Journal of
Sensor and Actuator Networks

MDPI

Review

Challenges of Malware Detection in the IoT and a Review of Artificial Immune System Approaches

Hadeel Alrubayyi *, Gokop Goteng, Mona Jaber and James Kelly

School of Electrical Engineering and Computer Science, Queen Mary University of London, London E1 4NS, UK; g.l.goteng@qmul.ac.uk (G.G.); m.jaber@qmul.ac.uk (M.J.); j.kelly@qmul.ac.uk (J.K.)
* Correspondence: h.s.alrubayyi@qmul.ac.uk

Abstract: The fast growth of the Internet of Things (IoT) and its diverse applications increase the risk of cyberattacks, one type of which is malware attacks. Due to the IoT devices' different capabilities and the dynamic and ever-evolving environment, applying complex security measures is challenging, and applying only basic security standards is risky. Artificial Immune Systems (AIS) are intrusion-detecting algorithms inspired by the human body's adaptive immune system techniques. Most of these algorithms imitate the human's body B-cell and T-cell defensive mechanisms. They are lightweight, adaptive, and able to detect malware attacks without prior knowledge. In this work, we review the recent advances in employing AIS for the improved detection of malware in IoT networks. We present a critical analysis that highlights the limitations of the state-of-the-art in AIS research and offer insights into promising new research directions.

Keywords: adaptive immunology; Artificial Immune Systems (AIS); Internet of Things (IoT); malware detection; security

Citation: Alrubayyi, H.; Goteng, G.; Jaber, M.; Kelly, J. Challenges of Malware Detection in the IoT and a Review of Artificial Immune System Approaches. *J. Sens. Actuator Netw.* **2021**, *10*, 61. https://doi.org/10.3390/jsan10040061

Academic Editor: Lei Shu

Received: 30 July 2021
Accepted: 14 October 2021
Published: 26 October 2021

1. Introduction

Today's world is more connected than ever before. Societies are reliant on technology, which has become inextricable from people's daily lives. For instance, smart cities, smart homes, and e-government are examples of data-driven technologies enabled by the Internet of Things (IoT) paradigm. Today's situation due to the Coronavirus 2019 (COVID-19) pandemic has accelerated the adoption of these technologies in various ways. For instance, e-health applications have developed to support the depleted health care staff and systems [1]. The widespread connectivity to the cyber-world increases the risk of cyber-attacks and hence may expose data previously assumed secure. For instance, on Internet of Medical Things (IoMT) systems, a high volume of patient data is exchanged, raising serious security concerns [1]. Consequently, many standards have been established to address these issues, such as implementing a secure socket layer and transport layer security, to prevent the leakage of confidential information [2]. Cybercrime is defined as any illegal action committed against computers or traditional crimes targeting individuals by using the internet [3]. Cybercrime is a serious threat to digital applications that hold personal information, such as Zoom [4] and the UK National Health Services (NHS) vaccination website [5]. The increasing role of IoT devices in digitized applications renders this threat even more important. In this work, we examine the ways in which IoT devices increase the risk of malware attacks and review pertinent detection and prevention methods.

IoT applications are the weak links in the information technology (IT) network, making them a major threat to the system's security [6]. Over 600 organizations were affected by the WannaCry malware attack in 2017, including health, educational, financial, and governance institutes, thus creating a global risk factor [7,8]. The NHS in the UK was one of the targeted organizations, where affected hospitals' yellow and medical staff were locked out of their digital system across England and Scotland. This incident caused missed appointments, deaths, and fiscal costs [8]. Malware attacks are one of the major security threats in the IoT

and malware detection, specifically detecting unknown malware files, is one of the ongoing challenges. IoT devices usually have a small amount of memory and processing capacity, which makes them lightweight. This characteristic limits the complexity of possible security solutions. There are three overarching challenges in the detection of malware attacks in the IoT. Firstly, the low computational power of most IoT devices limits the complexity of the security algorithm. Secondly, the rise of unseen malware attacks targeting IoT systems necessitates a fast-adapting detection mechanism (which requires complex algorithms). Thirdly, the rapid spread of IoT devices and the resulting increased security risk requires a highly robust protection mechanism.

Current malware detection solutions that work effectively on traditional networks are either too complex to implement on IoT networks or lack the adaptability and robustness to enable secure operations. Artificial Immune System (AIS) methods are inspired by the human immune system's methodology for fighting attacks. They are proven to be adaptive, distributed, robust, and not computationally expensive, which makes them suitable to secure the IoT. For this reason, we dedicate this review paper to investigating and analyzing the AIS methods in detecting malware files in the IoT.

Contribution and Organization of the Paper

This is the first paper to survey IoT security challenges and state-of-the-art malware detection methods with a focus on AIS. We present a critical analysis of the literature and offer a quantitative comparison, presenting new findings for the state-of-the-art solutions. Although the authors in [9] present an AIS survey for IoT security, they do not offer a quantitative performance analysis. The insights that we can draw from this comparison are not available in any other published paper, to the best of our knowledge. Furthermore, we present promising future research directions and implementation suggestions for employing AIS methods to secure the IoT.

The paper is structured into five main parts, as shown in Figure 1. In Section 2, we present an overview of IoT systems' characteristics and their impact on security, followed by malware attack analysis and detection techniques. Section 3 presents an introduction to the immune system techniques and AIS methods inspired by adaptive immunology. In Section 4, we offer a literature review and a critical analysis of AIS implementations in securing the IoT. Section 5 presents a performance analysis of the AIS for malware detection in the IoT state-of-the-art method. In Section 6, we present trends, promises, and implementation suggestions for future work.

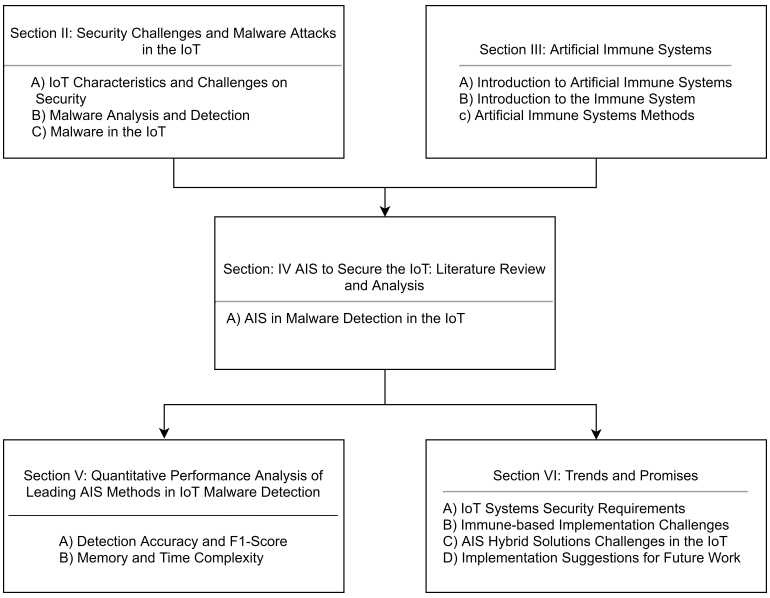

Figure 1. The structure of this review paper.

2. Security Challenges and Malware Attacks in the IoT

Malware is a major security threat to the IoT, and detecting unknown malware is one of the key challenges for two reasons. First, the limitations of IoT devices, such as their low power retention capability and low computational processing capability, represent a significant challenge when aiming to apply security solutions. Second, introducing new ways to connect networks, such as cloud services, opens the door to many security attacks, such as malware attacks. Furthermore, connecting new devices that were not part of traditional networks via these new connection methods, such as smart sensors, makes applying security measures more complex. For these reasons, traditional malware detection mechanisms are not suitable for the IoT environment.

In this section, we first present a brief background of IoT security challenges and limitations. This is followed by a study of existing methods for analyzing and detecting malware in general with a discussion of how they apply to IoT systems. Next, we examine IoT-related malware attacks, which have significantly increased in recent years and require immediate attention.

2.1. IoT Characteristics and Challenges on Security

The IoT is a system of interconnected machines with unique identifier numbers. The devices can communicate and share data within a network without human interaction. The IoT system consists of devices (often referred to as IoT devices) with unique identifiers that integrate seamlessly into the information network by using intelligent interfaces [10]. IoT systems often include connected, lightweight IoT devices and are employed in various applications, such as healthcare, environmental, smart cities, commercial, and industrial applications [11]. IoT devices employed in healthcare are referred to as IoMT devices. These include wearable monitoring medical devices, implantable medical treatment devices, and in-hospital connected medical devices and play a critical role in remote health monitoring and intervention [12]. As such, securing IoMT devices and systems is crucial and demands rigorous malware detection mechanisms. In environmental and agricultural applications, IoT devices such as temperature and humidity sensors are often battery operated and deployed in remote locations [13], thus requiring a malware detection mechanism that is computationally and energy efficient to extend the battery life. Smart cities leverage IoT systems using various types of devices, such as security cameras that capture sensitive data [14], and thus require strict security measures to prevent unlawful access. Industrial IoT (IIoT) applications refer to IoT systems in manufacturing and supply chains where humans work in the vicinity of machines operated by IoT devices [15]. In these cases, securing the IoT system against malware is pivotal for workers' safety and key to sustaining efficient IIoT operation. These IoT devices may be physical entities but also virtual things that interact, thus forming the IoT system with essential features as presented below (see Figure 2).

