Mesopotamian journal of Cybersecurity
Vol.5,No.1, **pp**. 273–300
DOI: https://doi.org/10.58496/MJCS/2025/018; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/cybersecurity

Review Article

Learning Techniques-Based Malware Detection: A Comprehensive Review

Sarah Fouad Ali[1,*], Musaab Riyadh Abdulrazzaq[1], Methaq Talib Gaata[1]

[1] *Department of Computer Science, Collage of Science, Mustansiriyah University, Baghdad, Iraq.*

ARTICLEINFO

Article History
Received 18 Jan 2025
Accepted 20 Mar 2025
Published 08 Apr 2025

Keywords
Cybersecurity
Malware detection
Internet of Things (IoT)
Machine Learning (ML)
Deep Learning (DL)
Federated Learning (FL)

ABSTRACT

The rapid proliferation of Internet of Things (IoT) devices has significantly increased the threat landscape, with malwares arising as a critical concern. Advanced learning methods such as machine learning (ML), deep learning (DL), and federated learning (FL) are essential for handling complex IoT data. ML provides tools for pattern identification and detecting anomalies. DL boosts malware detection by automatically extracting features and identifying patterns. FL enables collaborative model training across decentralized devices, ensuring data privacy, which is crucial for diverse IoT systems. This comprehensive review specifically synthesizes ML, DL and FL for malware detection in the IoT environment, highlighting key trends and developments. Additionally, several significant contributions have been provided, including an overview of various types of malwares and their approaches and a comparison with existing studies. Importantly, notable trends and advancements are highlighted, and the current limitations of these learning techniques in malware detection are identified. It concludes by outlining future research directions to develop robust, scalable malware detection mechanisms tailored to safeguard the prosperity of the IoT environment against evolving cyber threats.

1. INTRODUCTION

The rapid expansion of Internet of Things (IoT) devices has resulted in an increase in risks arising from distributed denial of service attacks (DDoS). Additionally, the vast scope and diversity of IoT ecologies present distinctive security obstacles, necessitating specialized and intelligent measures to safeguard essential infrastructure [1]. Moreover, network attacks in the form of DDoS are implemented with the intention of interfering with server responses and functionalities, which is widely common due to its simplicity in establishment and the difficulty it presents in terms of identification[2]. Owing to the heterogonous devises of the IoT as a smart home and the significant advancements in the home appliance sector, there has been a notable surge in demand for these products. The users are faced with many choices when selecting electrical appliances, a situation that can potentially lead to confusion. Consequently, recommendation systems have experienced widespread adoption across various digital domains. These automated recommendation systems, which utilize diverse methodologies, play a crucial role in suggesting suitable devices to users [3]. Moreover, these devices have led to a significant rise in streaming data. Among these data, certain elements exhibit characteristics of malicious software intent that evade detection by conventional security measures such as firewalls and anti-maliciousness software [4].

Cybersecurity encompasses an array of technological solutions and procedural frameworks that are meticulously engineered to safeguard networks and computing systems from various forms of assault and data expropriation, while simultaneously inhibiting unauthorized access through the establishment of comprehensive protocols aimed at achieving absolute defence against cybercriminal activities[5][6].

The structure of the IoT is composed of hardware, software, sensors, and communication components. Communication among these devices occurs through networks or the internet, facilitating data sharing. These devices vary from complex machinery across different sectors to basic devices. By collecting data from their environment, IoT devices observe and examine the physical world. These data can subsequently undergo processing for additional application or task automation [7].

*Corresponding author. Email: sarah.fuad@uomustansiriyah.edu.iq

The significance of malware detection is paramount, especially considering the substantial expenses associated with malware attacks. These attacks must be identified and denied effectively within the realm of network security, particularly in the context of domain name system (DNS) services [8].

The term malicious software poses a significant challenge within the realm of cybersecurity, which is constantly evolving to evade traditional detection methods based on signatures. Specifically, the realm of ML, particularly DL, displays promise as a means of identifying malware. DL revolves around the acquisition of multitiered data representations, with the upper tiers encapsulating more abstract notions. As a result, DL models can grasp highly intricate functions directly from unprocessed data without the need for extensive feature manipulation. This approach has proven remarkably efficient in discerning patterns within datasets of significant complexity and finds application across numerous fields. Nevertheless, conventional ML encounters difficulties when handling intricate, unprocessed data [9]. Malware denotes malicious software such as “viruses, Trojans, and spyware" engineered to infiltrate or harm computer systems. Malware creators employ tactics such as obfuscation to elude detection by antivirus software reliant on fixed pattern recognition. Nonetheless, various forms of malware often exhibit shared fundamental behaviours that can potentially be pinpointed through ML techniques, even when the code structure differs [9],[10].

Automated ML (Auto-ML) is a promising methodology that has the potential to lessen the domain-specific knowledge needed for the creation of customized DL through the automation of critical components of ML pipelines, specifically hyperparameter optimization and neural architecture search (NAS). By reducing the reliance on human trial-and-error in the design of DL models, Auto-ML is able to discover novel model structures with minimal computational resources in its more recent iterations [11].

Furthermore, ML models have the ability to undergo training using extensive amounts of network traffic and device behaviour data, facilitating accurate detection of ransomware attacks. Nevertheless, the ever-changing and varied characteristics of IoT device operations present difficulties for ML models in terms of adjusting and generalizing efficiently [12]. DL models acquire the ability to distinguish authentic from malicious domains by employing a training dataset comprising samples from both categories. The models are provided with labelled samples of legitimate and malicious domains to facilitate their training and learning process. Within the realm of DL, LSTM models are deemed optimal for addressing text classification issues owing to their intrinsic capacity to retain correlations from previous inputs [13].

Thus, the implementation of a system such as intrusion detection is a pressing concern with the goal of differentiating between irrelevant and pertinent data to maintain the availability and integrity of data [14]. This review provides valuable perspectives on the diverse methods learning techniques and their approaches used for identifying malware in IoT networks. Consequently, the core contributions of the review paper include the following:

1. Clarifies the foundation of malware and trend technologies for propagation malware with new approaches for detection.
2. Investigate the stealthiest techniques that are used for the malware.
3. Overview of malware detection in general and specifically detection in IoT environments, as well as their analysis techniques.
4. Expounds on crucial methodologies and strategies employed in the detection of malware.
5. Recent studies in this field have used different learning techniques.

The structure of this comprehensive reviewer paper is as follows: Section 2 describes the fundamentals of malware, which consists of the most common malware with their function and the techniques that malware uses for stealth. Section 3 presents a comprehensive review-based malware detection analysis and approaches. Furthermore, it covers aspects of malware detection on the IoT. Section 4 explores the prior research associated with the present subject and clarifies the systematic details of each study. Section 5 highlights the current approach limitations and discusses the challenges in this particular field. Section 6 discusses the possibilities of future research directions to improve the performance and effectiveness of different learning technique-based malware detection methods in the IoT environment. Section 7 concludes this paper by integrating the key findings and academic contributions.

2. MALWARE FOUNDATION

The term malware, also referred to as “Malicious software,” is used to gain unauthorized access to systems. It can slow down a computer and internet connection, steal or gather sensitive information, infiltrate personal computer systems, send spam from a device, target other computers, and send files to criminal entities. The definition of malware is continuously expanding due to the ongoing evolution of new exploits. Moreover, the volume, variety, and capabilities of malware threats are increasing due to the opportunities provided by technological advancements. IoT devices, smart devices, social media platforms, internet connections, smartphones, and more facilitate the development of intelligent, complex, and highly sophisticated forms of malware [15]. The constant variety and complexity of malware make classification difficult. Nevertheless, the classification of malware provides important insights to improve its understanding. Before examining the

working foundations of malware, let us define some common terms used to describe the different categories of malware, as shown in Table I [15], [16]:

TABLE I. MOST COMMON TERMS OF MALWARE

Terms	Year Appeared	Elaboration	Function
Virus	1971	Malicious programs have the ability to affix themselves to authentic files or software applications and propagate through the execution of the contaminated file.	Omit files Disabling system
Worm	1988	Worms represent self-replicating programs that spread across networks that hand down from one system to another without the need for user interaction.	Network damage Slowdown
Trojan Horse	1983	Trojans showing harmless but contain hidden malicious code. Moreover, they frequently disguise themselves as authentic software or files, deceiving users into initiating their execution.	Steal sensitive information
Spyware	Late 1990s	Same as trojans horse	Rerecording the keystroke of keyboarded for data theft
Adware	1995	Displays unwanted advertisements or redirects users to promotional websites. While not always malicious, the existence of a substantial volume of adware can lead to considerable irritation and negatively impact the system's performance.	Fills the computer with tons of ads in order to monetize
Ransomware	1989	Lead an individual's computing system to be data encrypted. It can cause significant data loss and financial harm.	Locks all file on computer and demands payment (usually in cryptocurrency) to unlock them
Backdoor	1998	Designed to breach system security by manipulating local security protocols, thereby enabling unauthorized remote access and control of a network.	Attackers use them to maintain control even after initial compromise
Keylogger	1980s	Insidious form of spyware.	Same as function of spyware
Botnet	1999	The composition involves compromised devices, also known as (bots), which are overseen by a central server.	Used for 1) DDoS attacks 2) spam distribution, 3) other malicious activities.
Rootkit	Mid 1990s	Attain heightened privileges in order to oversee the system and circumvent security protocols.	Hide malicious processes files, or network connection from detection

Importantly, practices of cybersecurity, consistent software upgrades, and trustworthy security applications are crucial in safeguarding against such vulnerabilities. In the field of literature, malware is often characterized by three prevailing properties. (1) The propagation method refers to how malware spreads from one system to another and determines how the malicious code infects new hosts. It establishes how malicious code spreads to new systems. Moreover, comprehending the significance of propagation aids in creating efficient defence mechanisms; (2) Concealment strategy, which encompasses methods employed by malicious software to avoid being detected and poses a challenge for security solutions in recognizing and examining harmful code; and (3) Persistence, which guarantees that malware stays operational on a compromised system even following reboots. This is accomplished by altering system configurations, generating registry entries, or setting itself up as a service. Table II lists specific malware examples on the basis of these properties [16],[17]:

TABLE II. MALWARE CATEGORIES BASED ON PROPAGATION, CONCEALMENT AND PERSISTENCE METHODS

Malware	Propagation Method	Concealment Strategy	Persistence
Virus	Email attachments	Polymorphism	Altering system configurations
Worm	Network spread	Metamorphism	Exploiting memory vulnerabilities
Trojan Horse	User execution	Fileless	Altering system configurations
Spyware	User interaction	Obfuscation	Altering system configurations
Adware	Software bunding	Stealth techniques	Altering file system
Ransomware	Email attachments	Encryption	Altering file system
Backdoor	Manual installation	Code obfuscation	Altering system configurations
Keylogger	User interaction	Memory-based	Altering system configurations
Botnet	Network propagation	Dynamic behaviour	Altering system configurations
Rootkit	Manual installation	Concealed processes	Altering system configurations

As shown in Table II, malware types and distribution techniques differ. Some spread across networks, whereas others depend on human behaviour, such as opening email attachments or running malicious code. Obfuscation, polymorphism, and metamorphism are common concealment methods. System settings are often changed for persistence; therefore, they must be secured and monitored. Owing to its developers' distinct strategies, each form of malware is hard to recognize and delete. If one knows these approaches, one may construct concentrated defences to reduce these hazards.

2.1 Stealthy Technique of Malware

One way to categorize malware knowledge is into "classic" and "modern" malware. Malware of yesteryear was both pervasive and target-agnostic. Typically, it uses well-known security holes and does not try to remain undetected for very long. The system was not severely impacted by these strikes, and they were very transitory. Regardless, malware has

become more sophisticated as a result of technical advancements. Advanced malware is usually targeted and persistent, with specific targets in mind. It can exploit unknown vulnerabilities, known as "zero-day" vulnerabilities, and employ complex methods to avoid detection. Therefore, advanced malware poses a greater and more challenging threat than traditional malware does [18]. Stealthy malware or ‘stealthy techniques can employ a variety of techniques.

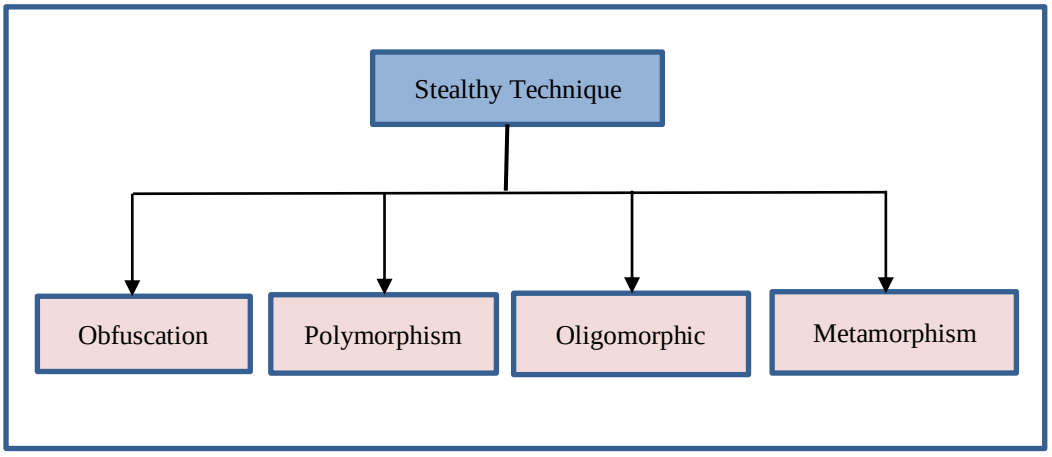

Fig. 1. The stealthy technique of malware.

