance, and compromise user privacy by collecting browsing habits and personal information [7].

Malware Type	Replication Method	Spread Without User Interaction	Objectives	Distribution Methods
Viruses	Attaches to files and programs	No	Data corruption, system damage, unauthorized access	Infected email attachments, compromised downloads
Worms	Self-replicating	Yes	Network propagation, resource consumption	Exploiting vulnerabilities, malicious links
Trojans	Disguised as legitimate software	Yes	Unauthorized access, information theft, system compromise	Email attachments, malicious downloads
Ransomware	Exploits Vulnerabilities	Yes	File Encryption, ransom demands	Email phishing, malicious downloads
Spyware	Stealthy	Yes	Unauthorized data collection, monitoring user activities	Infected websites, bundled software
Adware	N/A	Yes	Displaying unwanted ads	Bundled with free software downloads

Table 1: Differentiation of Malware

2 Ai-Based Malware Detection

2.1 IDS/IPS

Intrusion detection and prevention systems utilize an advanced mechanism to constantly monitor the network and detect potential security breaches. These systems maintain a log of pertinent information, address any issues that arise, and promptly alert security administrators. The functionalities of intrusion detection and prevention systems encompass various aspects, including sending notifications to administrators, discarding malicious packets, blocking undesirable network traffic originating from suspicious sources, terminating suspicious connections, and automatically adjusting configurations to counteract future intrusion attempts. There are numerous variations of intrusion detection and prevention, such as network intrusion prevention, host intrusion prevention, network behavior analysis, and wireless intrusion prevention, which can be employed for different applications [8].

2.2 Malware Analysis

The act of identifying and examining malicious software, unwanted entities, and their effects is commonly known as malware

analysis. This process involves uncovering indicators of compromise to detect infected machines, predict future attacks, assess their impacts, and identify compromised systems. Understanding the characteristics and objectives of a suspicious file plays a crucial role in malware detection, and this procedure is referred to as malware analysis. There are various approaches to conducting malware analysis, including static analysis, dynamic analysis, memory analysis, and hybrid analysis, which are used in different operating systems such as Windows, Linux, and Android [9].

Static scrutiny involves extracting static signatures or patterns from binary files without executing them. It is typically considered straightforward and efficient, but it struggles with analyzing obfuscated malware. On the other hand, dynamic malware examination allows malware to execute in an isolated environment, enabling the monitoring of its behaviors. This makes dynamic scrutiny resistant to syntactic obfuscation techniques [10]. However, dynamic scrutiny has limitations in tracking the behaviors of advanced malware like fileless malware. Another approach, memory examination, can reveal malicious behaviors associated with fileless malware [11].

2.3 Static Analysis

Static analysis is a technique used to examine malware without executing it. Its primary goal is to extract metadata from the malware. While static analysis is effective in identifying familiar malware, it faces limitations when dealing with complex and novel malware. Malware creators often employ obfuscation techniques to hide the true nature of their applications. Additionally, they use polymorphism and metamorphism techniques to modify the appearance of the code across different malware samples. Analyzing intricate malware using advanced static analysis approaches is a time-consuming process that requires extensive expertise in operating systems and disassembly. For example, PE Explorer is a tool commonly used to inspect Windows .exe and .dll files. Androguard is a well-known static analysis tool for analyzing Android applications, which facilitates the comparison of code similarities between two applications. By comparing the codes of two applications, Androguard can determine which methods are identical, similar, or present in one but absent in the other [12].

2.4 Dynamic Analysis

Dynamic analysis involves the examination of malware behavior and consequences upon execution. Understanding the actions performed by malware during execution is crucial. The primary goal is to collect real-time information about the behavior of malware and its impact on the system. This approach allows for the comprehensive observation of the malware's functionality and its influence on the surrounding environment during execution. Typically, the file is executed within a virtualized environment. Dynamic analysis is preferred over static analysis because it can detect malware easily, even if the malware's structure undergoes changes, as its behavior and characteristics remain constant. Wireshark

and TCP dump are useful tools for capturing and analyzing network packets. DroidBox is a sandbox tool specifically designed for Android applications. It logs an application's network communications, file accesses, launched services, loaded classes, cryptographic operations using the Android API, messages, and outgoing calls during execution [13].