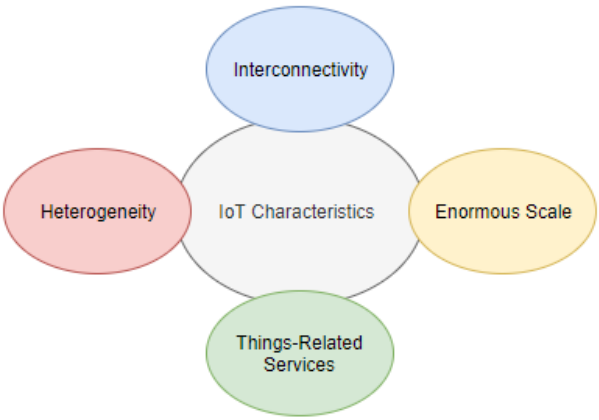

Figure 2. IoT characteristics.

- Interconnectivity refers to the connection of the device to the cloud and/or other devices. Connectivity is needed to enable the control of the device remotely, but mostly to access the data collected by the IoT device's sensors. For example, an IoMT device for heart disease prediction is remotely controlled to monitor the patient's heart rate [16]. The health parameters are collected in real-time and transmitted to a data center in the cloud. Therefore, securing this connection is vital to protect critical information.
- The IoT devices are heterogeneous as they may be built on different platforms and have different specifications. The hardware, such as a simple sensor to monitor the heart rate in [16], and virtual things, such as a data center built on the cloud, could be supplied by different vendors. These integrated IoT devices could use different security measures, leading to a lack of standardization in the network. Each connected device could use different security protocols, with their security bugs and limitations, exposing the system to different kinds of hacking.
- In the IoT environment, physical and virtual devices are capable of exchanging services within the constraints of the devices. Since the communication between different IoT devices is not controlled by a central processor/human, this could form a serious threat. If a malicious device is disguised as an accepted IoT device, it could start to disturb other devices by installing malicious files.
- The number of IoT devices is increasing exponentially and is generating an unprecedented amount of data. The expected number of IoT devices by 2025 is between 25 billion and 50 billion [17]. The scale is simply enormous, and data privacy and integrity are critical challenges in massive-scale networks. For instance, IoMT-based COVID-19 applications are creating massive amounts of real-time data that are stored in the cloud. However, as the amount of generated data continues to increase, the network pressure increases, which might lead to occurrences of erroneous interpretations [18].

The IoT involves smart devices and sensors, some of which use non-chargeable batteries. This makes battery life one of the predominant challenges in IoT security. Running security rules will drain the battery resources. Applying minimum security requirement measures is not recommended and is risky when devices have access to (or collect) sensitive data. Increasing battery size and capacity is not always possible, because these devices are designed to be lightweight and low-cost. In addition to device limitation and object identification, device authentication and authorization are examples of the IoT network-layer security challenges. Issuing certificates to each object in the IoT is extremely challenging due to the number of connected objects and lack of a global root certificate authority. The Domain Name System (DNS), which is used to identify objects and their attributes, is another IoT network-layer security challenge. Data integrity is problematic here due to the possibility of being hacked by a man in the middle or a DNS cash poisoning attack. This attack is the act of placing false information to redirect Internet traffic to malicious websites.

The threat of malware attacks arises in IoT due to these security challenges. Antivirus software is the main line of defense to detect known malware in real-time. However, the traditional security solutions have not been efficient and do not provide decentralized and strong security solutions in the IoT [9]. Due to the IoT device limitation and computing power, shifting similar solutions from traditional platforms to IoT might not be affordable [19]. Battery size and expected durability are challenges that make the implementation of security measures more limited, as a device has to be energy efficient as well as secure. Moreover, in IoT systems, network resources are integrated into devices that were never previously anticipated to be part of computer networks [20]. Integrating IoT devices into traditional networks introduces a new paradigm of security. The integrated systems inherit the traditional network security issues besides those targeting IoT devices [9]. Consequently, using traditional security measures is not enough to give IoT systems malware detection capabilities.

2.2. Malware Analysis and Detection

Malware is defined as malicious software that is executed within the system without the user's permission. Black hats, hackers, and crackers are all names for malware writers and developers. Writers have different intentions when creating this malicious executable software; e.g., internal threats, governance purposes, and spying on competitors. "Traditional" malware was often written using simple techniques and was designed with predictable intentions [21]. "Next-generation" malware, on the other hand, is designed with multiple malicious intents and leverages advances in technology for a more sophisticated design. The marriage of fast-spreading IoT systems and the inherent vulnerability and increased sophistication of malware attacks renders malware analysis and detection more critical but also more challenging.

2.2.1. Malware Analysis

Malware analysis techniques are essential to developing effective malware detection methods. These techniques involve the analysis of the process and functionality of the malware to build a suitable defense method. Three main malware analysis techniques achieve the same goal of determining how the malware works and how the attack will affect the network (see Figure 3).

- Static analysis, also called code analysis: In this technique, the infected file is inspected and analyzed without executing it. Low-level information is extracted such as the control flow graph (CFG), data flow graph, and system calls. Static analysis is fast at analyzing data and safe to use; also, it has a low level of false-positives, which means a higher detection rate. Moreover, the static analysis tracks all possible paths, which gives it a global view; however, it fails in detecting unknown malware using code obfuscation.
- Dynamic analysis, also called behavioral analysis: In dynamic analysis, the infected file is inspected during execution, which is usually conducted on an invisible virtual machine, so the malware file does not change its behaviors. Dynamic analysis is time-consuming and vulnerable, and it can only detect a few paths based on triggered files. Furthermore, it is neither safe nor fast, and it suffers from a high level of false positives. However, dynamic analysis is known for its good performance in detecting new and unknown malware.
- Hybrid analysis: this technique was designed to overcome the challenges and limitations of the previous two techniques. First, it analyzes the signature descriptions of any malware code and then combines that with other dynamic parameters to improve the analysis of malware.

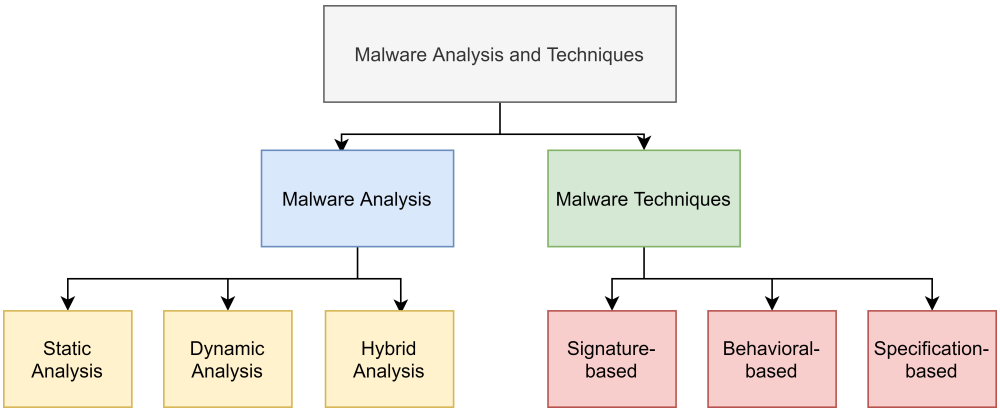

Figure 3. Malware analysis and detection.

The connection in IoT networks is currently enabled via cloud services. Static, dynamic, and hybrid malware analyses are mostly applied in the cloud to protect IoT devices.

2.2.2. Malware Detection Techniques

Based on the analysis results described in the previous section, we present detection techniques that are designed to detect malware attacks effectively. Three main methods are used in malware detection: the signature-based detection technique, behavior-based detection technique, and specification-based detection technique (see Figure 3).