1) Obfuscation

Intentionally complicating code to impede reverse engineering and analysis is known as code obfuscation. This practice serves to obscure the genuine intent of the code. In the realm of malicious software, obfuscation frequently encompasses encryption, manipulation of data, and various strategies aimed at perplexing analysts and antivirus scanning tools. Further elaboration of this technique obfuscation involves altering plain, easily readable code, scripts, or text into a complex form to hinder understanding and reverse engineering by researchers and automated analysis. Legitimate software developers employ obfuscation to safeguard their intellectual property, increase the challenge of copying or altering their code, prevent software reverse engineering, and protect copyright licences. Through this technique, malware creators can cloak their malicious code, posing challenges for researchers in analysing, detecting, and mitigating it [18], [19], [20], [21].

2) Polymorphism

In cybersecurity, this practice can be perceived as a type of obfuscation, enabling malware to alter its fundamental characteristics and actions, thereby increasing the challenge of detection. Furthermore, the influence of polymorphic malware on software applications surpasses that of conventional malicious software, which is detectable by antivirus programs. The initial emergence was characterized by its ability to alter and decrypt itself, resulting in a few undetectable threats through traditional signature-based systems. Malware creators have continuously devised numerous malicious schemes daily, utilizing techniques such as obfuscation code and code insertion. These individuals leverage polymorphic toolkits, such as mutating engines and polymorphic packers (referred to as polymorphism engines), to transform no obfuscated malware into polymorphic forms. The detection of malware produced through this method occurs postinfection of the targeted system, increasing the likelihood of achieving its objectives before being identified [22], [23], [24].

3) Oligomorphic

Oligomorphic malware represents a form of polymorphic malware that is distinct in nature from fully polymorphic malware, which has a wide range of manifestations. In contrast, oligomorphic malware adheres to a restricted number of preset configurations. The code of an oligomorphic nature alters its structure but remains constrained within a limited array of potential formats. This characteristic facilitates comprehensive coverage via signature-based detection methods. Despite their lower adaptability than fully polymorphic malware, oligomorphic approaches present considerable complexity for security mechanisms aiming to detect and obstruct malevolent code [25], [26], [27].

4) Metamorphism

Metamorphic malware elevates obfuscation to a greater degree by not only altering its visual representation but also completely transforming it with each iteration. The core functionality remains unchanged; however, the internal arrangement of the code varies with each execution. Continuous metamorphosis poses a significant challenge in terms of detection. The mechanism employed by metamorphic malware involves self-rewriting of its code, modification of instructions, and application of diverse transformations [28], [29], [30], [31]. Table III summarizes the above techniques with their key features;

TABLE III. SUMMARY OF STEALTHY TECHNIQUES

Technique	Description	Key Features
Obfuscation	Engages in the modification of the code architecture to obscure its genuine intent.	• Uses encryption and compression. • Makes reverse-engineering difficult.
Polymorphism	Modifies the code of the malware with every instance of infection while preserving its operational capabilities.	• Generates unique variants. • Evades signature-based detection.
Oligomorphic	Similar to polymorphism but with a limited set of variations.	• Fewer unique variants than polymorphic malware • Easier to detect than polymorphic malware.
Metamorphism	Completely rewrites its own code with each iteration, creating new variants.	• More complex than polymorphism. • Difficult to analyse due to constant change.

3. MALWARE DETECTION OVERVIEW

In the modern interconnected digital environment, malware risk presents a substantial obstacle to cybersecurity. Malicious software, encompassing viruses, worms, ransomware, and trojans, has advanced intricacy and developed. Conventional identification approaches reliant on signatures frequently prove inadequate in confronting these progressing risks. A potential solution for this problem has arisen in the form of implementing ML and DL approaches for the assessment and identification of malware[32], [33]. ML algorithms utilize historical data for the purpose of recognizing patterns and anomalies linked to malware. Various methods employed in this process include decision trees, random forests, support vector machines, and neural networks. The extraction of features, encompassing behavioural attributes or statistical characteristics from malware samples, is essential in the realm of ML-driven detection. Scholars have delved into the realm of ML models for conducting both static and dynamic evaluations of malware samples[34]. Advanced learning models (DL), particularly deep neural networks (DNNs), have shown potential in identifying intricate and diverse forms (polymorphism) of malicious software. In the form of detecting maliciousness, convolutional neural networks (CNNs) and recurrent neural networks (RNNs), which evaluate “raw binary information” or “sequences of system calls”, which involve the adjustment of preexisting DL models for the purpose of malware detection, have garnered significant interest [32] [35], [36].

3.1 Analysis techniques of malware detection

The analysis technique of malware detection involves the systematic inspection of executable files to extract valuable information. The primary objective is to delineate the scope of a cyberattack and identify the various features and actions of malicious software. The ultimate aim is to mitigate the risk of comparable cyber threats that target information systems. Malware detection analysis can be classified into three distinct types: static, dynamic, and hybrid, as shown in Fig. 1[37]. The static analysis involves scrutinizing an executable file without its execution, whereas the dynamic analysis necessitates the execution of the file to scrutinize its behaviours within a controlled setting. Hybrid methodologies combine insights gathered from both static and dynamic analyses to gather information about malware. All three analyses are illustrated in detail below [38], [39], [40].

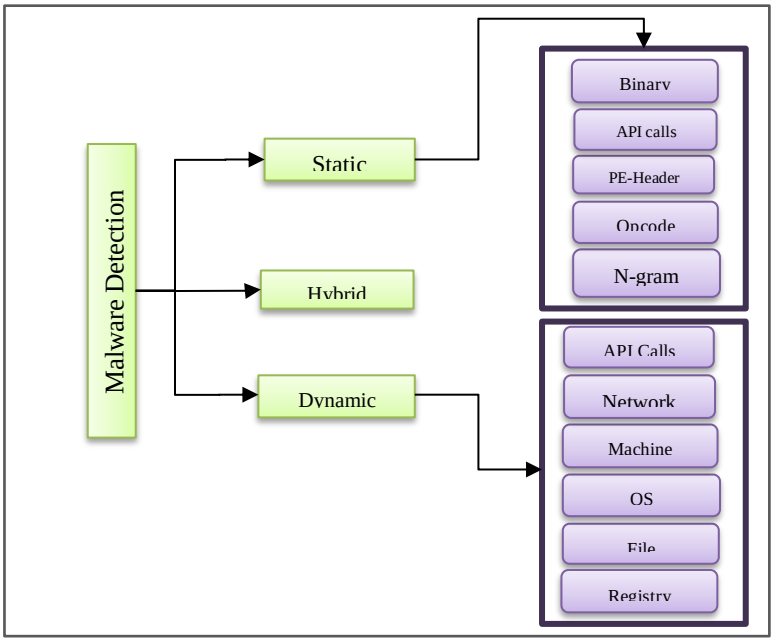

Fig. 2. Malware detection analysis [37].

- **Static Analysis Technique**

In the static analysis technique, the feature extraction method, also known as static feature extraction, can be utilized to derive characteristics from the contents of executable files without their execution. Through analysis of file formats such as portable executable files (PEs) and common object file formats (COFFs), these static features are extracted [41]. A PE file must submit unpacking and decompression prior to analysis. Tools such as 'IDA Pro' and 'OllyDbg', which offer a display of assembly instructions, provide insights into the malware, and facilitate the extraction of patterns for attacker identification, can be quite beneficial [42]. Notably, the portable executable (PE) file format, which is commonly used for Windows executable files, has garnered interest in the field of static malware analysis. ML has shown strong effectiveness in detecting malicious PE files, with certain studies utilizing deep neural networks [43].

- **Dynamic Analysis Technique**

A pivotal stage of the scholars of cybersecurity, as they commence comprehending the actions of a harmful specimen (for example, ransomware) during its operation within a secure setting. This process facilitates the examination of the traces left by the sample under review, encompassing window API calls, registries, file system processes, network communications, etc. Scrutinizing such records of events empowers scholars to formulate defensive tactics aimed at identifying a malevolent operation as it unfolds on a compromised system, causing harm (such as encrypting data in the scenario of ransomware) [44], [45]. In light of the forthcoming implementation of a maliciousness artifact in this analysis, it is imperative to establish a secure setting to ensure that adverse outcomes, such as the transmission of infections to adjacent networks or the compromise of the host computer by malicious software, are averted [46].

- **Hybrid Analysis Technique**

The term "hybrid"' combines both of the above techniques. This combination of static and dynamic analysis improves the accuracy of identifying malicious software by utilizing their individual strengths. Analysts and security researchers utilize API calls to increase their comprehension of malware operations, detect their presence, and devise efficient strategies to mitigate their impact, since both static and dynamic techniques use API calls. The examination of API calls the "application programming interface", which is conducted by malicious software (malware) during operation, enables the evaluation of their behaviours in a real-time setting, providing crucial insights into their capabilities, such as unauthorized data exfiltration and intrusion into computer systems [47]. Table IV shows a comparative summary of these techniques and reveals their operational mechanisms, effectiveness, and application contexts, which are crucial for understanding their roles in cybersecurity.

TABLE IV. SUMMARY OF ANALYSIS TECHNIQUES

Technique	Definition	Strengths	limitations
Static Analysis	Involve examining the code without executing it.	• Fast and efficient for identifying the signatures of known malware. • Can detect vulnerabilities in the code structure.	• Cannot stop obfuscated or polymorphic malware. • Many pieces of benign code mistakenly detected as malware. (high false positive rate (FPR))
Dynamic Analysis	Includes code execution to monitor behaviour in a controlled environment.	• Effective in identifying run-time behaviours and interactions with the system. • Can detect previously unknown malware through behavioural patterns.	• Resource-intensive and time-consuming. • May miss malware that does not exhibit malicious behaviour during analysis
Hybrid Analysis	Involve comprehensive detection by Combines both static and dynamic techniques.	• Capitalizes on the advantages presented by both methodologies, thereby enhancing detection efficacy. • Minimizes the occurrence of false positives (FP) and improves the detection of intricate malware.	• It requires more resources and complex to implementation. • May still struggle with advanced evasion techniques that used by sophisticated malware.

While static and dynamic analyses provide foundational detection capabilities, hybrid analysis emerges as a robust solution, integrating the strengths of both approaches. However, the evolving nature of malware continues to challenge all detection methods, necessitating ongoing research and adaptation in detection strategies.

3.2 Approaches of the malware detection

Scientifically understanding the term of malware detection, which is the process of detecting malware, involves the mechanization that is necessary to be deployed to uncover and recognize the malicious behaviors exhibited by the files being examined [48]. In addition, the individual technologies used to detect malware include certain advantages and

disadvantages; however, the incorporation of recent malware detection methods aims to increase the security of computer systems by identifying and preventing potential data loss and vulnerabilities stemming from malicious intrusions facilitated by harmful software. The system is further strengthened through the efforts of the malware detector in identifying malicious activities. While various detection methods, including signature-based, behaviour-based, model-based, and heuristic-based detection approaches, along with emerging techniques such as DL, cloud, mobile device, and IoT-based methods, are efficient in promptly and accurately spotting known viruses, signature-based systems fall short in detecting unknown malware due to the lack of files for assessment and subsequent virus total submission, posing a challenge for security software in identifying fileless malware [49], [50].

- **Signature-Based Model For Malware Detection**

The use of signature-based techniques for identifying malware plays a pivotal role within the framework of our cybersecurity approach. The current approach has been formulated with the aim of identifying and averting the emergence of recognized malware patterns and signatures. By recognizing these extensively known threats, we are able to promptly respond to and reduce their impact, thereby providing a robust initial defence for SMEs and smart homes [51]. Furthermore, the Antivirus software frequently employs this method of malware identification to isolate the signature of the analysed file and collocate it against the signatures catalogued in a repository of recognized security risks [52].

Signature-based models have shown inefficacy in the detection of zero-day malware. The constraints of signature-based and pattern-matching approaches include their inability to identify zero-day attacks and their restricted ability to recognize a specific category of malware [53]. While the conventional methods of signature-based detection present a straightforward and efficient means of recognizing established threats, they involve a number of constraints when encountering dynamic and diverse threats. The advantages of employing signature-based strategies lie in their rapid and effective identification of familiar threats, along with their straightforward implementation and administration. Nevertheless, these techniques prove inadequate when encountering unprecedented or previously unseen attacks, as they depend on predetermined signatures that may overlook emerging threats or variations of recognized malicious software. Furthermore, the susceptibility of signature-based detection to circumvention techniques, such as polymorphism and obfuscation, poses a challenge because it enables the bypass of signature matching and evasion of detection[54].

- **Behavior- or Anomaly-Based Model For Malware Detection**

Behaviour patterns in malware detection involve analysing incoming images on the basis of the tasks and actions they were preprogrammed to execute before determining the feasibility of proceeding with an operation. The scrutiny includes an examination of the functions of a file and potentially its threat level. The use of behaviour-based techniques in technology enables the swift identification of new and unexpected threats, acknowledging the inherent limitations of computers in achieving absolute accuracy. Unlike the signature model, which detects only known malware. acknowledging the inherent limitations of computers in achieving absolute accuracy [55].