2.5 Malware Detection Based on Signature

A digital fingerprint is a unique code injected into application software by malicious software creators, serving as a distinguishing identifier for harmful software. It is an efficient and swift method for detecting known malware. However, this technique has limitations when it comes to unfamiliar attacks. It is ineffective in identifying novel and unrecognized malicious software because there is no distinct digital fingerprint available for such attacks. Furthermore, malware developers can constantly modify their code or packaging methods to evade creating an identical digital fingerprint to previous versions, thus circumventing detection [14].

2.6 Malware Detection Based on Behavior

In this method, software behavior is utilized to determine whether it is harmful or benign. A sensor that focuses on behavior goes through three distinct stages:

a. data collection, which involves gathering information about the malware,
b. analysis of the collected data to extract the most relevant details and create a behavioral model or profile,
c. the identification phase which entails finding a correlation between the malware's profile and the one that represents malicious behavior [15].

2.7 Machine and Deep Learning

"In recent years, the fields of research have witnessed significant advancements attributed to the progress made in machine learning (ML) and deep learning (DL). Artificial intelligence has experienced remarkable growth, largely thanks to the contributions of ML and DL. ML, a captivating domain of computer science, has found successful applications in information retrieval, pattern recognition, and decision-making [16].

DL [17], on the other hand, relies on robust and versatile models that facilitate the extraction of relevant information for complex tasks. In this regard, DL holds great potential for the identification, categorization, and analysis of malicious software, as well as the recognition and detection of botnets. It also aids in mitigating cyber attacks, preventing intrusions, responding to incidents, analyzing network traffic, detecting advanced persistent threats (APTs), identifying cybercriminals, conducting thorough packet inspection, and performing analytics for cycybersecurity.

3 DI-based Malware Detection Models In Windows Platform

Jeon and Moon [18] developed an advanced deep learning-based technique for malware detection. Their approach combined static opcode sequences with dynamic recurrent neural networks (RNN) and convolutional recurrent neural networks. To condense

lengthy opcode sequences into concise ones, they employed a convolutional autoencoder. This allowed the recurrent neural network to utilize the opcode features generated by the autoencoder for malware classification. The opcode characteristics were extracted statically from executable files in the Windows environment. The performance of their approach was impressive, achieving a detection accuracy of 96% and a true positive rate of 95%.

To further improve malware detection, Yuan et al. [19] proposed a novel model called MDMC. This model leveraged Markov images and convolutional neural networks (CNN) to identify malware attacks. By utilizing a bytes transfer probability matrix, binary files were transformed into Markov images. The CNN played a crucial role in automatic feature engineering and classification. The experiments were conducted on a Microsoft dataset consisting of 10,868 malware samples, covering nine malware families. The results demonstrated the superior performance of MDMC, achieving an accuracy of 99.264%.

4 DI-based Malware Detection Models In Android Platform

Pektas and Acarman [20] conducted a study on Android application analysis, where they utilized a dataset comprising 25,000 benign and 24,650 malicious applications. Their objective was to develop a deep learning-based model for automatic identification and classification of Android applications. Initially, they employed a Convolutional Neural Network (CNN) to extract relevant features from the applications. The extracted features were then passed to a Long Short-Term Memory (LSTM) module to capture intricate relationships among them. The LSTM module generated a final feature set, which was subsequently input into a dense layer or fully connected neural network for classification. To optimize the hyperparameters of the network, the researchers utilized the grid search approach and implemented the model using TensorFlow and Keras frameworks. Their DL based approach achieved an impressive accuracy of 91.42% in identifying unknown Android malicious applications.

Ma et al. [21] proposed a framework called Droidect for classifying malicious Android applications on Android devices. The framework was based on a Bidirectional LSTM (Bi-LSTM) model. The researchers extracted behavioral features from API call sequences of APK files and applied an NLP-based semantic localization technique to construct dense vectors. These vectors were then fed into the Bi-LSTM model for classification. The evaluation of the framework was performed on a dataset comprising 11,982 benign files and 9,616 malicious files, demonstrating an accuracy of 97.22% for malware detection.