- In the signature-based technique, files are analyzed and compared to an existing list, and if they are listed in the list, they are classified as malware. This method is not effective for recognizing all malware that enters the network because some malware is encrypted, and thus extracting the signature takes time and a large amount of processing energy. Furthermore, it is not effective for new or unknown malware.
- The behavioral-based method monitors the program's behavior rather than reading its signature. This technique follows three steps: the first step collects information about the program, the second step interprets the data through conversion to intermediate representations, and the last step matches the intermediate representation with known behavior signatures. There are two approaches to this technique, the first of which is simulating the behavior of legitimate programs and comparing any new program to that model. This approach works for the detection of most malware, even new kinds. However, it is expensive to implement because of the different behaviors of each program in the network; for example, a video reader will use different services than a mail or a web client. The second approach is simulating the behavior of known malware and comparing it to new programs, which means new (unknown) malware cannot be identified.
- The specification-based method was introduced to overcome the disadvantages and limitations of the first two techniques. This technique uses different features for malware detection, including the following:
  - (a) API calls: Hofmeyr et al. were among the first to propose using application interface and system call sequences for malware detection [22].
  - (b) OpCode: Executable files are made of series of assembly codes, and in this method, researchers use this operational code to detect malware [23].
  - (c) N-Grams: this method uses executable programs' binary codes for malware detection [24].
  - (d) CFG: This is a graph that illustrates the control flow of programs, and it has been used to analyze malware behavior [25].
  - (e) Hybrid feature: in this machine learning method, researchers combine different techniques for malware detection to get better results. For example, Eskandari et al. in [26] used CFG and API calls for metamorphic malware detection.
  - (f) Game theoretic-based anomaly detection algorithms: Zhu, Quanyan, and T. Başar presented different solutions to malware detection using behavioral analysis, such as the data exfiltration detection and prevention and consensus algorithm, with censored data for distributed detection [27].
  - (g) Prospect theoretic approaches: These approaches are based on measuring the trustworthiness of the aggregated data in the system. In [28], the authors present a hardware trojan detection game based on prospective theory approaches. Furthermore, in [29], the authors introduce a prospect theory-based framework to ensure risk awareness and protect network operations.

The main limitation of the specification-based method is the difficulty to specify the whole set of legitimate behaviors that a system should exhibit accurately [30].

2.3. Malware in the IoT

The malware detection techniques presented in the previous section have been followed to implement malware detection methods in the IoT; for instance, SVELTE, which is a signature and anomaly-based intrusion detection method, has been used to protect the IoT from routing attacks based on the IPv6 routing protocol [31]. On one hand, applying

a signature-based technique for malware detection in the IoT is not the best approach because it is not designed to detect unknown/newly developed malware files; on the other hand, designing a behavioral-based or specification-based method to secure the IoT is computationally expensive due to the long simulation process it requires.

Major AI solutions to securing the IoT fall under either behavior or specification-based techniques, which are complex to implement in IoT systems. For instance, the authors in [32] evaluate the recent advances in AI/ML techniques in securing the IoT. They use 80% of the dataset only to train the module, which is computationally expensive, and state that, despite the advances in AI techniques in the IoT, the security method is still vulnerable when implemented in a real IoT system. Furthermore, the authors in [33] published a survey about AI solutions enhancing IoT security by presenting the challenges and limitations of algorithms. Besides the weak probability and instability of AI algorithms, they are computationally complex, with high resource consumption. Therefore, in this work, we analyze the AIS solutions to secure the IoT that are less complex for implementation with high detection probabilities.

As businesses and consumers continue to connect devices to the Internet without proper security measures, IoT devices are increasingly leveraged by cybercriminals to dispense malware payloads [34]. In the first half of 2019, SonicWall observed a 55% increase in IoT attacks—a number that outpaces the first two quarters of the previous year. A security vendor has detected over 100 million attacks on IoT devices in the first half of 2019, which highlights the continued threat to unsecured IoT devices [35]. Kaspersky, the Russian Anti-Virus vendor, has claimed to detect 106 million attacks coming from 267,000 unique IP addresses in the first half of 2019 [35]. This number of attacks was almost nine times more than what was reported for the first quarter of 2018, when only 12 million were detected, originating from 69,000 IP addresses. According to the authors in [35], a major reason driving this surge is consumers' increased propensity to buy smart home solutions without due diligence in terms of security measures. Due to all the reasons listed above, malware attacks are major security threats in the IoT and thus require an IoT-specific security solution.

The best way to secure the IoT based on its characteristics and architecture is to implement a distributed, dynamic, adaptive, and self-monitoring method. This leads us to investigate the AIS solutions and how these can be applied to secure the IoT against malware attacks.

3. Artificial Immune Systems

In this section, we introduce the AIS methods, which are based on the human immune system. We first introduce the AIS concept, then we offer a brief introduction to the immune system and its defense mechanisms, by which AIS methods are inspired. Then, we present the main AIS methods that simulate similar principles.

3.1. Introduction to Artificial Immune Systems

Nature has ingenious ways to solve problems. The knowledge retrieved from the observation of nature has been a source of inspiration for computer scientists throughout the years when devising solutions to challenging problems; in particular, problems for which the traditional methods fail to provide a suitable solution or would result in a complex solution requiring high computational power. In cases where analytic expressions are not available, nature-inspired computing may be able to find sub-optimal solutions efficiently. Nature-inspired algorithms abstract the phenomena found in the wild and are subject to evolutionary steps or computing layers to converge to a solution. Examples include ant colony optimization, particle swarm optimization, artificial neural networks (ANNs), and AIS [36]. AIS is a field composed of different methods inspired by many theories of the biological immune system. The immune system is responsible for protecting the body from any intrusions and any possible danger, called an antigen. In this work, we

consider malware to be an unwanted foreign intrusion, and we examine the application of the defense mechanisms used by the human adaptive immune system in fighting antigens.

3.2. Introduction to the Immune System

The first line of defense in the body is the innate immune system, which is composed of outside layers to protect the body, such as the skin, and inside defense layers, such as the acid in the stomach. Furthermore, blood cells, such as Neutrophils, kill any encountered malicious agent (antigen) then die, and macrophages can kill up to 100 germs before they die. Macrophages can also kill infected body cells such as cancerous cells. If the innate system fails to eliminate the threat (antigen), the adaptive immune system is initiated, which has two Lymphocytes cell types. First, B-cells are engaged when an antigen enters the body and before the disease occurs. They provide antibodies to stick to the antigen and "mark it" as a sign for the macrophages to kill it. Also, memory B-cells keep information about the attack for future reference. Second are T-cells, which come when the infection occurs. T-cells are divided into helper T-cells and Cytotoxic T-cells. Helper T-cells are divided into two types: effector T-cells, which provide an alert and information about the antigen, and memory T-cells, which keep information about the antigen for future reference. The role of Cytotoxic T-cells is to kill the infected body cells that cannot be treated.

In the following paragraphs, we explain the collaborative methods followed by B-cells and T-cells in fighting antigens. Understanding these phenomena will help us to devise an AIS for fighting malware.

The main part of our adaptive immune system is B-cells, which generate the antibodies. In the human body, there are 100 million types of B-cells, and the reason for this is that each kind of B cell generates different antibodies to catch any possible attack because different antibodies handle different antigens. Consequently, when a certain type of antigen enters the body that requires a certain type of B-cells to handle it, the body starts generating more of that specific type of B-cells.

Regarding the generation of antibodies, they are made of thin and thick chains that consist of different kinds of deoxyribonucleic acid (DNA). To generate different types of antibodies that can mark any type of antigens, a mix and match of different DNA strains is created by the body. Consequently, after the mix and match, each B cell will end with its own kind of antibodies.

Clonal Selection consists of four steps. First, B-cells generate a test patch of their antibodies that go to the surface as "bait" and are called B-cell receptors. B-cells float in their zone, trying to find a matching antigen (which their specific antibodies can catch). Second, when a B-cell bonds with a cognate antigen, it doubles its size and divides into two B-cells, and these two B-cells will double in size and divide, making four B-cells in total. This process is called proliferation, and it takes up to 12 h for each B cell to grow and divide; the proliferation process takes about a week, at the end of which the body will have enough of that specific type of B-cells to mount a real defense against the same kind of antigen. Then, B-cells send all generated antibodies to the bloodstream, and after all the hard work, most B-cells die. The main job of antibodies is simply to mark the antigen, not to kill it. Finally, the antigen is marked with antibodies, so it is the phagocyte's role (such as macrophages) to eat it and kill it. The antibody forms a bridge between antigens and macrophages.

3.3. Artificial Immune Systems Methods

Based on the knowledge of how the adaptive immune system works, to defend the human body, researchers have started to develop different methods that imitate a similar process to protect computer networks. The use of AIS in security applications is mostly in the detection of security incidences, such as intrusions at the host or the network carried out by malicious actors, using low-level scripts, automated tools, or malware. We identify four artificial immune system methods: negative selection, positive selection, clonal, and artificial immune networks.

The negative selection method uses the supervised learning classification algorithm, which was inspired by the "process of self-tolerance of B-cells, and CLONALG, which is inspired by clonal selection theory and consists of mutation and selection processes" [37]. The method works in two phases: the detector generation phase, and the matching and detection phase. First, the method generates detectors that do not match the protected data; then, it keeps matching these detectors with the data. If a match occurs, it means a change has happened in the protected data and action must be taken. This method was first introduced in [38], and the main idea was to develop a method that has similar techniques to the human immune system, where the system is capable of distinguishing between self-cells (the body cells) and non-self-cells (antigens). In computer networks, we map the self-cells to authorized system files and non-self-cells to malicious files.