- **Heuristic Based Model For Malware Detection**

It represents a methodology to detect and differentiate the customary and anomalous operations of a system with the aim of recognizing identified and unidentified malicious software intrusions and devising an appropriate resolution [56]. During the period from 2000--2010, the integration of heuristic-based detection methods with signature-based detection was a key approach in addressing malware threats. The heuristic method emerged as a particularly promising approach for identifying novel or unfamiliar forms of malware. Moreover, within this methodology, domain specialists rely on their expertise in malware analysis to establish regulations capable of distinguishing malicious software from harmless programs. These regulations must be adaptable for effectively detecting malware and its different versions with precision and a minimal rate of false positives. Nonetheless, the establishment of such rules is manually conducted by field experts, leading to a slow and arduous procedure [57].

The use of both signature- and heuristic-based malware detection has its own set of advantages and disadvantages. Although static and dynamic analyses represent robust mechanisms capable of functioning independently for malware analysis, an additional category that can yield advantages is the hybrid approach, which integrates both static features and dynamic features for analysis. The optimal level of security will be achieved through the utilization of both technologies. This methodology is particularly advantageous when attempting to reverse engineer intricate malwares. The true motives and capabilities of malware can be more effectively scrutinized through the application of a hybrid approach [58].

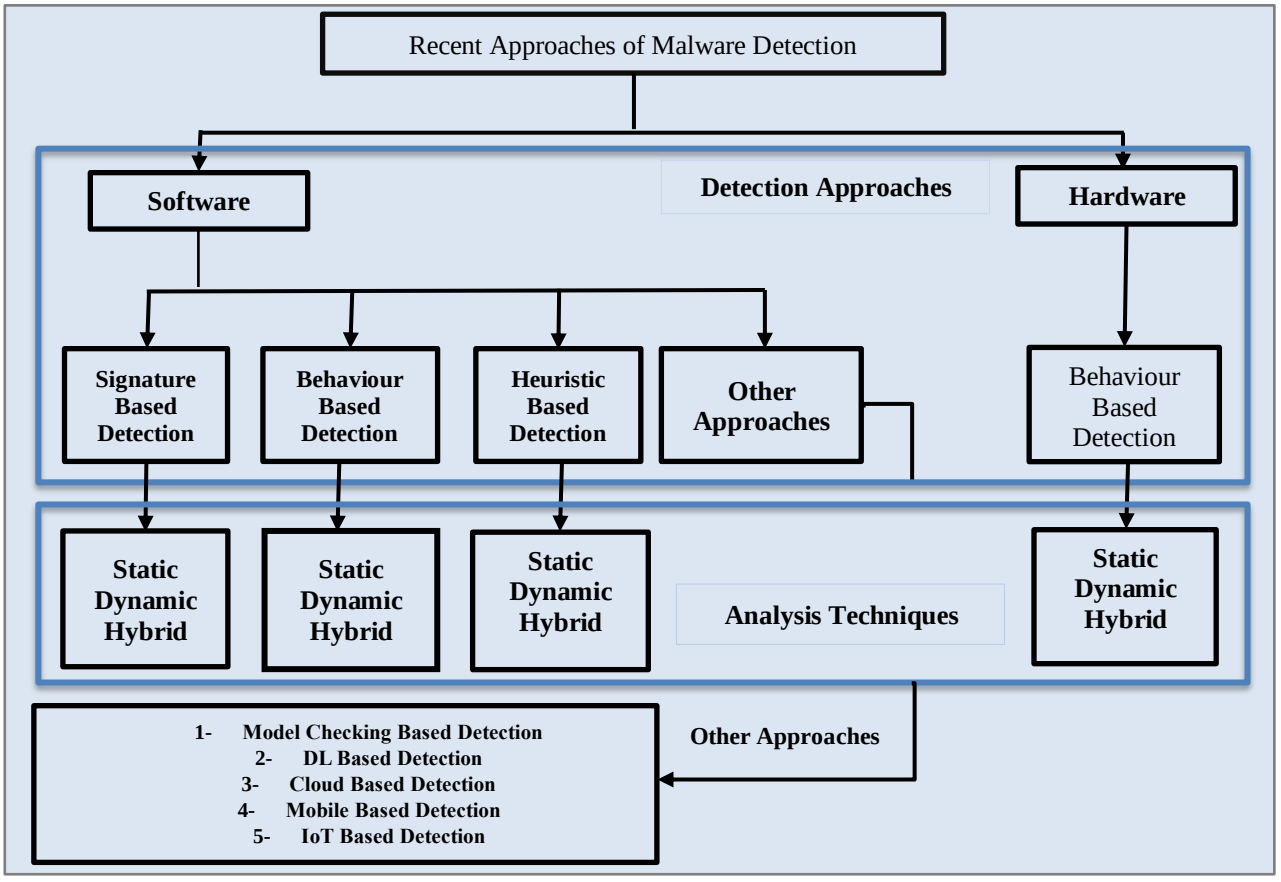

Fig. 3. Recent approaches to malware detection [59].

3.3 Malware Detection in IOT

Throughout recent decades, the vast majority of harmful software has been designed to exploit weaknesses in personal computers running on the Microsoft Windows system, which has maintained the top position in the worldwide market, with an 83% share. Nevertheless, the variety of computing devices has undergone rapid transformation in recent years, largely driven by advancements in IoT technology. IoT devices are constructed on a range of central processing unit (CPU) architectures, even extending to low-powered hardware such as Unix-based operating systems. This shift has made IoT devices increasingly attractive targets for malicious actors, primarily due to inadequate security protocols and mechanisms. In general, IoT malware shows particular features such as its utilization in orchestrating DDoS attacks; scanning open ports of IoT services such as the File Transfer Protocol (FTP), Secure Shell (SSH), or Telnet; and launching brute-force attacks to compromise IoT devices [60]. Ensuring the protection of the IoT system from malware is crucial for the safety of workers and fundamental for maintaining the effective operation of the Industrial Internet of Things (IIoT). These IoT devices can manifest as tangible entities or as intangible entities that engage in interactions [61].

To actualize the IoT, advancements in computing are imperative, transcending the realm of traditional mobile and smartphone applications to encompass the interconnection of mundane physical objects with an infusion of intelligence within their surroundings. Consequently, novel computer functionalities and challenges will emerge. As a result, there is a heightened ability to promptly cease unsafe activities due to the proliferation of IoT networks. Malicious software incidents frequently target the cybersecurity of personal computers, smartphones, and IoT devices. The concept of the IoT pertains to the interlinking of intelligent objects, spanning from minor appliances such as coffee makers to substantial vehicles, enabling autonomous communication devoid of human intervention, commonly referred to as device communications. The contemporary landscape has undergone a progressive evolution towards smarter technologies that possess the ability to engage with diverse devices. Owing to the diverse and complex nature of the IoT, ensuring universal privacy is increasingly challenging, as its rapid proliferation extends across multiple sectors, such as smart residences, healthcare facilities, and beyond. Various forms of vulnerabilities are discernible in this context [62], [63].

In the interim, conventional computers pose a significant threat to the IoT ecosystem by leveraging them to compromise other interconnected devices within the same environment. Observing these patterns, it is evident that IoT applications represent a burgeoning domain for exploration in the realm of security [64]. High-tech innovations and rapid developments, such as big data, 5G, computational intelligence, and the IoT , are combined to overhaul different industries during the Fourth Industrial Revolution [65]. Among these rapid techniques, the interplay among IoT, AI, and 5G drives the incorporation of intelligent technologies into various sectors, such as intelligent vehicles, manufacturing facilities, and urban areas [66].

The ongoing revolution of industries brought about by this transformation is accompanied by the significant impact of the expanding IoT market, not only altering industrial environments but also exerting a lasting influence on our everyday activities. Nevertheless, the interconnected nature of IoT devices renders them vulnerable to a wide range of cybersecurity risks, such as (botnet activities, cryptocurrency mining, and distributed denial-of-service (DDoS)) assaults that these attacks expose open port access in IoT devices [67].

4. RECENT RESEARCHS ON MALWARE DETECITION APPROACHES

This section presents the most recent studies from 2000--2024 on malware detection, either in the IoT environment or another environment, which are clarified in Table V. The table shows basic information for each study and its own contribution, with the technique used in feature selection being determined and how it represents these features with the learning technique and their evaluation with the dataset and operating system used. These studies were conducted with concentrated-based malware detection within the IoT as well as in other contexts.

TABLE V. RECENT STUDIES ON MALWARE DETECTION BASED ON DIFFERENT LEARNING METHODS

Ref	Year	Contribution	Features	IoT	Techniques	Evaluation	Dataset	OS
[68]	2020	Cloud-based nested virtual environment (convert behaviour of extracted data to image). this study aims to adopt a novel approach known as dynamic analysis for IoT (DAIMD) malware detection. This method utilizes dynamic analysis on IoT malware within a nested cloud-based Virtual Machine (VM) environment. moreover, it utilizes a Convolutional Neural Network (CNN) model to learn behaviour images that are condensed with a substantial volume of behavioural data.	Behaviour of Memory, Network, System call, Virtual file system (VFS), & Process (Dynamic Analysis)	yes	ML CNN	Accuracy= 99.28% FPR=0.63% FNR=0.74%	ZF-Net model (840 images) Dataset	Linux
[69]	2020	Reduces the number of API calls that represent sample (malware or good-ware). this study aims to detect malware and analyse it based on API calls-dynamic sequences.	Clustering API functions (Dynamic Analysis)	No	ML Clustering	Accuracy= 0.997% Precision= 0.990% FPR= 0.010% FNR= 0.007%	Malware & Good-ware (API call sequences) Dataset	Win
[70]	2020	Present multiple processes to study and generalize the heterogeneous sequences of API. this study aims to developing two feature extraction and generation approaches, namely, Method 1 and Method 2, which do not rely on expert domain knowledge of the arguments. in first method, all arguments of each API call are treated as a single feature, where the total number of features corresponds to the number of API calls. On the other hand, adopt generalizing each argument of every API call individually. moreover, the	Bit-vectors (Dynamic Analysis)	No	ML	Method1 Accuracy= 99.87% Method 2 Accuracy= 97.95%	Mal-share website	Win7

		resulting that related to the number of features being equivalent to the number of API arguments.						
[71]	2020	Present the first ML-based malware visualization detection method. this study aims to visualizing the global malware file format and provide the accuracy of detection to avoids potential intrusions arising from malware (AEs) adversarial examples and their corresponding modifications.	Greyscale Image	No	ML CNN & SVM	Accuracy= 97.73% Average (Tested)= 96.25%	MS BIG malware The Ember database	Linux
[72]	2021	Present Densely Connected Convolutional Networks (Dense-Net). this study aims to accomplish expedited preprocessing and training of binary samples and tackle the issues of the imbalanced data in classifying malware by implementing reweighting techniques of the class-balanced categorical cross-entropy loss function within the soft-max layer.	2D greyscale image (Behaviour Analysis)	No	DL CNNs	Dataset 1 Accuracy= 98.23% Dataset 2 Accuracy= 98.46% Dataset 3 Accuracy= 98.21% Dataset 4 Accuracy= 89.48%	Malimg, BIG 2015 , Male-Vis Unseen-Malicia	Linux
[73]	2021	Present a new hybrid visualization algorithm. This study aims to converting the dynamic analysis results into a visualization colour image as RGB-image.it uses the both static and dynamic analysis for robust result.	Colour image (Hybrid Analysis)	No	DL	Model 1 Accuracy= 91.41% Model 2 Accuracy= 94.70%	VIRUSSIGN (Open Malware Database)	Win7
[74]	2021	Present Hybrid Ensemble Learning (EL) framework consisting of fully connected & convolutional neural networks (CNNs) with the Extra-Trees classifier as a meta-learner for malware detection	Principal component analysis (PCA) used for reduced the feature dimensionality	No	ML EL	Recall= 100.0% Precision= 100.0% Accuracy= 100.0% F score= 100.0%	Windows Portable Executable (PE) malware benign files (Kaggle	Win10
[75]	2021	Present an intelligent behaviour-based detection system in the cloud environment. this study aims to adopt two phases, in the first phase adopt feature extraction whereas the second one is the detection phase.	Dynamic tools Feature Vectors using (Dynamic Analysis)	No	RULE & Learning-Based Detection	Detection Rate= 99.85% FPR=0.4% Accuracy= 99.7%	Das Mal-werk, Malware Bazaar Malware DB Malware Benchmark Mals-hare, Tek-defence Viru-Sign, Virus-Share, Kernel-Mode	Win 7 8 10 VMs 7 8 10
[76]	2022	Present Security farmwork detection that uses FL adopt both (supervised & unsupervised) which means anomaly detection and classification approaches that using multilayer perceptron and autoencoder neural network architectures. Moreover, Using Use case (B5G) Beyond 5G which uses to detecting cyberattacks that affecting IoT devices, managing sensitive data,	Coordinate	Yes	FL	Supervised rustle for 95% benign Accuracy= 99.42% TPR= 99.81% TNR= 99.40% Unsupervised results TPR= 99.98% TNR= 92.76%	N-BaIoT Dataset	PC

		having Non-IID (Independent and Identically Distributed) data, and with nontrusted stakeholders or clients						
[77]	2023	(Hybrid DL and ML-based malware detector). this proposed combining deep transfer learning & ML utilize first, deep transfer learning is used to extract all the deep features from the last fully connected layer of the DL model, and then ML models are used as the final detector, which fully utilizes the inherent connections between input and output.	Colour Image	No	ML DL	Accuracy= 99.06%	Microsoft malware Malimg Virus-Share Dataset	Win
[78]	2023	Present (DEMD-IoT) the system comprises a stack of three one-dimensional (Convolutional Neural Networks) with distinct architectures for capturing diverse patterns of IoT network traffic. Additionally, a Random Forest classifier functions as a meta-learner atop the three 1D-CNN base-learners, merge the results of the CNNs and accurately labelling each network flow. The methods successfully addressed the challenge of transforming malware into images on (2D-CNNs). ultimately decreasing the preprocessing duration and computational burden. This was achieved by employing 1D -CNNs and automatically extracting features without the need for manual intervention.	Image	Yes	DL EL	Accuracy= 99.90% Precision= 99.83% Recall= 99.97% F1-score= 99.90%	IoT-23 Dataset	Win10
[79]	2023	Developed (SB-BR-STM CNN and Ensemble classifiers) A novel DSBEL framework is suggested to identify packets infected with malware in an environment of the IoT. This framework consists of the integration of recently created SB-BR-STM CNN and ensemble classifiers. Whereas the SB is referred to Squeezed-Boosted, BR-Boundary-Region and the STM- Split Transform-Merge	Image	Yes	DSB EL	Accuracy= 98.50% Precision= 98.42% MCC= 91.91% F – Score= 97.12% Recall= 95.97%	IoT Malware Dataset	PC
[80]	2023	Integrating Bioinspired and ML Techniques This study adopts the ant colony optimizer algorithm for feature selection by choosing a reduced set of features in order to enhance the classification performance of a malware detection system based on images, specifically through the application of the SVM classifier. moreover, the Particle Swarm Optimization algorithm also adopted to optimize the Support Vector Machine parameters for the most suitable kernel function.	2D RGB image	yes	ML SVM	Accuracy= 95.56% Recall= 96.43% Precision= 94.12% F1-score= 95.26%	Public Dataset (network traffic sources)	PC