In addition, another study implemented an LSTM-based approach to detect malware in Android applications. The researchers obtained opcode sequences from benign and malware applications, sourced from Play Store and VirusShare respectively. Text processing techniques were employed to preprocess the opcodes, and a LSTM-based malware detector was constructed using Keras. This approach achieved a detection accuracy of 96%. Further-

more, the researchers explored various deep learning-based detectors using LSTM, GRU, Bi-LSTM, and stacked LSTM/GRU models. They utilized permissions, API call sequences, intent sequences, and intent filters for implementing these models [22].

5 DI-based Malware Detection Models In Linux Platform

Xu et al. [23] proposed HawkEye, a system designed for identifying malware attacks in Linux. The system utilizes control flow graphs (CFGs) and employs graph neural networks (GNNs) in combination with a multilayer perceptron-based classifier. To gather malware samples, the researchers accessed the AndroZoo and VirusShare repositories, while benign samples were obtained from executable files and libraries in a clean Ubuntu installation. To represent the structural information of both malware and benign executables, graph sets were defined using a CFG extractor. The CFG extractor captured details such as basic block addresses and assembly instruction/opcodes. These extracted features were then processed by the GNN module to generate graph embedding features, which were subsequently fed into the MLP classification module. The evaluation of the system showcased a remarkable detection accuracy of 96.82% when identifying malware in a Linux environment.

6 Types Of Ai Based Malware Detection

In order to develop a comprehensive comprehension of the diverse AI-based methodologies utilized for malware detection, it is crucial to grasp the fundamental concept of malware itself and its operational mechanisms. Malware, an abbreviation for malicious software, encompasses software that is purposefully crafted to inflict harm or render computer systems inoperable. This category of malicious software encompasses various types such as viruses, worms, Trojans, spyware, and adware. Given the substantial negative consequences caused by malware, extensive research has been devoted to analyzing and countering these malevolent programs. With the recent advancements in Artificial Intelligence (AI), cybersecurity experts have increasingly turned their focus to utilizing Machine Learning (ML) and Deep Learning (DL) techniques to enhance the identification and categorization of malicious files [24].

Recent research findings suggest that individuals with limited experience often encounter difficulties when differentiating between benign and malicious applications. This underscores the importance of designing computer systems and mobile applications that possess the capability to detect malicious activities and safeguard all stakeholders involved. A plethora of algorithms has been developed to identify malware activities, leveraging cutting-edge concepts such as Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) [25]. In the realm of cybersecurity, AI has gained substantial prominence, particularly in the field of malware detection, where AI-based approaches are increasingly prevalent. Malware analysis forms the bedrock of effective malware detection techniques and is imperative in understanding the classification

and functionality of malicious files [26].

Machine learning has gained popularity as an AI-based technique for malware detection. By analyzing patterns in datasets, machine learning algorithms can effectively identify malware, even if it's a previously unseen type. However, it is important to note that machine learning can be computationally demanding and requires a significant amount of training data to optimize the algorithms. On the other hand, deep learning (DL) has shown effectiveness in detecting sophisticated malware that constantly evolves [27].

6.1 Methods

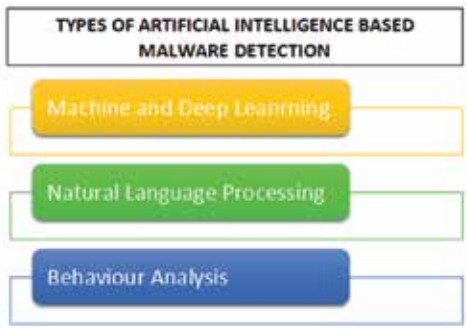

Fig 1: Types of AI based Malware Detection

6.1.1 Machine and Deep Learning

There is a diverse range of AI-based malware detection methods available, each with its own set of advantages and disadvantages. Among the most common methods are machine learning and deep learning. Extensive research has been conducted on utilizing deep learning algorithms for malware detection. Machine learning is an AI technique that can learn from data and improve its performance over time. It finds applications in areas such as spam filtering and fraud detection. On the other hand, deep learning is a more advanced form of machine learning that can learn from data in a manner resembling human learning. It is commonly used for tasks like image recognition and natural language processing. Both machine learning and deep learning have distinct strengths and weaknesses in the context of malware detection. However, it is worth noting that while deep learning approaches have demonstrated impressive results in malware detection, several of these studies have limitations due to either [28].