The detector generation and the matching and detection phases follow data representation and matching rules. Data representations are fundamental differences between many models of negative selection algorithms. It changes the matching rule process, detector generation, and the detection process. The main data representation method for this method is binary, assuming that all datasets are eventually implemented as binary bits. Other representations include numeric data, categorical data, boolean data, and textual data. These different representations could be grouped into two different categories: string representation and real-valued vector representation. The matching rule defines matching or recognition, which is the distance measured between the tested data and generated detectors. It is used in both the detector generation stage and detection stage. For all data representations, matching rule M can be formally defined as a distance measure between d and x within a threshold, where d is a detector and x is a data instance [39]. This matching rule introduces the concept of partial matching, where the detector and the data instance do not have to be exactly the same in every single bit to be matched. For example, if we have 11001100 as data, and we are applying a matching distance of 3, matched detectors could be (11001100, 11001111, 11001000, 00101100, etc.) where at least 5 bits match the original data of the detector.

This approach works by monitoring a wide network. Each copy of the detecting algorithm is unique, which means that if a copy at one site is found, the other sites still have their different copies. The detection is probabilistic, which means that there are different sets of detectors to protect each entity. In addition, the method should detect any foreign activities rather than checking for a certain pattern (for example, signature-based malware detection methods).

The positive selection method (inspired by negative selection) is inspired by the process of T-cell selection, where only T-cells that can recognize self-molecules (body cells) are used in the immune system. Unlike the negative selection method, this positive selection will generate detectors that recognize and match the self-protected data. Then, during the detection stage, if there is a detector that does not match the protected data, it means that some changes have occurred to the protected data. The Positive Selection Classification algorithm (PSCA) is a general classification algorithm that classifies unknown data using classifiers that can recognize self-class (system files) data. The authors in [37] applied PCSA to malware detection with the following steps: a learning stage, where the method learns how to classify data into two different classes (self and non-self), and stimulation and mutate stages. Finally, the radius is a threshold used for classification, as opposed to the usual classification approach where the minimal distance between several centers is used.

The clonal selection theory was proposed in [40], and states that B-cells undergo cloning, variation, and selection to mature affinity. The CLONALG method was proposed by Castro and Zuben, and it is inspired by the clonal selection theory; the CLONALG method was initially designed for optimization and pattern recognition issues [41]. According to the authors in [36] CLOALG requires the definition of five main factors. The size of the receptor population, selection strategy, number of receptors, the affinity function that returns real-valued measure, and the function to assign the rate of mutation and the

number of clones according to the affinity. A supervised data mining technique is used to simplify the cloning method. When an antigen enters the body, B-cells start cloning specific antibodies for that type of antigen, but if it is a new one, the immune system clones the most stimulated lymphocytes. Similarly, the CLONALG method generates a set of receptors R that can recognize a set of patterns P.

Artificial immune network (AIN) theory was proposed in [42]. AIN is an unsupervised learning algorithm that was inspired by B-cells' immunological memory, due to the existence of a mutually reinforcing network of themselves. This process means that B-cells interact with each other to spread information so that memory can be preserved, and active behavior is exhibited even when no immune response is taking place [36]. AIN mimics immune network theory and parts of clonal selection as well. The goal of the AIN system process is to set up a collection of repertoires for a given issue, where better-performing cells stifle low-similarity (comparable) cells in the system. This standard is accomplished through an intuitive procedure of presenting the population to outer data, to which it reacts with both a clonal selection reaction and inner meta-elements of intra-population reactions. Thus, it balances out the reactions of the population to the outside boosts.

As the human immune system can detect and react to antigens in our body, the AIS can determine and respond to malicious files that are different to the system files used in the training phase [43]. AIS can detect discrepancies in the system behavior and identify attacks without prior knowledge about them, which makes them ideal candidates for detecting unknown malware files. In the next section, we investigate the state-of-the-art AIS solutions in malware detection and in securing the IoT.

4. AIS to Secure the IoT: Literature Review and Analysis

AIS applications are artificial intelligence (AI) techniques inspired by the intelligence of the human body's immunology. Given its ability to detect unseen attacks and its low complexity, various AIS-based methods are proposed in the literature for IoT security. An immune-based architecture was presented in [44] to secure the IoT using edge technologies based on IoT system requirements. As rightly highlighted by the authors, the architecture meets IoT security requirements, such as adaptability and lightweight, and can secure IoT nodes from various security threats and attacks. However, the proposed method is to secure the IoT using edge technologies, which means it is limited to a certain IoT system architecture. Moreover, the availability of this method has not been considered during the evaluation process. In addition, to secure Internet protocol version 6 (Ipv6) in the IoT, a bio-inspired method was presented in [45]. An AIS-based method is implemented in the routing protocol for low-power and lossy networks to enhance the security level and performance with the given limited resources in the IoT. The main limitation of this approach is that it is time and energy consuming, which makes it difficult to secure IoT devices with limited resources. In the following section, we review AIS methods for IoT malware detection, including negative and positive selection algorithms and immune and artificial immune-based methods.

4.1. AIS in Malware Detection in the IoT

This section highlights the work conducted in malware detection using AIS in the IoT. The original negative selection algorithm uses Binary Encoding to represent self and non-self-datasets; later on, real-valued methods were proposed, and some researchers adopted different types of malware detection techniques such as variable-sized detectors [46], hypercube detectors [47], hyper-ellipsoid detectors [48], and multi-shaped detectors [49]. Deeper investigations have been conducted using a Hypersphere detector because it has simple mathematic calculations compared to the other types. These different data representation methods have not been applied to securing the IoT since they are not sufficiently lightweight to meet the IoT system requirements.

4.1.1. Negative and Positive Algorithms

One of the objectives of the main concept of negative selection is to produce enough detectors to cover the non-self-area, and most approaches generate these detectors randomly in different ways in an attempt to cover holes and overlaps and improve the detection rate. To overcome this challenge, many researchers have proposed combining two different AIS methods. The authors in [50] proposed using negative and positive detectors for malware detection. The main goal of the proposed method (the NPS) is to use fewer detectors while achieving high detection and recall rates, making it suitable to meet the constraints associated with IoT devices. One of the shortcomings of this method is that it has not been validated in an actual implementation. Furthermore, the authors in [51] proposed the MNSA algorithm, which is a combination of negative selection and positive selection detectors. The first set of detectors can recognize self-data, and the other set of detectors is used to detect non-self-data. The combination of the results of these two detector sets is supposed to improve the detection rate for unknown malware files in the system. To test the method's efficiency, randomly generated 12-bit long strings are used for both training and detecting stages of the algorithm. As result, it was claimed in [51] that the MNSA algorithm can detect up to 34% of all intrusions without any prior knowledge about the non-self, and it can confirm more than 90% of those detected files. The main limitation of this research is the fact that it was tested on random strings and not actual malware files. Furthermore, this method uses too many detectors in both negative and positive sets.

The authors in [37] proposed using the positive selection algorithm (PCSA) for malware detection. They define the PCSA as a general classification algorithm used for unknown data classification. Positive selection and clonal selection algorithm techniques were applied to secure the IoT. The algorithm has different stages, starting with the learning stage to produce classifiers: self and non-self. The main goal of this algorithm is to recognize self-data, and after the learning stage, the authors claim that all classifiers are available to classify unknown data. They also define two states after classification: overlap, where the unknown data is recognized by more than two kinds of classifiers; and hole, where the unknown data cannot be recognized by any classifier. To evaluate the proposed algorithm, the researchers in [37] compared their solution to another algorithm in [52]. In total, 3721 Windows malicious executables and 3458 benign Windows executables were collected for the experiment. There are four types of malicious files: backdoor, spyware, trojans, and worms. The main feature captured and used for malware detection here is I/O request packets (IRPs), for which they developed an MBMAS tool presented in [53] that can associate a process with its child process in run time. Researchers claimed a 99.30% accuracy result for the PSCA algorithm that they developed. The only limitation that this paper claimed is that IRP traces of programs vary from one host to another, and some IRPs repeat sometimes. This method has not been implemented in an IoT system, and we find this work not to be sufficiently robust to cope with the interconnective environment of the IoT.