[81]	2023	compares the performance of RF and NB algorithms in the detection of IoT malware attacks. Starting by data cleansing is conducted with the purpose of eliminating extraneous data. utilizing two ML for training the sampling data, namely, Random Forest and Naïve Bayes. Subsequently, an evaluation is carried out to compare the efficacy of these two algorithms to determine the more effective one for the detection malware network traffic.	correlation matrix	yes	ML	Naïve Bayes Accuracy= 99.42% Weighted Precision= 83.47% Micro Precision= 81.46% Weighted Recall= 84.32% Micro Recall= 84.24% Weighted F1-score= 83.89% Micro F1-score= 82.83% Random Forest Accuracy= 99.92% Weighted Precision= 98.27% Micro Precision= 98.26% Weighted Recall= 98.85% Micro Recall= 98.85% Weighted F1-score= 98.56% Micro F1-score= 98.55%	IoT-23 Dataset	PC
[82]	2023	this study aims to a sophisticated malware detection system that leverages DL and feature selection techniques. Utilizing two distinct malware datasets, the system aims to identify malware instances and distinguish them from benign activities. moreover, adopt correlation-based feature selection, the dens and LSTM.	Correlation degree between that feature selected	No	DL	(no feature selection) Accuracy = 99.99% F1-score = 100% (63.63% selected features) Accuracy = 99.75% F1-score = 99.8% (no feature selection) Accuracy = 98.38%, F1-score = 98.9% (18.22% selected features) Accuracy = 94.59% F1-score = 94.9%	Daratset1 Daratset2	PC
[83]	2023	A new DL-based automated framework called API-Mal-Detect for detecting malware attacks in windows.	Dynamic Analysis for identify sequences of API calls present in both benign and malicious executable files.	No	DL	Accuracy= 99.07%	Benchmark Datasets	Win10

[84]	2023	A hybrid CNN model for detecting malware in the IoT environment by adopting image-based detection this study aims to generate the image from binary of malware	RGB images	yes	DL	Accuracy= 98.65% precision= 98.7% Recall= 98.3% F1-score= 98.5 % MCC=97.5% Kappa= 97.65%	Dataset1 IoT malware (Kaggle site)	Win10
						Accuracy= 97.3% precision= 96.4% Recall= 96.85% F1-score= 96.63 % MCC= 95.25% Kappa= 94.85%	Dataset2 Virus-share	
[85]	2023	Enhanced malware detection accuracy and mitigation of vanishing and exploding gradients in the neural network. this study aims to present a novel hybrid approach, which utilizes evolving sequences of API calls for the purpose of detecting malicious software. This presented study integrates logistic regression from ML as the primary input weight for the neural network.	Behaviour Analysis	No	ML DL	Accuracy= 83% loss= 0.44	Dataset1 Balanced	PC
						Accuracy= 98% loss= 0.10	Dataset2 Imbalanced (API call sequence)	
[86]	2023	A stacked double-layer convolution network to extract binary file features. this study aims to merge binary file with assembly code based on hybrid detection model for malware.	Binary files Feature Vector Generation Module & Multi-Feature Fusion Module	No	DL	Accuracy= 99.54 % Recall= 99.41% precision= 99.40% F1-score= 99.40%	Dataset1 Kaggle Malware	----
						Accuracy= 95.44 % Recall= 93.62% precision= 92.06% F1-score= 92.81%	Dataset2 Mal-share	
[87]	2023	An ensemble DL method with parallel processing. this study aims to adopt a hybrid Particle Swarm Optimization for optimized the parameters of the DL.	Numerical feature vectors	No	DL	Accuracy= 99.2% Precision= 99.23% Recall= 99.4% F1-score=99.3%	Dataset1 Drebin,	Ubuntu
						Accuracy= 99.3% Precision= 99.2% Recall= 99.3% F1-score= 99.5%	Dataset2 NTAM	
						Accuracy= 99.5% Precision= 99.3% Recall= 99.7% F1-score= 99.6%	Dataset3 Dike	

[88]	2023	An (AWS) artificial immune system enabled validation framework for the evaluation malware detection solutions of AIS. this study utilizes the transfer learning with IoT to overcome the memory constraint	NPS	yes	AI	(30 detectors results) Accuracy= 92% Precision= 95% Recall= 97% (30 detectors results) Accuracy= 99% Precision= 100% Recall= 99%	NSL-KDD Dataset	-----
[89]	2023	Enhancing the IoT device security by performed an improved cryptography algorithm and malware detection. this study aims to adopt a deep LSTM to prevent attack and using (MA-BW) Mayfly Optimization -Black Widow method for choosing best key during the generation of key.	Anomaly contextual features	yes	DL	Accuracy= 95% Error= 5% Precision= 92% Specificity= 96% FNR= 10% FPR= 3% NPV= 96% Encryption, time= 6.02 s Decryption time = 0.0152 s Execution Time= 6.02 s	----	PC
[90]	2023	Designed a hybrid CNN-(Bi-LSTM) bidirectional long short-term memory framework for the detection. moreover, classification of OMM types (multiclass detection) this study adopts two layer of block CNN.	Bi-LSTM (Layers)	yes	DL	Binary-detection attack Accuracy=0.99% Precision= 1.00% Recall= 1.00% F1-score= 1.00% Attack-family detection Accuracy= 0.84% Precision= 0.85% Recall= 0.85% F1-score= 0.84%	CIC-Malmem-(2022) Dataset	Win10
[91]	2023	Presented a hybrid method for detect malware on IoT this study aims to Dis-assembling the obfuscation malware features and then assemble the extracted feature to graph conversion this level done during obfuscated detection. whereas the no obfuscated detection, this level processed using clustering.	Image based-classification (Statistical Features)	yes	ML	-----	Mirai Darlloz Dataset	-----
[92]	2024	(PPFL-SC) an efficient privacy-preserving FL with a	Protocol	yes	FL	Computation Cost &	Real-world Dataset	Mac

		secure collaborative supporting verification is proposed for improving the malware detection models. the PPFL attempt to allow devices to share their trained model under the assurance of security and privacy protection.				communicatio n efficiency, provide less computation cost & minimum communicatio n cost compared with other works		
[93]	2024	Build a clustering model based on the Control-Flow-Graph features to compare the labelled result with the most famous antiviruses. This study aims to adopt three issues linked to low-quality labels in the IoT malware field have been recognized, and these have been exemplified through case studies, along with the respective justifications.	ELF Features (Executable and linkable format)	yes	ML	Using 4 ML method and 4 label to evolute performance AV Class, 5-Engines-Vote, Kaspersky& Microsoft security. during compared the result of 4 label at the family level across 4 methods of ML, and the AV- Class tool performed the best in both detection rate and accuracy, followed by 5-Engines-Vote, Anti-Virus engine Kaspersky and Microsoft.	Virus-Share website	Win
[94]	2024	A novel approach of combining DL & ML. this study aims first, adopt to extract deep features-based DL, moreover the ML is used to classify the selected features for final detection. the novel approaches alter to convert the PEs into colour image	Colour image	No	DL ML	Accuracy= 99.30% Precision= 100% Recall= 98.60% F1 Score= 99.30%	Malimg Dataset	Win
[95]	2024	This study aims to develop novel techniques such as features detection, Top 10, op 20, and (Random features (RF)). moreover, adopt ML different method to reduce effectiveness of the data that have been unbalanced and achieve low-latency detection.	SK-learn Features	No	ML	Accuracy= 99.98%	UNSW-NB15 Dataset	Laptop
[96]	2024	This study present SFTA-KNN (segmentation-based fractal texture analysis) and Gabor-KNN malware detection method. this study aims to diminishes the likelihood of erroneous categorization, extracting significant characteristics, and improving the accuracy of detecting malicious software.	2D grayscale Image	No	ML	Accuracy= 96.29% Accuracy= 98.02%	Malimg MaleVis Dataset	PC
[97]	2024	This study aims to utilizing the (FABEMD) fast and adaptive	RGB image	No	DL	Resnet-18 Accuracy=	Malimg	Win

		bidirectional empirical mode decomposition methodology for augmenting the training dataset dimensions through the extraction of various IMF images from the original image. moreover, using the extracted IMF images as input videos for the purpose of supplying 3D VGG16 and 3D Resnet-18.				96.64% Precision= 99.9% Recall= 99.9% F1-score= 99.9% VGG-16 Accuracy= 96.14% Precision= 97.0% Recall= 96.0% F1-score= 96.0% Resnet-18 Accuracy= 99.46% Precision= 99.9% Recall= 99.0% F1-score= 99.0% VGG-16 Accuracy= 98.60% Precision= 99.0% Recall= 99.0% F1-score= 99.0%	MaleVis Dataset	
[11]	2024	Present automated ML for static malware detection. this study aims to Offering insights and analysis into the automation parameters utilized in the Auto-ML process for static malware data, while also demonstrating the impact of these parameters on the performance of the ultimately discovered optimal model	Static Analys	No	DL CNN	Accuracy= 0.918% F1-score= 0.921% (TPR: 0.1% FPR= 0.188% (TPR: 1% FPR) = 0.969%	EMBER-2018	VM
						Accuracy=0.990% F1-score=0.984% (TPR: 0.1% FPR= 0.963% (TPR: 1% FPR) = 0.995%	SOREL-20 M Dataset	
[98]	2024	Present a malware detection for static security service (Mal3S) to provide a secure Smart IoT environment to detect various type of malware. this study aims to the detection approach captures a range of malicious behavioural characteristics through the creation of five distinct types of static feature images derived from byte data and assembly code, each depicting the activities of malware in diverse manners.	Static Feature Image	yes	Spatial Pyramid Pooling Network (SPP-net)	Accuracy= 98.35% precision= 0.9867% Recall= 0.9900 F1-score=0.9884	KISA-challenge 2019-Malware Virus-Share BIG2015 Dataset	PC
[99]	2024	Present novel model called SIM-FED that combine DL&FL to detect IoT malware. this study aims to employment of a lightweight one-dimensional Convolutional Neural Network (CNN) to diminish preprocessing duration and	1D-Layers	yes	DL FL	Accuracy= 99.52% Precision= 99.47% Recall= 100.0% F1-score= 99.73%	IoT-23 Dataset	Windows 10

		computational, facilitating the process of automatic feature extraction. moreover, as hyperparameters play a crucial role in the performance of the model, the enhancement of the specified CNN method is executed through the modification of hyperparameters utilizing a remarkable optimization technique.						

As shown in Table V, various studies have effectively employed hybrid DL techniques, combining different methods to enhance malware detection in the IoT environment. M. Nobakht et al. [78] presented a significant advancement model in IoT malware detection by integrating DL and ensemble learning techniques known as the DEMD-IoT model. It uses a stack of three one-dimensional convolutional neural networks (1D-CNNs) to effectively learn different patterns from IoT network traffic, which are then combined using a random forest meta-learner0 for final predictions. The presented model not only simplifies the preprocessing phase by avoiding the complexities associated with 2D-CNNs but also enhances the model's performance through hyperparameter optimization (Grid-Search-CV). An evaluation of the performance of the IoT-23 dataset shows that DEMD-IoT achieves an outstanding accuracy of 99.9%, superior to existing state-of-the-art models in the field. The results provide the model's robustness and potential for scalability, setting the way for future enhancements, including the application of generative adversarial networks (GANs), for further improvements in malware detection capabilities. In general, DEMD-IoT remains a predictable solution to the growing challenges of IoT security.

S. Alsubai et al. [84] proposed an innovative malware detection framework tailored for the IoT environment that uses image-based techniques to convert malware binaries into RGB images. Key features are extracted via the YOLOv7 algorithm, whereas the DenseNet161 model is optimized via the Harris Hawks optimization algorithm for increased classification accuracy. The framework protects IoT resources with accuracies of 98.65 and 98.5 for the IoT malware dataset and 97.3 and 96.63 for the Virus-share dataset.