6.1.2 Natural Language Processing

Natural Language Processing (NLP) is a widely used AI-based method for detecting malware in the field of cybersecurity. NLP focuses on enabling computers to understand and process human language, ranging from basic word recognition to complex tasks like sentence comprehension and information extraction. In the context of malware detection, NLP offers several advantages. It can identify specific keywords and patterns commonly associated with malicious software [29]. For example, instructions to "delete all files" or "format the hard drive" are indicative of potential malware. NLP can also analyze sentence structures to identify suspicious commands such as "download this file" or "install this program." NLP is a powerful tool for detecting various forms of malware. However, it is essential to acknowledge that no single method is foolproof. To achieve optimal effectiveness, NLP should be combined with other malware detection techniques. By integrating NLP with

complementary approaches, security researchers can enhance the overall accuracy and efficacy of malware detection systems. It is important to adopt a multi-faceted approach to ensure comprehensive protection against evolving and sophisticated malware threats [30].

6.1.3 Behaviour Analysis

Email attachments are a common avenue for the distribution of malware, as opening an infected attachment can trigger the execution of malicious code and compromise a computer. Once infected, malware can carry out various nefarious activities, such as file deletion, data theft, or even taking control of the entire system. To combat this threat, many organizations have turned to AI-based malware detection methods. These methods utilize artificial intelligence algorithms to analyze email attachments and other files, with the aim of identifying potential threats. By doing so, they can proactively block emails containing suspicious attachments from reaching users' inboxes. AI-based malware detection methods employ various approaches. Some methods focus on analyzing the content of files to identify potentially malicious code. This involves examining the structure and code of the file for known patterns or indicators of malware. Other methods concentrate on observing program behavior to detect signs of malicious intent. By monitoring program execution and analyzing actions such as file modifications or network communications, these methods can flag potentially malicious programs. Some approaches combine both content-based and behavior-based analyses to achieve improved accuracy and comprehensive coverage. The adoption of AI-based malware detection methods in email security strengthens protection against threats stemming from attachments. By harnessing the capabilities of artificial intelligence, organizations can fortify their defenses by proactively identifying and blocking potentially malicious attachments. This helps safeguard users and systems from the risks associated with malware [31].

6.2 Issues and Challenges

The shortcomings of traditional methods in malware detection and analysis have prompted researchers to seek alternative technologies that can achieve real-time detection with high accuracy and a reduced false positive rate. Earlier approaches relied on statistical analysis of system changes or employed probabilistic methods that looked for specific literals to classify executable as malware. However, these probabilistic and statistical techniques provided only approximate assessments based on a limited set of malware features and encountered challenges when dealing with obfuscated malware [32].

a. The utilization of packed executables and the presence of small datasets can introduce uncertainties in the outcomes when deploying a real-time malware detection solution. These factors can significantly affect the effectiveness and accuracy of the implemented solution, underscoring the importance of developing robust techniques capable of handling such scenarios.

b. When employing a framework that relies on Windows audit logs, the effectiveness of the solution can be compromised by obfuscation techniques. In such instances, the approaches outlined in research papers may not be directly applicable or may require modifications to address the challenges posed by obfuscation. This adaptability is crucial to ensure the solution remains reliable and useful in detecting malware.

The field of malware analysis has witnessed the rise of deep learning as a prominent approach, primarily because of its capacity for automatic feature engineering. However, despite its advancements, there remain certain unresolved issues that demand attention. Notably, deep learning-based methods encounter difficulties when confronted with limited availability of data. This constraint calls for further exploration and research in various critical domains [33].

Addressing these challenges requires ongoing research and development in the field of AI-based malware detection, as well as collaboration between cybersecurity experts, AI researchers, and the wider security community to improve the effectiveness and reliability of these systems. Some suggestions that this paper gives for the betterment in the realm of Artificial Intelligence based Malware Detection are as follows [34]:

i. Comprehensive and Diverse Training Data: Acquiring a wide range of labeled training data that covers various malware types, families, and variants is crucial. It is important to continuously update the training data to account for emerging threats and the evolving landscape of malware.

ii. Integration of Behavioral Analysis: In addition to static file analysis, incorporating behavioral analysis techniques is valuable. By examining the dynamic behavior of programs and identifying anomalies, it becomes possible to detect malicious activities, even in the absence of known malware signatures.