4.1.2. Negative and Neural Networks

The authors in [54] proposed using a negative selection algorithm combined with neural networks (NSNN) for intrusion detection in the IoT. The research goal is to develop an algorithm that meets IoT requirements, is lightweight enough to apply to a wide range of IoT use cases, is capable of detecting previously unknown intrusion vectors, and provides an acceptable detection rate. The dataset used in this experiment is KDD NSL [55]. The authors use only the basic traffic features, which provide most of the needed information. The different types of intrusions are divided into 23 different sets (22 types of attacks and one normal). Then, the attack types are divided into three attack sets: denial of service (DOS), PROBE, and All Attack Types (AAT). They tested the algorithm against different percentages of normal and attacks of each type (10%, 20% . . . 90% attack and subsequently 10%, 20%. . . 90% of normal). Each one of the 27 sets iterated 100 times with different test data sets every time. The trained NSNN algorithm was tested against the dataset, and the

following coefficients were calculated: positive predictive value, negative predictive value, sensitivity, specificity, accuracy, Matthews correlation coefficient (MCC), and F1-Score (the harmonic mean of the precision and recall). This research succeeded in achieving an F1-Score of 0.77 in the DOS simulation, 0.72 in the PROBE simulation, and 0.73 in all AAT simulation results. The researchers in [54] claimed that their work is limited to the creation of the negative selection and neural network algorithm only. Currently, they make no claims about the best way to implement an online learning mechanism for it. Furthermore, they noted that the test set used in the experiment is dated, and the results should be used only for comparison purposes and not to demonstrate the actual performance of the algorithm. In addition to the presented shortcomings, we find the F1-Score of this algorithm to be unreliable in securing the IoT systems.

4.1.3. Immune and Artificial Immune Based Algorithms

The authors in [56] presented an AIS-based algorithm for malware detection (DeepDCA). DeepDCA uses a dendritic cell algorithm (DCA), which is a danger theory technique, and Self-Normalizing Neural Networks (SNN). The proposed approach focuses on the preprocessing phase, presenting the feature selection, the SNN signal categorization, signal processing, and anomaly metrics steps. The Bot-IoT dataset was used in the experiment, converting some of the categorical variables to easily apply the feature selection method. The method was evaluated using different file features, resulting in an F1-Score less than 50% when using imbalanced data for the best 10 file features in the dataset. When using balanced data for the 10 best file features in the dataset, the F1-Score increased to over 90%. Although this method achieves a high detection accuracy rate with low false negatives, it is neither sufficiently lightweight nor distributive to be implemented in IoT devices.

The artificial awareness architecture (AWA) was proposed by the authors in [57] as a model for artificial immune ecosystems. Their experiment shows that the proposed algorithm can detect intrusions in specific given IoT architectures; however, it does not detect outliers–anomalies.

Moreover, the researchers in [58] proposed a novel approach to securing the IoT based on immunology techniques. The proposed method adopts dynamic and circular defense processes against a security threat. It incorporates five links: security threat detection, danger computation, security response, security defense strategy formulation, and security defense. The first link is responsible for collecting and analyzing IoT network traffic, and the other links function based on the produced results. The method simulates AIS techniques for intrusion detection based on the following mechanisms: capturing the IoT traffic data and simulating the data to antigens in AIS; representing the detector simulation for the detection elements, such as the living time and the number of recognized antigens; thirdly, implementing a matching mechanism to determine if there is a match between a detector and an antigen. Also, the evolution process is represented by classifying the detectors into immature detectors, mature detectors, and memory detectors. In the experiment, cloning attacks, mutated cloning attacks, replay attacks, and mutated replay attacks were simulated. Even though this method can detect security threats and change detectors to adapt to the dynamic IoT environment, no real malware files were used in this experiment. In addition, this work was not implemented in a real IoT scenario.

Furthermore, the authors in [59] proposed an artificial immune-based method for intrusion detection in the IoT. The method involves many local intrusion detection submodels that share their learning attainments. The signature information in the IoT sense layer represents antigens in this method as binary strings. Detector sets are generated, and they include a number of antigens matched by the detector and the generation life of the detector. One of the main limitations of the proposed method is that it is not sufficiently lightweight to meet the IoT system requirements.

Finally, the authors in [60] proposed an AIS-based algorithm for intrusion detection in the IoT. It was claimed that the main signature information on the IoT datagram is extracted to be switched to a binary character string for experiment purposes. Different detector

stages are identified as immature, mature, and memory detectors. The authors stated that immature detectors meet the recognition diversity of intrusion detection, while mature detectors evolve to be immature detectors. Although this paper presents a new method of detecting unknown malware in the IoT environment, no simulation results were given. In addition, we find this method to be memory space and time consuming for IoT devices.

Table 1 shows a comparison of the AIS-implemented solutions for securing the IoT.

Table 1. Comparison of AIS applications for securing the IoT.

Method	Year	Experiment Results Included	Malware Files Used in the Experiment	Limitations and Shortcoming Presented	Method Covers Holes and Overlaps
NPS [50]	2021	✔	✔	✔	✘
MNSA [51]	2017	✔	✘	✔	✘
PCSA [37]	2011	✔	✔	✔	✔
NSNN [54]	2018	✔	✔	✘	✘
DeepDCA [56]	2020	✔	✔	✘	✘
AWA [57]	2017	✔	✘	✔	✘
Immune-base [58]	2013	✔	✘	✘	✘
AIS-based [59]	2012	✘	NA	✘	✘
Immune-based [60]	2011	✘	NA	✘	✘

5. Quantitative Performance Analysis of Leading AIS Methods in IoT Malware Detection

In this section, we highlight the main criteria to evaluate the performance of the most promising AIS methods in the literature for malware detection in the IoT [50,51,54]. The three most recent AIS solutions for securing the IoT are selected to present a quantitative performance analysis. These methods are selected because of their promising results (accuracy and false-negatives), which we were able to reproduce to enable the quantitative performance analysis. A false-negative denotes a malware that is falsely classified as benign. It follows that a better malware detection method is one that results in fewer false-negatives.

There are different datasets used to evaluate IoT security solutions. The most used datasets, as listed in [61], are the NSL-KDD, the Bot-IoT, the Botnet, and the Android malware datasets. In this performance analysis, we chose to use the NSL-KDD [55] for two reasons. First, unlike the other datasets, the NSL-KDD eliminates the redundant records in the previous dataset (KDD'99), resulting in a reduction of the number of borderline records compared to any other dataset [61]. This leads to more accurate results when evaluating an AIS-based security solution. Also, by eliminating the borderline records, we reduce the total number of records (see details in Table 2), unlike the Bot-IoT [62] which has 72,000,000 records. Using a larger number of records to evaluate an IoT security solution might overwhelm the system when running the solution in an actual IoT system setup. Second, the NSL-KDD dataset is used to evaluate the NPS and NSNN methods. Consequently, in order to enable a quantitative performance analysis, we reproduce the results of the MNSA using the same NSL-KSS dataset. The traffic data were captured by running 420 machines and 30 servers in 5 different departments. Although the NSL-KDD dataset is not IoT specific, it contains various malware attack types and offers different file features to test security solutions, which makes it a good fit for this experiment's purposes. In contrast to other machine learning approaches, AIS requires minimal data to create necessary detectors that are later used in the detection phase. In our case, 10% of randomly selected samples of the dataset are used in the detector generation phase, and the remaining 90% are used for testing. We compare the performance from two perspectives: in Section 5.1, we analyze the detection accuracy and F1-Score of each; in Section 5.2 we examine the complexity of each algorithm from both time and memory perspectives.

Table 2. NSL-KDD Dataset Used in the Experiment.

Total number of records used	1,074,992
Number of attack files	262,178
Number of benign files	812,814
List of attacks	Brute-force, Heartbleed attack, Botnet, Denial of service, Distributed Denial-of-Service, Web attacks, and infiltration of the network from inside
Number of traffic features	80
Some of the traffic features	Destination port, flow duration, average size of packet, number of forward packets per second, number of backward packets per second

5.1. Detection Accuracy and F1-Score

The NPS [50] uses both negative and positive detectors, and it overcomes two of the main challenges in securing the IoT applications. First, the method is lightweight, as it generates a smaller number of detectors compared to other AIS algorithms, such as the MNSA [51], with a higher detection rate accuracy, calculated using Equation (1). With 40 detectors in total (20 negative and 20 positive detectors), the NPS achieves up to a 91.92% detection rate, and a rate of up to 99.05% when using 60 detectors in total (30 negative and 30 positive detectors; see Figure 4). When reproducing the results of the MNSA, the detection rate accuracy increases to 80.51% when using 170 detectors in general (150 negative and 20 positive detectors). The mean detection accuracy rate for the NSNN [54] is 73.4%, which is lower than both NPS and MNSA algorithms. Second, it overcomes the false-negative detection challenge. As explained earlier, accuracy alone does not fully capture the detection performance as it does not highlight the false-negatives. In other words, an accuracy of detection of 75% may result from a 100% misclassification of malware (since 25% of the records are labeled as attacks—262.178/1,074,992, as shown in Table 2). To this end, we calculate the F1-Score (see Equation (4)), which is more representative of the performance when the data are not balanced.