In addition, R. A. Devi and A. R. Arunachalam [89] introduced LSTM-based DL for malware detection. They demonstrated optimal elliptic curve cryptography (IECC) IoT security research. The proposed technique categorizes normal and attack nodes on the basis of contextual trust values and detects anomalies, DOSs, probes, and R2L attacks with 95% accuracy. Compared with contemporary methods, secure data transfer with IECC integration reduces the encryption and decryption times to 6.02 seconds and 0.0152 seconds, respectively. This research extends the existing knowledge by offering a framework for malware detection and safe data management in IoT networks and addressing important security issues.

S. S. Shafin et al. [90] introduced a novel method for identifying obfuscated memory malware (OMM) in smart city apps for low-resource IoT devices. Compact-CBL and robust-CBL, which use a CNN and bidirectional long short-term memory, achieve 99.92% and 99.98% binary detection accuracy, respectively. They also identify attack families effectively; RobustCBL has 84.56% accuracy. The models outperform state-of-the-art techniques on the CIC-Malmem-2022 dataset without losing a minimal footprint for IoT implementations. CompactCBL, the smallest, is ideal for real-time applications because of its 0.255 ms/sample detection speed. This study provides realistic and effective solutions for malware detection in smart city infrastructures and has made significant progress.

M. Nobakhtet al.'s SIM-FED model [99] integrates DL with FL to protect data and identify dangerous actions. This is IoT malware detection. It reduces computation and preprocessing costs via a lightweight one-dimensional CNN and hyperparameter tuning. The model outperforms other DL and FL models with 99.52% accuracy on the IoT-23 dataset. Federated aggregation solutions were also evaluated, with (Fed-Avg) chosen for the best local model integration. SIM-FED showed low performance reduction from white-box and black-box cyberattacks. These results demonstrate the model's efficacy and security, filling a major gap in the IoT security literature.

ML and FL have been extensively studied for their ability to identify malware in IoT environments, beyond hybrid DL approaches. J. Jeon et al. . [68] developed DAIMD, a dynamic analytic approach for IoT malware detection, to identify existing and developing threats. In layered cloud environments, the DAIMD model dynamically assesses malware activity via a CNN. It emphasizes memory, network, and system calls. Using extracted behaviour data as images improves classification and training. The results suggest that DAIMD may identify new and variant malware that static analysis methods miss, decreasing IoT device infection damage. Researchers conclude that dynamic analysis is essential for managing IoT security concerns in a changing environment.

V. Rey et al. [76] proposed FL for IoT malware detection. This approach addresses important privacy and security issues. The authors utilize the N-BaIoT dataset to simulate corrupted IoT device communications to evaluate their strategy. The

study shows that federated models may protect data privacy while performing similarly to centralized methods. Hostile attacks on the federated paradigm reveal vulnerabilities in standard aggregating methods. Additional aggregating functions enhanced resistance to malicious actors, offering a solution to these dangers. Even though FL offers robust IoT security solutions, the findings show that elasticity against adversarial attacks needs more development. The present study enhances our understanding of FL's IoT cybersecurity and privacy solutions.

Additionally, A. El-Ghamry et al. [80] introduced an innovative image-based IoT malware detection system that uses classical ML techniques, specifically support vector machines (SVMs). For feature selection, this study optimized a method that uses particle swarm optimization (PSO) and ant colony optimization (ACO). Moreover, this study effectively addresses the challenges of low computational resources typical in IoT environments while achieving a stunning accuracy of 95.56%. This study demonstrates a significant improvement in malware detection performance compared with existing methods by transforming network traffic data into images and employing advanced feature extraction methods. The findings highlight the potential for deploying this efficient detection model in real-time IoT applications, ensuring enhanced security against evolving malware threats.

Y. Z. Wei et al. [81] examined the efficiency of the RF and NB algorithms in identifying malware in IoT network traffic. This work uses the IoT-23 dataset, which comprises benign and malicious traffic samples, for model training and stratified sampling to solve class imbalance. The findings show that the random forest model outperforms the naïve Bayes model, with a 98.55% micro average F1 score. Cybersecurity professionals may use this study to determine that random forest is the best way to detect fraudulent network traffic in IoT scenarios.

A. Khan et al.'s extensive study of IoT malware detection approaches highlighted the rising threat of Mirai and its variants [91]. This research highlights the complexity and fragility of IoT devices, emphasizing the necessity for effective detection. According to the review of ML, graph-based analysis, and image-based identification, traditional methods function well with simple malware but fail with complicated versions. This paper proposes a hybrid model that uses both methodologies to increase the detection accuracy. It fixes malware obfuscation and obtains a 99% F-measure for no obfuscated malware. Future development aims to improve detection efficiency and expand design support.

Alamer [92] A unique protocol called (PPFL-SC) secure collaboration, which uses group-oblivious sign-crypton (GOSC), is described in this paper. It aims to improve IoT malware detection. The suggested approach balances privacy with IoT device model sharing by shielding users' data from hostile cloud sites. The findings reveal that PPFL-SC is as accurate as classic FL while protecting privacy. Device dropout rates during model training are significantly reduced by the protocol's incentive mechanism. The findings reveal that PPFL-SC increases the efficiency, security, and accuracy of collaborative malware detection systems in IoT networks.

T. Lei et al. [93] investigated the challenges and evaluation of IoT malware classification label sources through the application of clustering methodologies, underscoring the critical significance of high-fidelity labels for precise research results. It delineates three principal concerns pertaining to label integrity: detection methodologies, nomenclature standards, and label obsolescence. The investigation evaluates 63 malware publications across a spectrum of platforms, including IoT, Windows, and Android, to gauge the efficacy of various label sources. The authors advocate for the utilization of the AV class for familial classification and propose specific (Anti-Virus engines—Ad-Aware, Bit-Defender, and Emsisoft) for minor families, as well as (Jiangmin, NANO-Antivirus, and Avira) for more extensive families—to improve precision in variant-level classification. The results suggest that the endorsed engines yield dependable labels, which are essential for enhancing the accuracy of forthcoming IoT malware research.

In summary, various learning methodologies exhibit potential efficacy in augmenting the performance of detection mechanisms within the IoT environment. Through the strategic integration and synthesis of diverse data sources, feature collections, or analytical frameworks, these methodologies produce a more holistic and resilient depiction of the data, thereby facilitating enhanced accuracy and dependability in the identification of malware assaults within the IoT environment. Table VI illustrates the pros and cons of recent research in the IoT environment.

TABLE VI. PROS AND CONS OF IOT RESEARCH

Ref.	Pros	Cons.	Summary
[68]	• Effective Detection: identifying both established and novel variants of (IoT) malware via dynamic analysis, thereby responding to the continuously evolving characteristics of malware threats. • Utilization of CNN: The application of the CNNs significantly augments the	• Evasion Tactics: complicating detection efforts, which Some malware may alter their behaviour when executed in controlled environments, • Implementation Complexity: The integration of various analysis techniques and ML models	This scholarly investigation substantially advances the domain of (IoT) security by proffering a comprehensive framework for the identification of novel and variant malware, employing sophisticated analytical methodologies, and

	understanding and categorization of malware behaviours, thereby elevating the precision of detection methodologies. • Behaviour Visualization: By converting behaviour data into images, the DAIMD allows for intuitive understanding and classification of malware behaviours. • Minimizes Damage: aims to reduce the potential damage from malware infections on IoT devices, which is critical as IoT technology expands.	can make the implementation of this paper complex. • Data Quality Dependence: The effectiveness of this paper relies on the quality of behaviour data collected; incomplete data can lead to inaccurate detection. • Resource Intensive: Dynamic analysis and processing large datasets for CNN training can be resource-heavy, requiring significant computational power.	tackling the distinct challenges presented by IoT devices.
[76]	• Data Privacy: he main featured of FL basically is safeguarding user privacy, which mean allows for training of models as decentralize without sharing sensitive data. • Robustness Against Attacks: This paper augmenting the comprehensive security of the system, thereby integrates resilient model aggregation methodologies that can alleviate the repercussions of adversarial incursions. • Performance: federated models are able to achieve performance metrics on par with those of centralized models while maintaining data privacy, a key requirement in sensitive contexts such as IoT. • Scalability and Collaborative Learning.: The architecture of FL inherently decentralized possesses the potential to scale effectively to support a substantial array of IoT devices, rendering it appropriate for forthcoming networks such as Beyond 5G (B5G). Furthermore, can collaboratively enhance the model by transmitting only update pertaining to the model rather than the raw data, which can subsequently foster improved generalization across heterogeneous data distributions.	• Complexity: led to inherent complexity, primarily due to the requisite synchronization and communication between the participating clients and the central server, which may lead to additional computational overhead. • Vulnerability: clients have the potential to introduce corrupted updates that adversely influence the performance of the global model. Therefore, led to a significant threat to the integrity of the mode. • Communication Costs: affect the operational efficiency of (IoT) devices, especially when updates the model are extensive or when a substantial number of clients are engaged. • Asynchronous challenges: Customer failures or delayed responses can significantly disrupt the training process,	This scholarly adopt FL enhances data privacy and model robustness while achieving performance comparable to that of centralized methods. However, it faces challenges such as exposure to malicious clients and implementation complexity, which can disrupt training. In addition, high communication costs and the need for precise synchronization can complicate deployment.
[79]	• Accuracy: shows 99.9% of accuracy in detect IoT malware, thus outperforming many state-of-the-art models in the field. • Ensemble learning approach: By utilizing a clustering technique that integrates three one-dimensional CNNs, it brilliantly captures the diverse patterns inherent in IoT network traffic, resulting in enhanced predictive reliability. • Optimization: Applying the (Grid-Search-CV) algorithm for hyperparameter optimization significantly improves the effectiveness of the model, enabling it to more skill fully navigate the complexities of IoT traffic. • processing time: this paper led to make model more efficient in terms of time and computational resources. by choosing one-dimensional CNNs over two-dimensional CNNs which simplifies the preprocessing phase. • Addressing IoT privacy: The model is specifically designed for IoT environments, recognizing that IoT traffic has unique characteristics compared to traditional internet traffic, which is critical for effective malware detection.	• Extended Execution Durations: A notable disadvantage of this model is that an increase in the quantity of hyperparameters can significantly prolong the execution time. This phenomenon may pose a considerable obstacle for individuals intending to implement the model in time-sensitive applications. • Dataset: the limitation of this paper is effectiveness of the model depend on the IoT-23 dataset, suggesting that its performance may not be as strong when applied to alternative datasets. • Complexity: this model architecture is quite complex due to its ensemble nature.	The paper suggests that further improvements, such as parallel processing and the use of Generative Adversarial Networks (GANs), are necessary to enhance the model's capabilities, indicating that the current model may not be fully optimized.
[80]	• Novel Approach: lead to unique contribution to the field through introduces a novel optimized ML method for IoT malware detection using visual representation of network traffic, • Effective Feature Selection: provided the SVM performance through reducing the number of selected features which is done using ant colony optimizer.	• Computational Cost: Despite the optimization of the method, all IoT environments characterized by resource constraints, therefore still the implementation of ML techniques incurs significant computational demands,	this scholarly presents an optimized (IoT) malware detection system with high accuracy and suitability for resource-constrained environments, thanks to ACO-based feature selection. However, it may still face computational overhead and limitation of dataset, potentially