iii. Ensemble Models: Utilizing ensemble models that combine predictions from multiple AI algorithms or models can enhance detection accuracy. Ensemble methods leverage the strengths of different models while mitigating individual model weaknesses, leading to improved overall performance.

iv. Adversarial Training and Testing: Training AI models using adversarial samples can enhance their resilience against adversarial attacks. By exposing models to manipulated or modified malware samples during training, they can learn to detect and counteract adversarial attempts to evade detection.

v. Continuous Learning and Updates: Implementing mechanisms for ongoing learning and updates is vital to keep AI models current with emerging threats. Regularly retraining models using new data and periodically updating detection algorithms and techniques will enable systems to stay

ahead of evolving malware tactics.

7 Traditional Methods

7.1 Signature-Based Detection

Signature-based detection is one of the most prevalent techniques used in traditional antimalware systems. It relies on the identification of known malware patterns, referred to as signatures, which are stored in databases. When a file is scanned, its content is compared against these signatures. If a match is found, the file is flagged as malware and appropriate actions are taken [35].

The strengths of signature-based detection lie in its effectiveness against known malware strains and its efficiency in rapidly identifying threats. By leveraging a vast database of signatures, it can quickly identify and quarantine files that exhibit known malicious patterns. This approach has been refined over the years, enabling anti-malware software to keep pace with the constantly evolving threat landscape. However, its primary limitation is the inability to detect novel or modified malware that does not match existing signatures. Attackers can easily evade signature-based detection through techniques like polymorphism or code obfuscation, which alter the characteristics of malware without changing its underlying functionality. To mitigate these limitations, heuristics and behavioural analysis techniques are often combined with signature-based detection to enhance the detection capabilities of anti-malware solutions.

7.2 Behavioural Analysis

Behavioural analysis focuses on observing and monitoring the behaviour of files, programs, or processes to identify potential malware. This approach aims to detect malicious activity by examining actions such as system modifications, unauthorized network communications, or attempts to exploit vulnerabilities.The advantage of behavioural analysis is its capability to detect previously unknown malware or variants that have undergone modification. By analysing the behaviour of software, it can identify suspicious activities and anomalies indicative of malware. For example, if a program attempts to modify critical system files or establish connections with suspicious external servers, it raises red flags and triggers appropriate response measures. However, this technique may generate false positives due to legitimate software exhibiting similar behaviour or false negatives if malware remains dormant during analysis. Additionally, behavioural analysis can be resource-intensive, requiring the continuous monitoring of system activities and the establishment of baselines for normal behaviour. To address these challenges, machine learning algorithms and anomaly detection techniques are often employed in behavioural analysis. These approaches leverage historical data and behavioural patterns to identify deviations from normal activities, enhancing the accuracy of malware detection [36].

8 Limitations Of Artificial Intelligence In Malware Detection

Due to the rise in malware activity brought on by the quick development of technology, security has grown to be a significant concern and now threatens the safety and security of

both computer systems and stakeholders. One of the most urgent concerns is safeguarding the data from fraudulent attempts in order to preserve stakeholder security, notably that of end users. Malware is a collection of harmful programming code, scripts, active content, or intrusive software that is meant to damage legitimate computer programmers, mobile, or web apps [37].

A study found that novice users can't tell the difference between perilous and trustworthy programmer. Therefore, malicious activity detection should be built into computer systems and mobile applications to safeguard stakeholders. There are several techniques to identify malware activity that make use of cutting-edge ideas like artificial intelligence, machine learning, and deep learning. In this study, we focus on AI-based strategies for identifying and thwarting malware activity [38].

Cyberattacks are increasingly using machine learning (ML) and artificial intelligence (AI) techniques. AI aids in the creation of hidden channels and the malware's concealment. AI also facilitates difficult-to-detect cyber-physical sabotage and new varieties of phishing attempts. Malware developers are increasingly using AI and ML techniques to enhance the effectiveness of their attacks [39].

Defenders must consequently prepare for unusual malware with cutting-edge, evolving features and functionalities. The ability of AI to automate difficult operations poses a difficulty in the face of the defensive application of anti-malware AI techniques. This article reviews the current state of evasion and attack methods used by AI-enhanced malware against AI-supported defense systems [40].