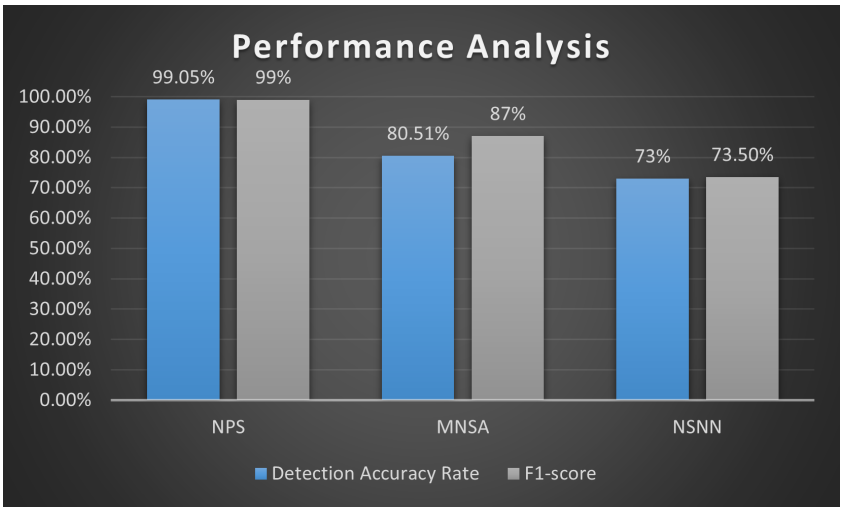

Figure 4. Accuracy and F1-Score results of NPS, MNSA, and NSNN using NSL-KDD dataset.

As shown in Figure 4, calculating the F1-Score for the NPS, we obtain a score of 96% when using 40 detectors in total. When using 60 detectors, the F1-Score for the NPS algorithm increases to 99%. The F1-Score for the MNSA increases to 87% when using 170 detectors, and the F1-Score for the NSNN is 73.5%. Overall, the NPS achieves almost a 14% improvement.

We present here a detailed explanation of the concepts used in the performance analysis:

- True positive (TP): malware is detected as a malicious application;
- True negative (TN): benign software is detected as non-malicious application;
- False positive (FP): benign software is detected as a malicious application;
- False negative (FN): malware is detected as non-malicious application.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

5.2. Memory and Time Complexity

IoT devices are lightweight with limited computing power; therefore, reducing memory usage and computing time when applying security methods is essential. We calculate the space complexity for the NPS, MNSA, and NSNN using Equations (5) and (6), where m is the alphabet size (m = 2 in binary representation), L is the string size, N_S is the amount of self-data, and N_R is the number of detectors. Table 3 shows the values of the three methods for each parameter. Using 16-bit strings with an equal number of detectors in both negative and positive sets in the NPS results in a 65% decrease in memory usage compared to generating 12-bit strings with larger detector sets in the MNSA. To calculate the space complexity for the NSNN, we assume that the string length is $\geq$7 since the R-Continuous Bit Matching (RCBM) is 7. RCBM is the number of matching bits between two strings: self and non-self. In this case, the NPS uses 90% less memory space than the NSNN.

When calculating the time complexity using Equation (5), the results show that the NSNN needs less computing time compared to the other two methods—MNSA and NPS. The following Figure 5 shows the result of the space and time complexity analysis.

$$Time = (m^L \times N_S \times N_R) \tag{5}$$

$$Space = (L \times N_S \times N_R) \tag{6}$$

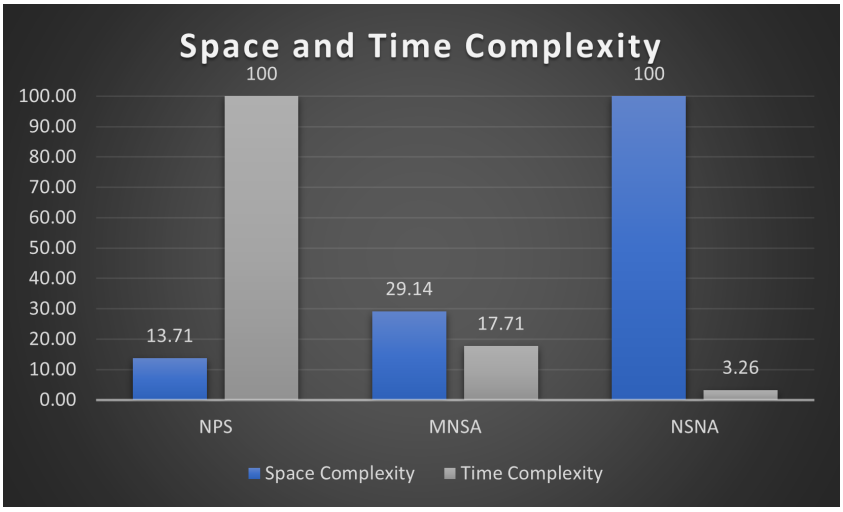

Figure 5. Memory and time complexity of NPS, MNSA, and NSNN.

Table 3. Space and time complexity calculations.

Method	M	L	N_S	N_R
NPS	2	16	1000	60
MNSA	2	12	1000	170
NSNN	2	7	1000	1000

6. Trends and Promises

6.1. IoT System Security Requirements

In the previous section, various implementations of AIS for securing the IoT were reviewed. Our study shows that there is a revived interest in addressing malware detection using the AIS method accompanying the spread of IoT systems. Researchers have proposed different ways of improving the detection rate for unknown malware in the IoT, and the NPS [50] seems to be a promising method based on the improved performance compared to the state-of-the-art using the same dataset. It remains to be proven that NPS can deliver the same performance on different datasets with different types of attacks, as well as using different file features. To this end, this line of research is a growing field that attempts to capture the characteristics of IoT systems and propose innovative AIS-based methods, respectively. Table 4 highlights five main properties to be taken into consideration when applying AIS applications to the IoT.

Table 4. IoT Systems' Properties.

Property	Definition
Robust	The capability of a system to cope with issues during execution and continue operating despite data conditions
Lightweight	The capability to operate and execute with minimal computational complexity
Fault tolerance	The capability to function given a defect within hardware or software in the system, and adapt to the changing environment to build up a trustworthy network
Adaptive	The capability to adapt and learn the system behavior over runtime
Distributed	The capability to run and communicate within a distributed environment

6.2. Immune-Based Implementations Challenges

Many AIS applications contain some of these properties, but implementing an AIS algorithm that meets all the requirements remains unsolved. For instance, designing an immune-based method results in implementing a robust and adaptive solution for securing the IoT; however, the method is neither lightweight nor fault-tolerant and not necessarily distributed [58–60].

6.3. AIS Hybrid Solution Challenges in the IoT

Implementing a method based on AIS techniques is difficult. For instance, clonal selection algorithms are adaptive but computationally expensive. Moreover, clonal selection suffers from high false-positives, and the degree of damage cannot be inferred instantly. On the other hand, the negative selection algorithm has high false-negatives and is not suitable for dense environments. Combining two or more AIS algorithms might be the solution to overcome some of these challenges, such as applying negative selection and neural network techniques in NSNN, which results in fault-tolerant, adaptive, and distributed solutions; however, it is not lightweight [54]. Furthermore, negative and positive selection algorithm techniques were combined in MNSA to improve the detection rate in the IoT [51]. Even though the goal of implementing this method was met, the solution does not meet all the IoT system's requirements, such as robustness. The same scenario applies to PCSA, which is not fault-tolerant as well [37]. Based on the characteristics of AIS methods and IoT system properties, we contemplated the reviewed AIS solutions in IoT and investigated which properties are applied in each solution. Table 5 below shows the result of this analysis.

Table 5. IoT system properties adopted in AIS solutions.

Method/Properties	Robust	Lightweight	Fault Tolerant	Adaptive	Distributed
NPS: negativeselection + positiveselection [50]	✔	✔	✔	✔	✔
MNSA: negativeselection + positiveselection [51]	✘	✘	✘	✔	✔
PCSA: positiveselection [37]	✘	✔	✘	✔	✔
NSNN: negativeselection + neuralnetwork [54]	✘	✘	✔	✔	✔
AWA: artificialimmune ecosystem [57]	✔	✘	✘	✔	✔
Immune system based method [58]	✔	✘	✘	✔	✘
Artificial Immune based method [59]	✔	✘	✘	✔	✔
Immune System based method [60]	✔	✘	✘	✔	✔

6.4. Future Research Directions

Based on the insights drawn in Section 4 and the comparative results in Section 5, we see three promising directions for future research. First, a promising research direction would be to investigate the implementation options based on the limitations of the IoT devices and the IoT system architecture overall. In many IoT system scenarios, one or multiple gateways are used as the main connection point between IoT devices and the cloud. Therefore, the gateway could be considered as a key security layer in the IoT architecture. As the gateway has more computational power and would support the implementation of security solutions, we suggest installing a hybrid AIS solution to secure the IoT on the gateway. A hybrid AIS solution combines multiple AIS techniques for malware detection to achieve a better detection accuracy rate. However, the IoT gateway is the main connection point for the IoT devices, so a downside to implementing a security method on the IoT gateway is that it could be a single point of failure. This obstacle could be overcome by having a backup security solution.