	• Parameter Tuning: this paper enhancing the classifier's performance across different kernel functions through employs the PSO algorithm to tune SVM parameters, • Lightweight Solution: the novelty is the bioinspired techniques (ACO and PSO) contribute to building an effective and lightweight malware detection system suitable for IoT environment.	• Dataset Limitations: the generalizability of the results is affected through the use of the size and diversity of the dataset. • Transfer Learning Issues: avoids transfer learning due to issues like negative transfer and overfitting, which are common challenges in image classification tasks. • Lack of Standardization: There is no standardization in determining connected activities or algorithm decisions, making it difficult to address negative transfer effectively. • Overfitting Concerns: which is a common issue in prediction methods and can limit the model's applicability to new data.	affecting real-world applicability. Additionally, the complexity of integrating multiple algorithms could pose implementation challenges.
[81]	• Comprehensive Evaluation: evaluates two ML algorithms, (Random Forest and Naïve Bayes), providing a detailed comparison of their performance in detecting malware in IoT network traffic. • Real-World Dataset: The research utilizes the IoT-23 dataset, which is a labelled dataset containing IoT malware infection traffic and benign traffic, ensuring that the study is grounded in real-world data. • Practical Implications: recognize the effective of ML algorithms for detecting malware network flows, which is being references for help personalized cybersecurity Furthermore, the superlative performance of Random Forest over Naïve Bayes. • Focus on IoT Security: the focusing on IoT devices thus addresses a critical area of cybersecurity, which are often vulnerable due to their lack of security features and the increasing number of cyberattacks targeting them.	• dataset limitation: the restricted accuracy and their robustness of developed ML system. which lead to limited size of the used dataset and hardware. • Computational Constraints: The substantial dimensions of the IoT-23 dataset present significant computational difficulties attributable to constrained hardware resources, necessitating the implementation of subsampling techniques and a reduction in the number of malicious categories incorporated into the analytical process. • Limited Scope of Algorithms: the only considers that the supervised learning method, specifically (Random Forest and Naïve Bayes), without exploring the potential of unsupervised learning, which could provide additional insights into malware detection. • Hardware Limitations: affected the ability to process the full dataset thereby the device used has limited RAM and storage,	This scholarly effectively compares Random Forest and Naïve Bayes for IoT malware detection using a real-world dataset, providing practical insights for cybersecurity applications Nevertheless, it is constrained by computational limitations and exclusively investigates supervised learning methodologies, which may confine the breadth of its conclusions.
[88]	• Innovative Approach: the novelty of adopted AIS for detected malware in IoT environment, these approaches given the constraints of IoT devices in terms of computational power and memory. Furthermore, providing more applicable visionary of real-world environment through validate the AIS method in an IoT practical framework. • High Detection Accuracy: This paper reports a 91% detection accuracy in the most restricted IoT system, which is significant given the challenging conditions of IoT environments. • Transfer Learning: highlights the ability of AIS solutions to transfer learning between IoT devices, which is crucial for networks with highly constrained devices. • Comprehensive Evaluation: Through using multiple datasets with different types of malware attacks, this paper ensuring a thorough evaluation of the AIS solutions' performance.	• Simulation vs. Real-World Discrepancy: noted that a discrepancy between simulation results and real-world performance, with a projection from 99% to 91% as accuracy detection, thus indicating possibilities overestimation in simulated environments. • Limited Real-Time Testing: the NPS method has not tested on real time platform, this results in a constriction of the comprehension of its efficacy in real-world contexts. • High Memory Requirement: Some AIS methods reviewed, such as the MNSA, require significant memory to generate detectors, making them less suitable for IoT systems with limited resources. • Unclear Classification Details: highlight high detection rates but lacks clarity on the number of classes used for classification, which could affect the interpretation of results. • Dependency on AWS: relies on Amazon Web Services for simulating IoT environment, which may not fully replicate all aspects of real-world IoT environment.	This scholarly adopted AIS solutions effectively to detect unknown malware in IoT systems with a 91% accuracy, supporting transfer learning between devices, which is crucial for resource-limited networks. However, real-world accuracy is lower than the 99% reported in simulations, indicating challenges in practical applications and limitations of simulation data reliance.
[89]	• Enhanced Security for IoT Devices: introduces an advanced Elliptic Curve Cryptography (IECC) method that provides superior security for (IoT) devices throughout the process of data transmission in comparison to currently utilized techniques. • Efficient Malware Detection: The adopted of a Deep (LSTM) for the purpose of malware detection is underscored as a proficient	• Lack of Publication Details: in this paper may affect its traceability and credibility, which is not provide specific publication details like the publication date and publisher. • Limited Contextual Information: Some figures and tables referenced in the text are not provided, which may hinder a full understanding of the results and discussions.	This scholarly presents an improved Elliptic Curve Cryptography (IECC) algorithm that enhances IoT security and achieves high accuracy in malware detection, but lacks specific publication details and comprehensive contextual information, which may limit its applicability and understanding.

	methodology for the discernment of various categories of attacks within (IoT) networks, including but not limited to anomaly detection, the (DOS) attacks, probing activities, and Remote-to-Local intrusions. • Performance Metrics: The methodology put forth demonstrates a considerable degree of accuracy (95%), precision (92%), and specificity (96%), thereby underscoring its efficacy in the domains of malware identification and mitigation. • Optimized Cryptographic Process: The IECC algorithm demonstrates reduced processing and memory usage compared to other cryptographic techniques such as ECC, AES, DES, and Blowfish, making it more efficient for IoT applications. • No Competing Interests: ensuring the integrity of the research, that the authors affirm the absence of any conflicting financial interests,	• Focus on Specific Algorithms: While the paper discusses the IECC and Deep LSTM models, it may not cover other potential approaches or algorithms that could be relevant for IoT security, limiting the scope of the research. • Generalization of Results: does not discuss the potential limitations or challenges in generalizing the proposed approach to different IoT environments or scenarios, which could be important for practical applications	
[90]	• Enhanced Security: proposes an improved ECC algorithm and a deep LSTM model for malware detection, which enhances IoT device security during data transmission. • High Accuracy: achieves 95% accuracy, indicating its effectiveness in detecting and preventing malware attacks. • Efficient Performance: The improved ECC algorithm shows efficient encryption and decryption times, contributing to better security performance.	• Complexity: the implement of DL and cryptographic algorithms may require significant computational resources, which could be challenging for some IoT devices. • Limited Context: primarily focuses on IoT security, which may not address other potential vulnerabilities in different contexts or environments. • Post-Quantum Vulnerability: mentions that postquantum attacks can still succeed against the proposed methods, indicating a potential area for improvement.	the manuscript enhances IoT security with an improved (ECC) method and deep (LSTM) model, achieving 95% accuracy in malware detection. However, the complexity of implementing these models may pose challenges for resource-constrained IoT devices, and the approach may not address vulnerabilities in other contexts. Nevertheless, the acknowledges potential vulnerabilities to postquantum attacks.
[91]	• High Accuracy: The models achieve high accuracy in detecting malware, outperforming existing models. • Lightweight and Fast: CompactCBL (CNN-BiLSTM) is small and fast, suitable for real-time IoT applications. • Broad Applicability: Suitable for various IoT devices in smart cities, requiring real-time responses. • Innovative Architecture: Combines CNNs and BiLSTMs for effective malware detection.	• Limited Multiclass Detection: Improvement needed in detecting individual attack types. • Dataset Specific: Validated only on the CIC-Malmem-2022 dataset, requiring further real-world testing. • Resource Constraints: RobustCBL may still be challenging for extremely limited environments.	This scholarly presents a lightweight, efficient malware detection model suitable for IoT devices, demonstrating high performance on the CIC-Malmem-2022 dataset; however, it struggles with detailed multiclass attack detection and requires further enhancements for zero-day threats, with future plans to explore semi-supervised learning and real-world testing
[92]	• Enhanced Detection: The hybrid model improves accuracy by combining static and graph-based analysis, effectively handling complex and obfuscated malware. • Comprehensive Approach: this includes uses (both static and dynamic) methods for analysing various malware types, including novel variants. • Efficient Classification: Signature generation through clustering aids in accurate malware classification. • Obfuscation Handling: The model addresses the challenge of malware obfuscation, a common issue in IoT malware detection.	• Obfuscation Limitations: Static analysis may struggle with obfuscated or encrypted malware, reducing effectiveness. • Sample Scarcity: A lack of IoT malware samples can impede robust model development and testing. • Complexity and Resources: Graph-based analysis, while effective, can be complex and resource-intensive, limiting real-time detection capabilities. • Complexity and Resources: Graph-based analysis, while effective, can be complex and resource-intensive, limiting real-time detection capabilities.	this scholarly provided the hybrid IoT malware detection model that enhances accuracy by combining static and graph-based analysis, effectively handling complex malware, but struggles with obfuscated samples and requires significant resources for graph-based methods.
[93]	• Privacy and Security: Utilizes group-oblivious sign-crypton to ensure data privacy and meets all security requirements for FL. • Incentive Mechanism: The Stackelberg incentive model is designed to encourage IoT device participation, which helps in reducing dropout rates during model updates. • Security Assurance: The security analysis confirms that PPFL-SC meets all necessary security requirements for privacy-preserving FL, making it robust against potential attacks	• Computational Overhead: The encryption operations, while preserving performance, still contribute to computational costs, particularly as model parameters increase. • Communication Overhead: the increasing number of devices in proposed method can lead to higher communication costs for the CC-server, which might affect scalability. • Device Dropout Risks: Although the paper addresses dropout issues, the related work	this scholarly present the PPFL-SC model offers significant advantages, such as enhanced privacy through group-oblivious sign-crypton, which ensures data security during FL, and the Stackelberg incentive model, which boosts IoT device participation by encouraging collaboration with diverse datasets, thereby improving the robustness of malware detection models. However, the model may

	• Efficiency and Practicality: Empirical assessments carried out on authentic datasets corroborate the operational efficacy and practicality of the proposed model, thus illustrating its significance in real-world applications. • Lower Communication Cost: Compared to the plaintext Fed-AVG protocol, the proposed scheme is more efficient with lower communication costs, even as the number of gradients increases.	section highlights that dropout can still pose risks to confidentiality and system efficacy. • Complexity of Implementation: The integration of advanced cryptographic techniques and incentive mechanisms may increase the complexity of implementation and require significant resources. • Limited Dataset Sources: The paper notes that current malware detection systems often rely on limited and static datasets, which can quickly become outdated.	encounter challenges like computational overhead during the verification phase and potential issues from device dropouts, which could impact the efficiency and continuity of the FL process
[98]	• Enhanced Detection Accuracy: this paper Mal3S achieves an average detection accuracy of 98.02% coupled with a classification accuracy of 98.43%, thus exceeding current malware detection techniques. • Static Feature Utilization: The approach effectively uses five types of static features (byte, opcode, API calls, DLL, and string) to detect malware without executing the files, which allows for quick and accurate detection. • Image Generation for Analysis: By converting static features into images of various sizes, Mal3S maintains spatial information, which aids in the detailed analysis and classification of malware, including obfuscated variants. • Capability: exhibited a significant capacity for generalization across a wide range of malware classifications, thus validated through the utilization of three separate datasets.	• Static Analysis Limitations: The approach relies on static analysis, which can struggle with detecting complex malware patterns and variants due to its dependence on fixed code signatures. • Obfuscation Challenges: Malware that uses obfuscation techniques may hide key behavioural information, leading to the generation of less meaningful static features. • Lack of Dynamic Analysis: The method does not incorporate dynamic analysis, which could enhance the detection of real-time behaviours in obfuscated malware. • Complex Feature Representation: Managing and analysing the extensive behavioural information from static features requires a complex, multidimensional approach.	this scholarly adopted Mal3S which is a malware detection system designed for Smart IoT environments, utilizing a multi SPP-net model to analyse static features like bytes, opcodes, and API calls. It effectively detects various malware types, including obfuscated ones, by compensating for missing static feature information with other features. demonstrating strong generalization across different malware datasets
[99]	• Privacy-Preserving: The SIM-FED model enhances security by detecting malware without sharing data, thus preserving privacy in distributed environments. • High Accuracy: this approach achieves a remarkable 99.52% accuracy, outperforming other models in all evaluation metrics. • Robustness: shows resilience against both white-box and black-box cyber-attacks, with minimal performance degradation • Scalability: this paper maintains stable performance even when the number of clients increases, thus indicating good scalability.	• Complexity: the Implementing FL models like SIM-FED can be complex due to the need for coordination among multiple clients. • Resource Intensive: Although optimized, the model may still require significant computational resources for training and maintaining the global model. • Dependency on Client Participation: The performance models can be affected by the variability in client participation and data quality	the scholarly present the SIM-FED model offers privacy-preserving malware detection with high accuracy and robustness against cyber-attacks, while maintaining scalability with varying client numbers. However, it may be complex to implement, resource-intensive, and dependent on client participation quality.

5. CHALLENGES AND LIMITATIONS

Although the research discussed in this review introduces novel methodologies for identifying malware within IoT environments, several prevalent challenges and limitations are simultaneously encountered.

Malware detection in IoT systems is difficult because of device variety and resource restrictions. The enormous diversity of IoT devices and platforms, many of which lack security standards, is a major hurdle. Variability hinders the creation of a universal malware detection system. For IoT devices, computing performance, storage, and energy resources might hinder resource-intensive detection algorithms such as ML, DL, and FL.

IoT devices create large amounts and diverse data, making malware detection program datasets difficult to train. To ensure data quality and usefulness, substantial preprocessing is frequently needed. Owing to expert knowledge and human tagging, supervised learning data labelling may be complex and expensive.

Integrating FL, DL, and ML has constraints. Solution prospects are also plentiful. Malware is complicated and ever-changing, making feature extraction and selection difficult for ML systems. DL can handle complex tasks, but it requires large labelled datasets and considerable computing power, which may be challenging in IoT environments. The opaqueness of DL algorithms raises concerns about their interpretability and transparency, which are essential for understanding and decreasing malware threats.

FL can help preserve the privacy of data by training models on the edge, but it also faces challenges in the IoT environment. Data privacy is a problem, as the model to be updated can be infused and the data can be compromised by the user. Since attackers might disrupt FL, data transmitted between devices must be secure and reliable and therefore need secure

aggregation and the use of encryption techniques. Moreover, as mentioned above, the IoT devices and their data are heterogeneous, which creates challenges for model generalization requiring an adaptive algorithm. Furthermore, communication is another issue that can be improved through model compression and communication-efficient solutions, particularly for resource-constrained devices. Finally, the 'limited computation' and 'energy resources' of IoT devices require lightweight and efficient FL implementations. Tackling these issues is very important for realizing the potential of FL in IoT applications. Owing to the variety of IoT devices, not all devices will share equally during training, which might lead to skewed models.

In addition, establishing and sustaining these learning methods in the IoT is difficult. Upgrades and patches to new detection methods for IoT devices with no connectivity are logistically difficult. Owing to the constant change of malware, detection models must be updated; however, this may be resource intensive and may not be practical for many IoT installations. It will be a while before all detection systems and devices are compatible and interoperable.

In conclusion, ML, DL, and FL confront unique challenges when identifying IoT malware. These complexities require a comprehensive approach that includes data preparation, model enhancements, and IoT device security and interoperability. The IoT ecosystem's rising malware threat can only be addressed with a comprehensive plan such as this one. For more clarity, the following points summarize the limitations and challenges according to the specific environment within the IoT:

- Dataset Limitations: The dataset used may not represent all malware variations or new attack vectors.
- Resource Constraints: Deploying DL models is complex because of the limited processing power of IoT devices.
- Real-Time Constraints: Detection systems must operate with minimal latency in real-world IoT systems.
- Device Diversity: The heterogeneity of IoT devices complicates model generalization.
- Evolving Threats: IoT malware evolves rapidly, requiring continuous model updates.