8.1 AI-Based Real-Time Malware Detection in Data Centers

To enhance security in Data Centers (DCs) and Smart Cities (SCs), an AI-powered edge computing approach called pAElla has been developed. pAElla utilizes real-time malware detection (MD) on an IoT-based monitoring system for DCs/SCs. By analyzing power measurements' spectral density and employing autoencoders, pAElla achieves promising results with a high F1-score and low false alarm and malware miss rates [41, 42].

8.2 False Positive/Negative Rate

There may be some similarities between the fingerprints and characteristics of malicious files and samples. False positive and false negative rates are a problem for a number of malware detection methods. However, an expansion in misleading positive or bogus negative rates diminishes the model recognition precision, misleading up-sides are definitely huger than misleading negatives in the powerful malware identification models. On a user's computer, if a legitimate file is mistakenly identified as malicious, the operating system and other applications may cease to function [43].

8.3 Insufficient Training Data

AI models, particularly those based on machine learning, heavily rely on extensive and diverse training datasets to learn patterns and make accurate predictions. However,

acquiring comprehensive and up-to-date labeled training data for malware detection presents a challenge. The constantly evolving nature of malware makes it difficult to maintain training datasets that are current and representative of the wide range of threats [44].

To mitigate this limitation, researchers are exploring techniques such as data augmentation, transfer learning, and active learning [45]. Data augmentation involves generating synthetic malware samples or manipulating existing samples to expand the training set. Transfer learning leverages knowledge from pre-trained models on related tasks to enhance the performance of malware detection models. Active learning methods prioritize the selection of the most informative samples for manual labeling, optimizing the use of limited resources and improving the training dataset [46, 47].

To overcome this limitation, AI-based malware detection systems are often supplemented with other techniques such as behavior monitoring, heuristics, and anomaly detection. These methods focus on identifying suspicious behaviors or deviations from expected patterns, providing an additional layer of defense against zero-day attacks. Collaboration and information sharing among security communities are also crucial in rapidly detecting and responding to emerging threats [48].

8.4 Interpretability and Explainability

Many AI models, especially deep learning models, are often considered black boxes, lacking interpretability and explain ability. These models operate through complex mathematical computations, making it challenging to understand the underlying reasons behind their decisions. In the context of malware detection, this lack of transparency can be problematic, as it becomes difficult for security analysts to trust and validate the alerts generated by AI models [49].

Researchers are actively working on developing methods for interpreting and explaining AI models' decisions in the context of malware detection. Techniques such as model-agnostic methods, attention mechanisms, and rule extraction algorithms aim to provide insights into the decision-making process of the model. By gaining understanding of the factors and features that contribute to the model's predictions, security analysts can gain more confidence in the alerts generated by AI-based malware detection systems [50].

8.5 Resource Requirements

Some AI models used in malware detection, particularly deep learning models, can be computationally expensive and demand significant computational resources. Deploying such resource-intensive models on devices or networks with limited resources can pose challenges, impacting the performance and scalability of the malware detection system [51].

To overcome this limitation, researchers explore techniques such as model compression, pruning, and hardware acceleration. Model compression methods aim to reduce the size and complexity of the AI model without

significantly compromising performance. Pruning techniques remove unnecessary connections or parameters from the model, reducing computational requirements. Hardware acceleration, such as utilizing specialized hardware like GPUs or dedicated AI accelerators, can expedite the inference process and improve efficiency [52].

By optimizing resource utilization, researchers strive to make AI-based malware detection systems more accessible and practical for deployment across various platforms and environments [53].

9 REFERENCES

[1] Tahir, R. A study on malware and malware detection techniques. International Journal of Education and Management Engineering, vol. 8, no. 2, 2018.

[2] Molina-Coronado, Borja; Mori, Usue; Mendiburu, Alexander; Miguel-Alonso, Jose (1 January 2023). "Towards a fair comparison and realistic evaluation framework of android malware detectors based on static analysis and machine learning". Computers & Security. vol. 8, no. 1, 2018.

[3] Peter Szor. The Art of Computer Virus Research and Defense. Pearson Education. p. 204. 2005.

[4] Landwehr, C. E; A. R Bull; J. P McDermott; W. S Choi. A taxonomy of computer program security flaws. DTIC Document. 2012.