Second, conducting a quantitative analysis by calculating the detection accuracy rate and the F1-Score to evaluate given security solutions is another promising research direction. Using only a particular dataset to validate the results might not be sufficient for certain system architecture. Therefore, we suggest using different datasets conducted using different network scenarios and employing different file features to evaluate malware detection methods in the IoT.

The third promising research direction is as follows: to evaluate a security solution's ability to detect unknown malware files, the solution should be implemented in an actual IoT network. Creating different IoT system scenarios with different setups and processing power is key to evaluating a security solution in real time.

7. Conclusions

IoT systems are interconnected and heterogeneous devices with limited computational capacity. The number of IoT applications and their integration into traditional networks is increasing rapidly. This has led to new and fast-spreading security threats, not least malware attacks, that traditional security solutions fail to address adequately. Traditional IoT malware detection techniques employ signature-based and behavioral-based methods. We have demonstrated that these are either unsuitable for detecting unknown malware files or are not cost-efficient for IoT applications. AIS represents a research direction inspired by the human body's adaptive immune system for the detection of new threats. AIS methods are generally attractive for malware detection owing to their ability to detect unknown attacks and intelligently keep records of any attack for future use. In addition, they are a prime contender in the design of IoT malware detection because the offered features are the best match with IoT system characteristics. The features of AIS methods, such as their adaptivity, distributed implementation, lightweight computation, and robustness, are

compatible with the IoT devices' specific requirements. To this end, this article surveys recent research in the field of AIS for malware detection. We provide a critical analysis of existing works, draw key insights, and identify promising future research directions in which novel AIS techniques can be developed to address imminent and increasing IoT security challenges.

Author Contributions: Conceptualization, H.A. and G.G.; methodology, H.A.; software, H.A.; validation, H.A., G.G., M.J. and J.K.; formal analysis, H.A., G.G., M.J. and J.K.; investigation, H.A.; writing—original draft preparation, H.A.; writing—review and editing, H.A., G.G., M.J. and J.K.; visualization, H.A.; supervision, G.G., M.J. and J.K. All authors have read and agreed to the published version of the manuscript.

Funding: Hadeel Alrubayyi is a recipient of a Kingdom of Saudi Arabia Ministry of Education Scholarship. Furthermore, the Kingdom of Saudi Arabia Ministry of Education supports the studies of Hadeel Alrubayyi.

Data Availability Statement: We use the NSL-KDD dataset to run the performance analysis in this research. The dataset is available at https://www.unb.ca/cic/datasets/nsl.html (accessed on 1 October 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chamola, V.; Hassija, V.; Gupta, V.; Guizani, M. A Comprehensive Review of the COVID-19 Pandemic and the Role of IoT, Drones, AI, Blockchain, and 5G in Managing its Impact. *IEEE Access* **2020**, *8*, 90225–90265. [CrossRef]
2. Hatzivasilis, G.; Soultatos, O.; Ioannidis, S.; Verikoukis, C.; Demetriou, G.; Tsatsoulis, C. Review of Security and Privacy for the Internet of Medical Things (IoMT). In Proceedings of the 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), Santorini Island, Greece, 29–31 May 2019; pp. 457–464. [CrossRef]
3. Donalds, C.; Osei-Bryson, K.M. Toward a cybercrime classification ontology: A knowledge-based approach. *Comput. Hum. Behav.* **2019**, *92*, 403–418. [CrossRef]
4. The Biggest Data Breaches in the First Half of 2020. 2020. Available online: https://www.keepnetlabs.com/the-biggest-data-breaches-in-the-first-half-of-2020/ (accessed on 1 October 2021).
5. Irwin, L. List of Data Breaches and Cyber Attacks in May 2021. IT Governance UK Blog. Available online: https://www.itgovernance.co.uk/blog/list-of-data-breaches-and-cyber-attacks-in-may-2021-116-million-records-breached (accessed on 1 October 2021)
6. Outdated Software Leaves NHS 'Vulnerable to Cyber Attack'. Available online: https://www.digitalhealth.net/2019/04/outdated-software-leaves-nhs-vulnerable-to-cyber-attack-new-research-says/ (accessed on 1 October 2021).
7. Saleem, M. Brexit Impact on Cyber Security of United Kingdom. In Proceedings of the 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Dublin, Ireland, 14–16 June 2019; pp. 1–6. [CrossRef]
8. Ghafur, S.; Kristensen, S.; Honeyford, K.; Martin, G.; Darzi, A.; Aylin, P. A retrospective impact analysis of the WannaCry cyberattack on the NHS. *NPJ Digit. Med.* **2019**, *2*, 1–7. [CrossRef] [PubMed]
9. Aldhaheri, S.; Alghazzawi, D.; Cheng, L.; Barnawi, A.; Alzahrani, B. Artificial Immune Systems approaches to secure the internet of things: A systematic review of the literature and recommendations for future research. *J. Netw. Comput. Appl.* **2020**, *157*, 102537. [CrossRef]
10. Othman, M.; El-Mousa, A. Internet of Things Cloud Computing Internet of Things as a Service Approach. In Proceedings of the 2020 11th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 7–9 April 2020; pp. 318–323. [CrossRef]
11. Asghari, P.; Rahmani, A.M.; Javadi, H.H.S. Internet of Things applications: A systematic review. *Comput. Netw.* **2019**, *148*, 241–261. [CrossRef]
12. Koutras, D.; Stergiopoulos, G.; Dasaklis, T.; Kotzanikolaou, P.; Glynos, D.; Douligeris, C. Security in IoMT Communications: A Survey. *Sensors* **2020**, *20*, 4828. [CrossRef] [PubMed]
13. Marathe, S.; Nambi, A.; Swaminathan, M.; Sutaria, R. CurrentSense: A novel approach for fault and drift detection in environmental IoT sensors. In Proceedings of the International Conference on Internet-of-Things Design and Implementation, Charlottesvle, VA, USA, 18–21 May 2021; pp. 93–105.
14. Lv, Z.; Qiao, L.; Kumar Singh, A.; Wang, Q. AI-Empowered IoT Security for Smart Cities. *ACM Trans. Internet Technol.* **2021**, *21*. [CrossRef]
15. Xenofontos, C.; Zografopoulos, I.; Konstantinou, C.; Jolfaei, A.; Khan, M.K.; Choo, K.K.R. Consumer, Commercial and Industrial IoT (In)Security: Attack Taxonomy and Case Studies. *IEEE Internet Things J.* **2021**. [CrossRef]
16. Khan, M.; Algarni, F. A Healthcare Monitoring System for the Diagnosis of Heart Disease in the IoMT Cloud Environment Using MSSO-ANFIS. *IEEE Access* **2020**, *8*, 122259–122269. [CrossRef]