6. FUTURE WORK AND RESEARCH DIRECTIONS

Even if the academic literature has improved, many promising areas for IoT malware detection research still exist. Future research should concentrate on strengthening FL, ML, and DL frameworks against sophisticated malware attacks. These models increase IoT security by using adaptive learning and advanced anomaly detection to respond to new threats. Monitoring and updating models to keep ahead of new attack vectors makes IoT systems more resilient.

Real-time malware detection technologies are essential. Researchers can rapidly and successfully construct systems that identify and eradicate new threats via DL architectures. Processing power and algorithm efficiency must increase for the IoT to perform real-time analysis without losing speed or resources.

Data privacy and integrity are crucial when IoT security uses FL frameworks. Researchers should improve cryptography and secure multiparty computing for safe and reliable FL. Protocols must be created to keep the data secure and secret when they are spread over several devices, some of which may be unreliable.

Owing to the exponential proliferation of IoT devices, future research must address scalability issues. DL and ML techniques that can manage massive volumes of data from various IoT networks are essential. implement that does not overload computers. Making models that are not too heavy for low-resource devices ensures wide usage without straining computers.

To win over consumers and fulfil legal standards, IoT-based learning systems must be clear and easy to explain. Future research should focus on creating interpretable models that explain decision-making processes. If we can make malware detection methods more visible, users can understand and hold security measures responsible.

IoT-based collaborative detection networks may dramatically increase malware detection. Researchers may create more accurate detection frameworks by simplifying data exchange and inter-device connections. Further work should address synchronization and coordination issues to ensure that these collaborative frameworks operate well.

Adversarial training in IoT learning frameworks is a key topic of research. Researchers can train models to withstand complex malware attacks via hostile instances. This method makes it simpler to develop safer IoT systems that can survive more complicated assaults, ensuring long-term stability and security.

Future research should combine IoT-based learning methods with blockchain and edge computing. Blockchain technology may make data transfers more secure and transparent, whereas edge computing can speed up data processing. Combining these technologies with ML, DL, and FL may improve and secure IoT ecosystems.

As the IoT grows, so will the need for advanced and effective virus detection systems. Researchers can develop more resilient IoT security frameworks by improving model robustness, implementing real-time detection systems, secure FL,

finding scalable solutions, explaining their work, forming collaborative detection networks, training adversaries, and integrating other technologies. To sustain future IoT ecosystems, these efforts are crucial.

In the future, exploring a case study that uses the IoT-23 dataset can be beneficial. This highlights the real-world usability of malware detection techniques in IoT systems. The IoT-23 dataset includes labelled IoT network traffic data for normal and malicious activities and provides an excellent opportunity to evaluate malware detection methods. After the anomaly detection model's training through DL methods using the dataset, an assessment of its performance takes place in different scenarios, which encompasses, although not exclusively, smart home networks and industrial IoT systems. The model possesses the ability to identify and mitigate (Mirai botnet) activities through the analysis of traffic patterns associated with smart devices, subsequently issuing alerts to the user in the event of a potential security breach. By deploying the trained model on edge devices, such as IoT gateways, the system can effectively provide real-time detection of threats while tackling IoT-specific challenges such as low computational resources. This method shows how to use the techniques, building a bridge from analysis to actual use for IoT security. By demonstrating the successful execution of the suggested methods, such case studies lend strength to their effectiveness as well.

7. CONCLUSION

This all-encompassing assessment concludes that FL, ML, and DL have great promise for identifying malware in a wide range of systems, including IoT settings. Although these approaches have considerable potential, many studies have focused on making them more generalizable, creating algorithms that use fewer resources, and making detection systems better suited to address new threats. By integrating complementary information from many data sources, learning approaches have shown great promise in improving malware detection accuracy and robustness. To effectively handle the complexity and variety of IoT data while maintaining high computational efficiency, further research is needed to determine the best feature techniques. The formation of progressive and resilient security frameworks—assuring the integrity and dependability of linked systems via intricate approaches such as ML, DL, and FL—requires addressing the previously highlighted research limits. Academic research must advance at the intersection of these fields if we are to build a resilient infrastructure to withstand the inevitable challenges posed by the IoT in the future, where we can maximize the benefits of extensive connectivity while mitigating the risks posed by bad actors.

Conflicts of interest

The authors declare that they have no conflicts of interest.

Funding

None.

Acknowledgement

The authors would like to thank Mustansiriyah University (https://uomustansiriyah.edu.iq/) in Baghdad-Iraq for its support in the present work.

References

[1] A. Nuhu, A. F. M. Raffei, M. F. Ab Razak, and A. Ahmad, “Distributed Denial of Service Attack Detection in IoT Networks using Deep Learning and Feature Fusion: A Review,” *Mesopotamian J. CyberSecurity*, vol. 4, no. 1, pp. 47–70, 2024.

[2] M. N. Jasim and M. T. Gaata, “K-Means clustering-based semi-supervised for DDoS attacks classification,” *Bull. Electr. Eng. Informatics*, vol. 11, no. 6, pp. 3570–3576, 2022.

[3] B. A. Jaafar, M. T. Gaata, and M. N. Jasim, “Home appliances recommendation system based on weather information using combined modified k-means and elbow algorithms,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 19, no. 3, pp. 1635–1642, 2020.

[4] M. Riyadh and D. R. Alshibani, “Intrusion detection system based on machine learning techniques,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 23, no. 2, pp. 953–961, 2021.

[5] M. M. Mijwil, R. Doshi, K. K. Hiran, A.-H. Al-Mistarehi, and M. Gök, “Cybersecurity Challenges in Smart Cities: An Overview and Future Prospects,” *Mesopotamian J. cybersecurity*, vol. 2022, pp. 1–4, 2022.

[6] M. Mijwil, Y. Filali, M. Aljanabi, M. Bounabi, and H. Al-Shahwani, “The purpose of cybersecurity governance in the digital transformation of public services and protecting the digital environment,” *Mesopotamian J. cybersecurity*, vol. 2023, pp. 1–6, 2023.

[7] D. Singh and S. Khurana, “Malware Detection in IoT Devices Using Machine Learning: A Review,” in *2024 International Conference on Computational Intelligence and Computing Applications (ICCICA)*, IEEE, 2024, pp.

203–209.
[8] S. S. Hussain, M. F. Ab Razak, and A. Firdaus, “Deep Learning Based Hybrid Analysis of Malware Detection and Classification: A Recent Review,” *J. Cyber Secur. Mobil.*, pp. 91–134, 2024.
[9] J. BOODAI, A. ALQAHTANI, and K. RIAD, “DEEP LEARNING FOR MALWARE DETECTION: LITERATURE REVIEW,” *J. Theor. Appl. Inf. Technol.*, vol. 102, no. 4, 2024.
[10] M. Ghahramani, R. Taheri, M. Shojafar, R. Javidan, and S. Wan, “Deep Image: A precious image based deep learning method for online malware detection in IoT Environment,” *Internet of Things*, p. 101300, 2024.
[11] A. Brown, M. Gupta, and M. Abdelsalam, “Automated machine learning for deep learning based malware detection,” *Comput. Secur.*, vol. 137, p. 103582, 2024.
[12] F. Mofidi, S. G. Hounsinou, and G. Bloom, “L-IDS: A Multi-Layered Approach to Ransomware Detection in IoT,” in *2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, 2024, pp. 387–396.
[13] A. Javed, I. Rashid, S. Tahir, S. Saeed, A. M. Almuhaideb, and K. Alissa, “AdamW+: Machine Learning Framework to Detect Domain Generation Algorithms for Malware,” *IEEE Access*, 2024.
[14] M. Riyadh, B. J. Ali, and D. R. Alshibani, “IDS-MIU: An Intrusion Detection System Based on Machine Learning Techniques for Mixed type, Incomplete, and Uncertain Data Set.,” *Int. J. Intell. Eng. Syst.*, vol. 14, no. 3, 2021.
[15] N. Z. Gorment, A. Selamat, L. K. Cheng, and O. Krejcar, “Machine learning algorithm for malware detection: taxonomy, current challenges and future directions,” *IEEE Access*, 2023.
[16] C. P. Chenet, A. Savino, and S. Di Carlo, “A survey on hardware-based malware detection approaches,” *IEEE Access*, 2024.
[17] M. U. Rana, M. A. Shah, and O. Ellahi, “Malware Persistence and Obfuscation: An Analysis on Concealed Strategies,” in *2021 26th International Conference on Automation and Computing (ICAC)*, IEEE, 2021, pp. 1–6.
[18] M. Luoma-aho, “Analysis of Modern Malware: obfuscation techniques,” 2023.
[19] K. Brezinski and K. Ferens, “Metamorphic malware and obfuscation: a survey of techniques, variants, and generation kits,” *Secur. Commun. Networks*, vol. 2023, 2021.
[20] C. Catalano, G. Specchia, and N. G. Totaro, “Enhancing Code Obfuscation Techniques: Exploring the Impact of Artificial Intelligence on Malware Detection,” in *International Conference on Product-Focused Software Process Improvement*, Springer, 2023, pp. 80–88.
[21] H. J. Asghar *et al.*, “Use of cryptography in malware obfuscation,” *J. Comput. Virol. Hacking Tech.*, vol. 20, no. 1, pp. 135–152, 2024.
[22] J. R. S. Alrzini and D. Pennington, “A review of polymorphic malware detection techniques,” *Int. J. Adv. Res. Eng. Technol.*, vol. 11, no. 12, pp. 1238–1247, 2020.
[23] M. A. Khoje, A. Kaswan, and J. Pawar, “Exploring Polymorphic Malware Analysis Techniques: A Comprehensive Survey”.
[24] R. Hakobyan and T. Jamgharyan, “Polymorphic Malware Analysis Model”.
[25] M. Gopinath and S. C. Sethuraman, “A comprehensive survey on deep learning based malware detection techniques,” *Comput. Sci. Rev.*, vol. 47, p. 100529, 2023.
[26] D. Dholariya, D. Panchal, and P. Singhal, “Anti-Virus Tempering Methodologies,” *Int. Res. J. Adv. Eng. Manag.*, vol. 2, no. 04, pp. 1232–1239, 2024.
[27] V. Priya and A. S. Sofia, “Review on Malware Classification and Malware Detection Using Transfer Learning Approach,” in *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, IEEE, 2023, pp. 1042–1049.
[28] Y. T. Ling, N. F. M. Sani, M. T. Abdullah, and N. A. W. A. Hamid, “Metamorphic malware detection using structural features and nonnegative matrix factorization with hidden markov model,” *J. Comput. Virol. Hacking Tech.*, pp. 1–21, 2022.
[29] A. Kashtalian, S. Lysenko, O. Savenko, A. Nicheporuk, T. Sochor, and V. Avsiyevych, “Multi-computer malware detection systems with metamorphic functionality,” *Radioelectron. Comput. Syst.*, vol. 2024, no. 1, pp. 152–175, 2024.
[30] F. Habib, S. H. Shirazi, K. Aurangzeb, A. Khan, B. Bhushan, and M. Alhussein, “Deep Neural Networks for Enhanced Security: Detecting Metamorphic Malware in IoT Devices,” *IEEE Access*, 2024.
[31] K. O. Babaagba and J. Wylie, “An Evolutionary based Generative Adversarial Network Inspired Approach to Defeating Metamorphic Malware,” in *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, 2023, pp. 1753–1759.
[32] A. Bensaoud, J. Kalita, and M. Bensaoud, “A survey of malware detection using deep learning,” *Mach. Learn. With Appl.*, vol. 16, p. 100546, 2024.
[33] B. Ait Messaad, K. Chetioui, Y. Balboul, and H. Rhachi, “Analyzing and Detecting Malware Using Machine