[5] Richardson, Ronny; North, Max "Ransomware: Evolution, Mitigation and Prevention". International Management Review. 13 (1): 10–21. 2019.

[6] Russinovich, Mark "Sony, Rootkits and Digital Rights Management Gone Too Far". Mark's Blog. Microsoft MSDN. 2009.

[7] Casey, Henry T. "Latest adware disables antivirus software". Tom's Guide. Yahoo.com. Archived from the original on 27 November 2015. Retrieved 25 November 2015.

[8] M. Korcak, J. Lámer, F. Jakab, Intrusion Prevention/Intrusion Detection System (IPS/IDS) for Wifi Networks. International journal of Computer Networks & Communications. 2016.

[9] P. Maniriho, A. N. Mahmood, M. J. M. Chowdhury, A study on malicious software behaviour analysis and detection techniques: Taxonomy,current trends and challenges, Future Generation Computer Systems 130. Pp. 1–18. 2022.

[10] F. Biondi, T. Given-Wilson, A. Legay, C. Puodzius, J. Quilbeuf, Tutorial: An overview of malw Leveraging Applications of Formal Methods,Springer, pp. 565–586. 2018.

[11] R. Sihwail, K. Omar, K. A. Z. Ariffin,

An effective memory analysis for malware detection and classification, CMC-Computers Materials &Continua. Vol. 67, no. 2. pp 2301–2320. 2021.

[12] R. Jusoh, A. Firdaus, S. Anwar, M. Z. Osman, M. F. Darmawan, M. Faisal, Malware Detection Using Static Analysis in Android: a review of FeCO (Features, Classification, and Obfuscation). 2021

[13] O. Or-Meir, N. Nissim, Y. Elovici, L. Rokach, Dynamic Malware Analysis in the Modern Era—A State of the Art Survey. 2005.

[14] A. M. Abiola, M. F. Marhusin, Signature-Based Malware Detection Using Sequences of N-grams. International Journal of Engineering and Technology (UAE). 2004.

[15] H. S. Galal, Behavior-based features model for malware detection. Journal of Computer Virology and Hacking TecTechniques. 2004.

[16] Bernardi, L.; Mavridis, T.; Estevez, P. 150 successful machine learning models: 6 lessons learned at booking.com. roceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, New York, NY, USA, 4–8 August 2019.

[17] Zhu, D.; Jin, H.; Yang, Y.; Wu, D.; Chen, W. DeepFlow: Deep learning-based malware detection by mining Android application for abnormal usage of sensitive data. In Proceedings of the IEEE Symposium on Computers and Communications, Heraklion, Greece, 3–6 July 2017.

[18] S. Jeon, J. Moon, Malware-detection method with a convolutional recurrent neural network using opcode sequences, Information Sciences. Vol. 535. pp 1–15. 2020.

[19] B. Yuan, J. Wang, D. Liu, W. Guo, P. Wu, X. Bao, Byte-level malware classification based on markov images and deep learning, Computers &Security. Vol. 92. 2020.

[20] A. Pektas¸, T. Acarman, Learning to detect android malware via opcode sequences, Neurocomputing. Vol. 396. pp 599–608. 2020.

[21] Z. Ma, H. Ge, Z. Wang, Y. Liu, X. Liu, Droidetec: Android malware detection and malicious code localization through deep learning, arXiv preprint arXiv: 2002.

[22] R. Feng, J. Q. Lim, S. Chen, S.-W. Lin, Y. Liu, Seqmobile: An efficient sequence-based malware detection system using rnn on mobile devices, in: 2020 25th International Conference on Engineering of Complex Computer Systems (ICECCS), IEEE, pp. 63–72. 2020

[23] P. Xu, Y. Zhang, C. Eckert, A. Zarras, Hawkeye: cross-platform malware detection with representation learning on graphs, in: International Conference on Artificial Neural Networks, Springer, pp. 127–138. 2021.

[24] Marais B., Quertier T., Morucci S., AI-based Malware and Ransomware Detection Models. arXiv:2207.02108 [cs.CR], 2022.

[25] James O., Employing Artificial Intelligence Techniques for the Detection and Prevention of Malware, 2022.

[26] Wolsey A., The State-of-the-Art in AI-Based Malware Detection Techniques: A Review. arXiv:2210.11239 [cs.CR], 2022.