17. Zhang, J.; Li, G.; Marshall, A.; Hu, A.; Hanzo, L. A New Frontier for IoT Security Emerging From Three Decades of Key Generation Relying on Wireless Channels. *IEEE Access* **2020**, *8*, 138406–138446. [CrossRef]
18. Lin, H.; Garg, S.; Hu, J.; Wang, X.; Piran, M.J.; Hossain, M.S. Privacy-enhanced Data Fusion for COVID-19 Applications in Intelligent Internet of Medical Things. *IEEE Internet Things J.* **2020**. [CrossRef]
19. Jeon, J.; Park, J.; Jeong, Y. Dynamic Analysis for IoT Malware Detection With Convolution Neural Network Model. *IEEE Access* **2020**, *8*, 96899–96911. [CrossRef]
20. Greensmith, J. Securing the Internet of Things with Responsive Artificial Immune Systems. In Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation (GECCO '15); Association for Computing Machinery, Madrid, Spain, 18–21 May 2021; pp. 113–120. doi:10.1145/2739480.2754816 [CrossRef]
21. Aslan, Ö.; Samet, R. A comprehensive review on malware detection approaches. *IEEE Access* **2020**, *8*, 6249–6271. [CrossRef]
22. Hofmeyr, S.; Forrest, S.; Somayaji, A. Intrusion Detection Using Sequences of System Calls. *J. Comput. Secur.* **1998**, *6*, 151–180. [CrossRef]
23. Bilar, D. Opcodes as Predictor for Malware. *Int. J. Electron. Secur. Digit. Forensic* **2007**, *1*, 156–168. [CrossRef]
24. Schultz, M.; Eskin, E.; Zadok, F.; Stolfo, S. Data mining methods for detection of new malicious executables. In Proceedings of the 2001 IEEE Symposium on Security and Privacy. S & P 2001, Oakland, CA, USA, 14–16 May 2001; pp. 38–49. [CrossRef]
25. Jalote, P. *An Integrated Approach to Software Engineering*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
26. Eskandari, M.; Hashemi, S. Metamorphic malware detection using control flow graph mining. *Int. J. Comput. Sci. Netw. Secur* **2011**, *11*, 1–6.
27. Buttyán, L.; Baras, J.S. *Decision and Game Theory for Security*; Springer: Berlin/Heidelberg, Germany, 2010.
28. Saad, W.; Sanjab, A.; Wang, Y.; Kamhoua, C.A.; Kwiat, K.A. Hardware Trojan Detection Game: A Prospect-Theoretic Approach. *IEEE Trans. Veh. Technol.* **2017**, *66*, 7697–7710. [CrossRef]
29. Vamvakas, P.; Tsiropoulou, E.E.; Papavassiliou, S. Exploiting prospect theory and risk-awareness to protect UAV-assisted network operation. *EURASIP J. Wirel. Commun. Netw.* **2019**, *2019*, 1–20. [CrossRef]
30. Pandey, S.K.; Mehtre, B. A lifecycle based approach for malware analysis. In Proceedings of the 2014 Fourth International Conference on Communication Systems and Network Technologies, Bhopal, India, 7–9 April 2014; pp. 767–771.
31. Raza, S.; Wallgren, L.; Voigt, T. SVELTE: Real-time intrusion detection in the Internet of Things. *Ad hoc Netw.* **2013**, *11*, 2661–2674. [CrossRef]
32. Abusnaina, A.; Anwar, A.; Alshamrani, S.; Alabduljabbar, A.; Jang, R.; Nyang, D.; Mohaisen, D. Systemically Evaluating the Robustness of ML-based IoT Malware Detectors. In Proceedings of the 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S), Taipei, Taiwan, 21–24 June 2021; pp. 3–4.
33. Wu, H.; Han, H.; Wang, X.; Sun, S. Research on Artificial Intelligence Enhancing Internet of Things Security: A Survey. *IEEE Access* **2020**, *8*, 153826–153848. [CrossRef]
34. SonicWall 2019 Report: 55 Rise in IoT Malware Attacks. 2019. Available online: https://www.openaccessgovernment.org/iot-malware-attacks/69870/ (accessed on 1 October 2021).
35. Muncaster, P. Over 100 Million IoT Attacks Detected in 1H 2019. 2019. Available online: https://www.infosecurity-magazine.com/news/over-100-million-iot-attacks/ (accessed on 1 October 2021).
36. Fernandes, D.; Freire, M.; Fazendeiro, P.; Inácio, P. Applications of artificial immune systems to computer security: A survey. *J. Inf. Secur. Appl.* **2017**, *35*, 138–159. [CrossRef]
37. Fuyong, Z.; Deyu, Q. Run-time malware detection based on positive selection. *J. Comput. Virol.* **2011**, *7*, 267. [CrossRef]
38. Forrest, S.; Perelson, A.S.; Allen, L.; Cherukuri, R. Self-nonself discrimination in a computer. In Proceedings of the 1994 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, USA, 16–18 May 1994; pp. 202–212. [CrossRef]
39. Ji, Z.; Dasgupta, D. Revisiting negative selection algorithms. *Evol. Comput.* **2007**, *15*, 223–251. [CrossRef] [PubMed]
40. Burnet, M. *The Clonal Selection Theory of Acquired Immunity*; Vanderbilt University Press Nashville: Nashville, TN, USA, 1959; Volume 3.
41. De Castro, L.; Von Zuben, F. Learning and optimization using the clonal selection principle. *IEEE Trans. Evol. Comput.* **2002**, *6*, 239–251. [CrossRef]
42. Jerne, N. Towards a network theory of the immune system. *Ann. Immunol.* **1974**, *125*, 373–389.
43. Scaranti, G.; Carvalho, L.; Barbon, S.; Proença, M. Artificial Immune Systems and Fuzzy Logic to Detect Flooding Attacks in Software-Defined Networks. *IEEE Access* **2020**, *8*, 100172–100184. [CrossRef]
44. Roman, R.; Rios, R.; Onieva, J.; Lopez, J. Immune System for the Internet of Things Using Edge Technologies. *IEEE Internet Things J.* **2019**, *6*, 4774–4781. [CrossRef]
45. Saleem, K.; Chaudhry, J.; Orgun, M.; Al-Muhtadi, J. A bio-inspired secure IPv6 communication protocol for Internet of Things. In Proceedings of the 2017 Eleventh International Conference on Sensing Technology (ICST), Sydney, NSW, Australia, 4–6 December 2017; pp. 1–6. [CrossRef]
46. Ji, Z.; Dasgupta, D. Real-valued negative selection algorithm with variable-sized detectors. In *Genetic and Evolutionary Computation Conference*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 287–298.
47. Dasgupta, D.; Gonzalez, F. An immunity-based technique to characterize intrusions in computer networks. *IEEE Trans. Evol. Comput.* **2002**, *6*, 281–291. [CrossRef]

48. Shapiro, J.; Lamont, G.; Peterson, G. An evolutionary algorithm to generate hyper-ellipsoid detectors for negative selection. In Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation, Washington, DC, USA, 25–29 June 2005; pp. 337–344.
49. Balachandran, S.; Dasgupta, D.; Nino, F.; Garrett, D. A Framework for Evolving Multi-Shaped Detectors in Negative Selection. In Proceedings of the 2007 IEEE Symposium on Foundations of Computational Intelligence, Honolulu, HI, USA, 1–5 April 2007; pp. 401–408. [CrossRef]
50. Alrubbayi, H.; Goteng, G.; Jaber, M.; Kelly, J. A Novel Negative and Positive Selection Algorithm to Detect Unknown Malware in the IoT. In Proceedings of the IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Vancouver, BC, Canada, 10–13 May 2021.
51. Pamukov, M.; Poulkov, V. Multiple negative selection algorithm: Improving detection error rates in IoT intrusion detection systems. In Proceedings of the 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Bucharest, Romania, 21–23 September 2017; Volume 1, pp. 543–547. [CrossRef]
52. Igawa, K.; Ohashi, H. A negative selection algorithm for classification and reduction of the noise effect. *Appl. Soft Comput.* **2009**, *9*, 431–438. [CrossRef]
53. Zhang, F.; Qi, D.; Hu, J. MBMAS: A System for Malware Behavior Monitor and Analysis. In Proceedings of the 2009 International Symposium on Computer Network and Multimedia Technology, Wuhan, China, 18–20 January 2009; pp. 1–4. [CrossRef]
54. Pamukov, M.; Poulkov, V.; Shterev, V. Negative Selection and Neural Network Based Algorithm for Intrusion Detection in IoT. In Proceedings of the 2018 41st International Conference on Telecommunications and Signal Processing (TSP), Athens, Greece, 4–6 July 2018; pp. 1–5. [CrossRef]
55. NSL-KDD Dataset. Available online: https://www.unb.ca/cic/datasets/nsl.html (accessed on 1 October 2021).
56. Aldhaheri, S.; Alghazzawi, D.; Cheng, L.; Alzahrani, B.; Al-Barakati, A. DeepDCA: Novel Network-Based Detection of IoT Attacks Using Artificial Immune System. *Appl. Sci.* **2020**, *10*, 1909. [CrossRef]
57. Parrend, P.; David, P.; Guigou, F.; Pupka, C.; Collet, P. The AWA Artificial emergent aWareness Architecture model for Artificial Immune Ecosystems. In Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC), Donostia, Spain, 5–8 June 2017; pp. 403–410. [CrossRef]
58. Liu, C.; Zhang, Y.; Zhang, H. A Novel Approach to IoT Security Based on Immunology. In Proceedings of the 2013 Ninth International Conference on Computational Intelligence and Security, Emeishan, China, 14–15 December 2013; pp. 771–775. [CrossRef]
59. Chen, R.; Liu, C.M.; Chen, C. An artificial immune-based distributed intrusion detection model for the internet of things. In *Advanced Materials Research*; Trans Tech Publ.: Zurich, Switzerland, 2012; Volume 366, pp. 165–168.
60. Liu, C.; Yang, J.; Chen, R.; Zhang, Y.; Zeng, J. Research on immunity-based intrusion detection technology for the Internet of Things. In Proceedings of the 2011 Seventh International Conference on Natural Computation, Shanghai, China, 26–28 July 2011; Volume 1, pp. 212–216. [CrossRef]
61. Albulayhi, K.; Smadi, A.A.; Sheldon, F.T.; Abercrombie, R.K. IoT Intrusion Detection Taxonomy, Reference Architecture, and Analyses. *Sensors* **2021**, *21*, 6432. [CrossRef] [PubMed]
62. The bot-IOT Dataset. Available online: https://research.unsw.edu.au/projects/bot-iot-dataset (accessed on 1 October 2021).