Learning and Deep Learning," in *The International Conference on Artificial Intelligence and Smart Environment*, Springer, 2023, pp. 518–525.
[34] M. S. Akhtar and T. Feng, "Evaluation of machine learning algorithms for malware detection," *Sensors*, vol. 23, no. 2, p. 946, 2023.
[35] M. J. H. Faruk *et al.*, "Malware detection and prevention using artificial intelligence techniques," in *2021 IEEE International Conference on Big Data (Big Data)*, IEEE, 2021, pp. 5369–5377.
[36] A.-A. M. Majid, A. J. Alshaibi, E. Kostyuchenko, and A. Shelupanov, "A review of artificial intelligence based malware detection using deep learning," *Mater. Today Proc.*, vol. 80, pp. 2678–2683, 2023.
[37] A. Djenna, A. Bouridane, S. Rubab, and I. M. Marou, "Artificial intelligence-based malware detection, analysis, and mitigation," *Symmetry (Basel).*, vol. 15, no. 3, p. 677, 2023.
[38] D. Gibert, C. Mateu, and J. Planes, "The rise of machine learning for detection and classification of malware: Research developments, trends and challenges," *J. Netw. Comput. Appl.*, vol. 153, p. 102526, 2020.
[39] S. A. Roseline, S. Geetha, S. Kadry, and Y. Nam, "Intelligent vision-based malware detection and classification using deep random forest paradigm," *IEEE Access*, vol. 8, pp. 206303–206324, 2020.
[40] F. Deldar and M. Abadi, "Deep learning for zero-day malware detection and classification: a survey," *ACM Comput. Surv.*, vol. 56, no. 2, pp. 1–37, 2023.
[41] A. Abusitta, M. Q. Li, and B. C. M. Fung, "Malware classification and composition analysis: A survey of recent developments," *J. Inf. Secur. Appl.*, vol. 59, p. 102828, 2021.
[42] V. Khushali, "A Review on Fileless Malware Analysis Techniques," *Int. J. Eng. Res. Technol.*, vol. 9, no. 05, 2020.
[43] R. Baker del Aguila, C. D. Contreras Pérez, A. G. Silva-Trujillo, J. C. Cuevas-Tello, and J. Nunez-Varela, "Static Malware Analysis Using Low-Parameter Machine Learning Models," *Computers*, vol. 13, no. 3, p. 59, 2024.
[44] M. A. Ayub, A. Siraj, B. Filar, and M. Gupta, "RWArmor: a static-informed dynamic analysis approach for early detection of cryptographic windows ransomware," *Int. J. Inf. Secur.*, vol. 23, no. 1, pp. 533–556, 2024.
[45] S. Talukder, "Tools and techniques for malware detection and analysis," *arXiv Prepr. arXiv2002.06819*, 2020.
[46] J. A. Herrera-Silva and M. Hernández-Álvarez, "Dynamic feature dataset for ransomware detection using machine learning algorithms," *Sensors*, vol. 23, no. 3, p. 1053, 2023.
[47] G. Karat, J. M. Kannimoola, N. Nair, A. Vazhayil, V. G. Sujadevi, and P. Poornachandran, "CNN-LSTM Hybrid Model for Enhanced Malware Analysis and Detection," *Procedia Comput. Sci.*, vol. 233, pp. 492–503, 2024.
[48] F. A. Aboaoja, A. Zainal, F. A. Ghaleb, B. A. S. Al-rimy, T. A. E. Eisa, and A. A. H. Elnour, "Malware detection issues, challenges, and future directions: A survey," *Appl. Sci.*, vol. 12, no. 17, p. 8482, 2022.
[49] V. Vasani, A. K. Bairwa, S. Joshi, A. Pljonkin, M. Kaur, and M. Amoon, "Comprehensive analysis of advanced techniques and vital tools for detecting malware intrusion," *Electronics*, vol. 12, no. 20, p. 4299, 2023.
[50] U.-H. Tayyab, F. B. Khan, M. H. Durad, A. Khan, and Y. S. Lee, "A survey of the recent trends in deep learning based malware detection," *J. Cybersecurity Priv.*, vol. 2, no. 4, pp. 800–829, 2022.
[51] D. L. S. Punyasiri, "Signature & Behavior Based Malware Detection," 2023.
[52] L. S. Fascí, M. Fisichella, G. Lax, and C. Qian, "Disarming visualization-based approaches in malware detection systems," *Comput. Secur.*, vol. 126, p. 103062, 2023.
[53] K. S. Sangher, A. Singh, and H. M. Pandey, "Signature based ransomware detection based on optimizations approaches using RandomClassifier and CNN algorithms," *Int. J. Syst. Assur. Eng. Manag.*, pp. 1–17, 2023.
[54] A. Samad, "Hybrid Approaches in Threat Detection: Integrating Traditional Signature-Based Methods with AI and ML Techniques for Enhanced Accuracy," 2023.
[55] J. Raghunath, S. Kiran, G. S. N. Rao, J. R. A. Kumar, R. Anasuya, and C. S. Kumar, "A MACHINE LEARNING TECHNIQUE TO DETECT BEHAVIOR BASED MALWARE," *Semicond. Optoelectron.*, vol. 42, no. 1, pp. 1268–1278, 2023.
[56] T. Alsmadi and N. Alqudah, "A Survey on malware detection techniques," in *2021 International Conference on Information Technology (ICIT)*, IEEE, 2021, pp. 371–376.
[57] S. K. Sahay, A. Sharma, and H. Rathore, "Evolution of malware and its detection techniques," in *Information and Communication Technology for Sustainable Development: Proceedings of ICT4SD 2018*, Springer, 2020, pp. 139–150.
[58] S. Soja Rani and S. R. Reeja, "A survey on different approaches for malware detection using machine learning techniques," in *Sustainable Communication Networks and Application: ICSCN 2019*, Springer, 2020, pp. 389–398.
[59] Ö. A. Aslan and R. Samet, "A comprehensive review on malware detection approaches," *IEEE access*, vol. 8, pp. 6249–6271, 2020.
[60] Q.-D. Ngo, H.-T. Nguyen, V.-H. Le, and D.-H. Nguyen, "A survey of IoT malware and detection methods based on static features," *ICT Express*, vol. 6, no. 4, pp. 280–286, 2020.
[61] H. Alrubayyi, G. Goteng, M. Jaber, and J. Kelly, "Challenges of malware detection in the IoT and a review of

artificial immune system approaches," *J. Sens. Actuator Networks*, vol. 10, no. 4, p. 61, 2021.
[62] R. Qamar and B. A. Zardari, "An analysis of the Internet of Everything," *Mesopotamian J. CyberSecurity*, vol. 2023, pp. 85–92, 2023.
[63] N. A. Bajao and J. Sarucam, "Threats Detection in the Internet of Things Using Convolutional neural networks, long short-term memory, and gated recurrent units," *Mesopotamian J. cybersecurity*, vol. 2023, pp. 22–29, 2023.
[64] C. C. Uchenna, N. Jamil, R. Ismail, L. K. Yan, and M. A. Mohamed, "Malware threat analysis techniques and approaches for iot applications: A review," *Bull. Electr. Eng. Informatics*, vol. 10, no. 3, pp. 1558–1571, 2021.
[65] S. K. Jagatheesaperumal, M. Rahouti, K. Ahmad, A. Al-Fuqaha, and M. Guizani, "The duo of artificial intelligence and big data for industry 4.0: Applications, techniques, challenges, and future research directions," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 12861–12885, 2021.
[66] A. A. Almazroi, M. A. Alqarni, M. A. Al-Shareeda, and S. Manickam, "L-CPPA: Lattice-based conditional privacy-preserving authentication scheme for fog computing with 5G-enabled vehicular system," *PLoS One*, vol. 18, no. 10, p. e0292690, 2023.
[67] J. E. M. Díaz, "Internet of things and distributed denial of service as risk factors in information security," in *Bioethics in Medicine and Society*, IntechOpen, 2020.
[68] J. Jeon, J. H. Park, and Y.-S. Jeong, "Dynamic analysis for IoT malware detection with convolution neural network model," *IEEE Access*, vol. 8, pp. 96899–96911, 2020.
[69] E. Amer and I. Zelinka, "A dynamic Windows malware detection and prediction method based on contextual understanding of API call sequence," *Comput. Secur.*, vol. 92, p. 101760, 2020.
[70] D. Rabadi and S. G. Teo, "Advanced windows methods on malware detection and classification," in *Annual Computer Security Applications Conference*, 2020, pp. 54–68.
[71] X. Liu, Y. Lin, H. Li, and J. Zhang, "A novel method for malware detection on ML-based visualization technique," *Comput. Secur.*, vol. 89, p. 101682, 2020.
[72] J. Hemalatha, S. A. Roseline, S. Geetha, S. Kadry, and R. Damaševičius, "An efficient densenet-based deep learning model for malware detection," *Entropy*, vol. 23, no. 3, p. 344, 2021.
[73] X. Huang, L. Ma, W. Yang, and Y. Zhong, "A method for windows malware detection based on deep learning," *J. Signal Process. Syst.*, vol. 93, pp. 265–273, 2021.
[74] N. A. Azeez, O. E. Odufuwa, S. Misra, J. Oluranti, and R. Damaševičius, "Windows PE malware detection using ensemble learning," in *Informatics*, MDPI, 2021, p. 10.
[75] Ö. Aslan, M. Ozkan-Okay, and D. Gupta, "Intelligent behavior-based malware detection system on cloud computing environment," *IEEE Access*, vol. 9, pp. 83252–83271, 2021.
[76] V. Rey, P. M. S. Sánchez, A. H. Celdrán, and G. Bovet, "Federated learning for malware detection in IoT devices," *Comput. Networks*, vol. 204, p. 108693, 2022.
[77] K. Shaukat, S. Luo, and V. Varadharajan, "A novel deep learning-based approach for malware detection," *Eng. Appl. Artif. Intell.*, vol. 122, p. 106030, 2023.
[78] M. Nobakht, R. Javidan, and A. Pourebrahimi, "DEMD-IoT: A deep ensemble model for IoT malware detection using CNNs and network traffic," *Evol. Syst.*, vol. 14, no. 3, pp. 461–477, 2023.
[79] S. H. Khan *et al.*, "A new deep boosted CNN and ensemble learning based IoT malware detection," *Comput. Secur.*, vol. 133, p. 103385, 2023.
[80] A. El-Ghamry, T. Gaber, K. K. Mohammed, and A. E. Hassanien, "Optimized and efficient image-based IoT malware detection method," *Electronics*, vol. 12, no. 3, p. 708, 2023.
[81] Y. Z. Wei, M. Md-Arshad, A. A. Samad, and N. Ithnin, "Comparing Malware Attack Detection using Machine Learning Techniques in IoT Network Traffic," *Int. J. Innov. Comput.*, vol. 13, no. 1, pp. 21–27, 2023.
[82] E. S. Alomari *et al.*, "Malware detection using deep learning and correlation-based feature selection," *Symmetry (Basel).*, vol. 15, no. 1, p. 123, 2023.
[83] P. Maniriho, A. N. Mahmood, and M. J. M. Chowdhury, "API-MalDetect: Automated malware detection framework for windows based on API calls and deep learning techniques," *J. Netw. Comput. Appl.*, vol. 218, p. 103704, 2023.
[84] S. Alsubai, A. K. Dutta, A. M. Alnajim, R. Ayub, A. M. AlShehri, and N. Ahmad, "Artificial intelligence-driven malware detection framework for internet of things environment," *PeerJ Comput. Sci.*, vol. 9, p. e1366, 2023.
[85] A. Almaleh, R. Almushabb, and R. Ogran, "Malware API Calls Detection Using Hybrid Logistic Regression and RNN Model," *Appl. Sci.*, vol. 13, no. 9, p. 5439, 2023.
[86] X. Yang, D. Yang, and Y. Li, "A Hybrid Attention Network for Malware Detection Based on Multi-Feature Aligned and Fusion," *Electronics*, vol. 12, no. 3, p. 713, 2023.
[87] M. N. Al-Andoli, K. S. Sim, S. C. Tan, P. Y. Goh, and C. P. Lim, "An ensemble-based parallel deep learning classifier with PSO-BP optimization for malware detection," *IEEE Access*, 2023.

[88] H. Alrubayyi, G. Goteng, and M. Jaber, "AIS for Malware Detection in a Realistic IoT System: Challenges and Opportunities," *Network*, vol. 3, no. 4, pp. 522–537, 2023.
[89] R. A. Devi and A. R. Arunachalam, "Enhancement of IoT device security using an Improved Elliptic Curve Cryptography algorithm and malware detection utilizing deep LSTM," *High-Confidence Comput.*, vol. 3, no. 2, p. 100117, 2023.
[90] S. S. Shafin, G. Karmakar, and I. Mareels, "Obfuscated memory malware detection in resource-constrained IoT devices for smart city applications," *Sensors*, vol. 23, no. 11, p. 5348, 2023.
[91] A. Khan, G. Choudhary, S. K. Shandilya, D. M. Sharma, and A. K. Sharma, "A hybrid mechanism for advance IoT malware detection," in *International Conference on IoT, Intelligent Computing and Security: Select Proceedings of IICS 2021*, Springer, 2023, pp. 247–259.
[92] A. Alamer, "A privacy-preserving federated learning with a secure collaborative for malware detection models using Internet of Things resources," *Internet of Things*, vol. 25, p. 101015, 2024.
[93] T. Lei, J. Xue, Y. Wang, T. Baker, and Z. Niu, "An empirical study of problems and evaluation of IoT malware classification label sources," *J. King Saud Univ. Inf. Sci.*, vol. 36, no. 1, p. 101898, 2024.
[94] K. Shaukat, S. Luo, and V. Varadharajan, "A novel machine learning approach for detecting first-time-appeared malware," *Eng. Appl. Artif. Intell.*, vol. 131, p. 107801, 2024.
[95] M. Azeem, D. Khan, S. Iftikhar, S. Bawazeer, and M. Alzahrani, "Analyzing and comparing the effectiveness of malware detection: A study of machine learning approaches," *Heliyon*, vol. 10, no. 1, 2024.
[96] I. T. Ahmed, B. T. Hammad, and N. Jamil, "A Comparative Performance Analysis of Malware Detection Algorithms Based on Various Texture Features and Classifiers," *IEEE Access*, 2024.
[97] W. Al-Khater and S. Al-Madeed, "Using 3D-VGG-16 and 3D-Resnet-18 deep learning models and FABEMD techniques in the detection of malware," *Alexandria Eng. J.*, vol. 89, pp. 39–52, 2024.
[98] J. Jeon, B. Jeong, S. Baek, and Y.-S. Jeong, "Static Multi Feature-Based Malware Detection Using Multi SPP-net in Smart IoT Environments," *IEEE Trans. Inf. Forensics Secur.*, 2024.
[99] M. Nobakht, R. Javidan, and A. Pourebrahimi, "SIM-FED: Secure IoT malware detection model with federated learning," *Comput. Electr. Eng.*, vol. 116, p. 109139, 2024.