[27] Austin B., Maanak G., Senior M., IEEE, Mahmoud A., Automated Machine Learning for Deep Learning Based on Malware Detection, 2023.

[28] Umm-e-Hani T., Faiza B., Muhammad H., Asifullah K., Yeon S., A Survey on the Recent Trends in Deep Learning Based Malware Detection, 2022

[29] A. Vaswani et al., "Attention Is All You Need," in Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, pp. 5998-6008. 2017. Available: https://arxiv.org/abs/1706.03762

[30] J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), Minneapolis, MN, USA, pp. 4171-4186. 2019. Available: https://arxiv.org/abs/1810.04805

[31] V. Ramanathan et al., "Deep Behavior Mining: Discovering Unusual Crowd Activities in Videos," in IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol. 36, no. 5, pp. 898-910, 2014.

[32] A. Faitouri, Z. Anazaida, A. Fuad, A. Bander, A. Taiseer, and A. Asma, "Malware Detection Issues, Challenges, and Future Directions: A Survey," 2022.

[33] M. J. Hossain Faruk et al., "Malware detection and prevention using artificial intelligence techniques," in 2021 IEEE International Conference on Big Data (Big Data), IEEE, 2021.

[34] A. Libri, A. Bartolini, and L. Benini, "pAElla: Edge AI-based real-time malware detection in data centers," IEEE Internet of Things Journal, vol. 7, no. 10, pp. 9589-9599, 2020.

[35] Christodorescu, M., Jha, S., & Song, D. Malware detection using behavioral analysis. ACM Conference on Computer and Communications Security (CCS),

Alexandria, VA, USA. 2002.

[36] Szor, P. The Art of Computer Virus Research and Defense. Addison-Wesley Professional. 2001.

[37] A. Mishra, Y. I. Alzoubi, M. J. Anwar, and A. Q. Gill, "Attributes impacting cybersecurity policy development: An evidence from seven nations," in *Proceedings of the IEEE International Conference on Big Data (Big Data)*, pp. 2982-2987. 2021.

[38] M. J. Hossain et al., "Malware detection and prevention using artificial intelligence techniques," in *Proceedings of the 2021 IEEE International Conference on Big Data (Big Data)*, pp. 4845-4850. 2021.

[39] R. Faruk et al., "Detection of cyber attacks using machine learning," in *AIP Conference Proceedings*, vol. 2405, no. 1, 2022.

[40] A. Djenna et al., "Artificial Intelligence-Based Malware Detection, Analysis, and Mitigation," in *Proceedings of the International Conference on Information Science and Systems (ICISS)*, 2022.

[41] A. Libri, A. Bartolini, and L. Benini, "pAElla: Edge AI-based real-time malware detection in data centers," in *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9589-9599, 2020.

[42] N. Muchammad et al., "Malware Detection: Issues and Challenges," in *Journal of Physics: Conference Series*, vol. 1807, no. 1, p. 12-31. 2021.

[43] A. Ribeiro, S. Singh, and C. Guestrin, ""Why should I trust you?" Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135-1144. 2016.

[44] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," arXiv preprint arXiv:1605.07277, 2016.

[45] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," arXiv preprint arXiv:1510.00149, 2015.

[46] C. Zhang et al., "Optimizing convolutional neural networks for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 1-9. 2015.

[47] A. M. Saxe et al., "On the information bottleneck theory of deep learning," in *Journal of Statistical Physics*, vol. 177, no. 3-4, pp. 718-751, 2019.

[48] C. Kolias et al., "DDoS in the IoT: Mirai

and other botnets," in *Computer*, vol. 50, no. 7, pp. 80-84, 2017.

[49] K. Rieck et al., "Learning and classification of malware behavior," in *Journal of Machine Learning Research*, vol. 9, pp. 2721-2764. 2008.

[50] M. Guo, G. Wang, H. Hata, and M. A. Babar, "Revenue maximizing markets for zero-day exploits," in *Autonomous Agents and Multi-Agent Systems*, vol. 35, no. 2, pp. 36-51, 2021.

[51] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why should I trust you?" Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135-1144. 2016.

[52] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," arXiv preprint arXiv:1605.07277, 2016.

[53] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," arXiv preprint arXiv:1510.00149, 2015.