2023

Machine-Learning-Based Vulnerability Detection and Classification in Internet of Things Device Security

Sarah Bin Hulayyil
Cardiff University

Shancang Li
Cardiff University

Li Da Xu
Old Dominion University, lxu@odu.edu

Original Publication Citation

Bin Hulayyil, S., Li, S., & Xu, L. D. (2023). Machine-learning-based vulnerability detection and classification in Internet of Things device security. *Electronics*, *12*(18), 1-24, Article 3927. https://doi.org/10.3390/electronics12183927

electronics

MDPI

Article

Machine-Learning-Based Vulnerability Detection and Classification in Internet of Things Device Security

Sarah Bin Hulayyil [1,2], Shancang Li [1,*] and Lida Xu [3]

[1] School of Computer Science and Informatics, Cardiff University, Cardiff CF10 3AT, UK ; binhulayyilsh1@cardiff.ac.uk
[2] College of Applied Studies and Community Service, King Saud University, Riyadh 11451, Saudi Arabia
[3] Department of Information Technology & Decision Sciences, Old Dominion University, Norfolk, VA 23529, USA
* Correspondence: lis117@cardiff.ac.uk

Abstract: Detecting cyber security vulnerabilities in the Internet of Things (IoT) devices before they are exploited is increasingly challenging and is one of the key technologies to protect IoT devices from cyber attacks. This work conducts a comprehensive survey to investigate the methods and tools used in vulnerability detection in IoT environments utilizing machine learning techniques on various datasets, i.e., IoT23. During this study, the common potential vulnerabilities of IoT architectures are analyzed on each layer and the machine learning workflow is described for detecting IoT vulnerabilities. A vulnerability detection and mitigation framework was proposed for machine learning-based vulnerability detection in IoT environments, and a review of recent research trends is presented.

Keywords: IoT security; vulnerability detection; cyber attacks; device security

Citation: Bin Hulayyil, S.; Li, S.; Xu, L. Machine-Learning-Based Vulnerability Detection and Classification in Internet of Things Device Security. *Electronics* **2023**, *12*, 3927. https://doi.org/10.3390/electronics12183927

Academic Editor: Hung-Yu Chien

Received: 30 July 2023
Revised: 6 September 2023
Accepted: 8 September 2023
Published: 18 September 2023

1. Introduction

The growth of technologies such as artificial intelligence (AI), smart sensors, cloud computing, the sixth-generation wireless (6G), and edge computing has significantly enhanced the capability of the Internet of Things (IoT), which is successfully transforming our daily lives through developments such as *smart homes, smart cities, Industry 4.0, healthcare, and more* [1,2]. They are now becoming broadly embraced because they offer cost-effectiveness and improve people's quality of life [3]. The IoT is also significantly impacting many industrial sectors, including *cyber security, manufacturing, shopping and retail, agriculture, transportation, the infrastructure industry, smart grids, the hospitality industry, etc.* Many enabling techniques have been developed in the IoT in parallel with their architectures, as *Strategy Analytics* released that the IoT devices will rise more than USD 38 billion by the end of 2025 and USD 50 billion by 2030 [4]. Many IoT ecosystems are being developed using sensors and microprocessors and then transferring sensitive data within IoT networks.

As IoT devices transmit data by wire or wireless, most of this data should be secured while transferring and storing. Aruba et al. conducted a study that indicated that 84% of all reported security issues involving data breaches took place on the IoT in 2019 and these attacks could deter the adoption of IoT techniques [5]. Securing IoT systems is crucial and it is necessary to ensure that confidentiality, integrity, availability, non-repudiation, authentication, and authorization are presented in IoT environments. Moreover, a weakness in any of these security properties could lead to many attacks. The architecture of the IoT systems comprises the perception layer, network layer, and application layer; all these layers must be secured to avoid any attacks occurring [6]. Securing the IoT systems in each layer is challenging because of the associated limitations, such as energy consumption, limited memory capacity, and low-performance processing. However, there are certain

techniques that have been proposed and used in specific IoT layers, such as blockchain and authentication methods [7,8].

Furthermore, there are many challenges and security issues involving IoT systems and most of these relate to authentication attacks such as *DoS*, *ransomware*, *replay attack*, *and spoofing attacks* [9]. The top 10 security vulnerabilities include *poor passwords*, *insecure services*, *insecure interfaces*, *lack of update mechanism*, *insecure/outdated components*, *inadequate privacy protection*, *insecure data transfer/storage*, *lack of device management*, *insecure default setting*, *and lack of physical protection* (https://www.cardinalpeak.com/blog/top-10-iot-security-vulnerabilities, accessd on 5 September 2023). Importantly, it is impossible to fully secure IoT devices and systems against any potential attacks and prevent attackers from compromising IoT systems. However, it is crucial to have a good understanding of the critical vulnerability and better design IoT systems without security holes.

In the past decade, many research efforts have been conducted to enhance IoT security standards and actions by both industry and academia. The significant development of security solutions for IoT systems is related to handling large volumes of data created by the IoT, including using emerging techniques, such as *AI*, *blockchain technology*, security strategies, protocols, standards, and actions, to secure these IoT assets from cyber attacks. Recently, many enabling techniques have been developed using AI and machine learning (ML) to detect, identify, monitor, and protect IoT devices, applications, users, and data against cyber attacks [1]. Figure 1 indicates the number of recent studies that have used ML in IoT vulnerability; this figure is based on SCOPUS, IEEE, and Science Direct databases, and it shows the increase of use of ML to manage the vulnerabilities in IoT environments. However, most of these studies use ML to detect different types of attacks, which means they implement ML and deep learning (DL) algorithms to detect the attacks after they have started. As a result, this paper investigates ML and DL used to detect attacks and then provides a brief overview of various algorithms that are used with different datasets, including a summary of their limitations, to encourage future research to enhance IoT security.

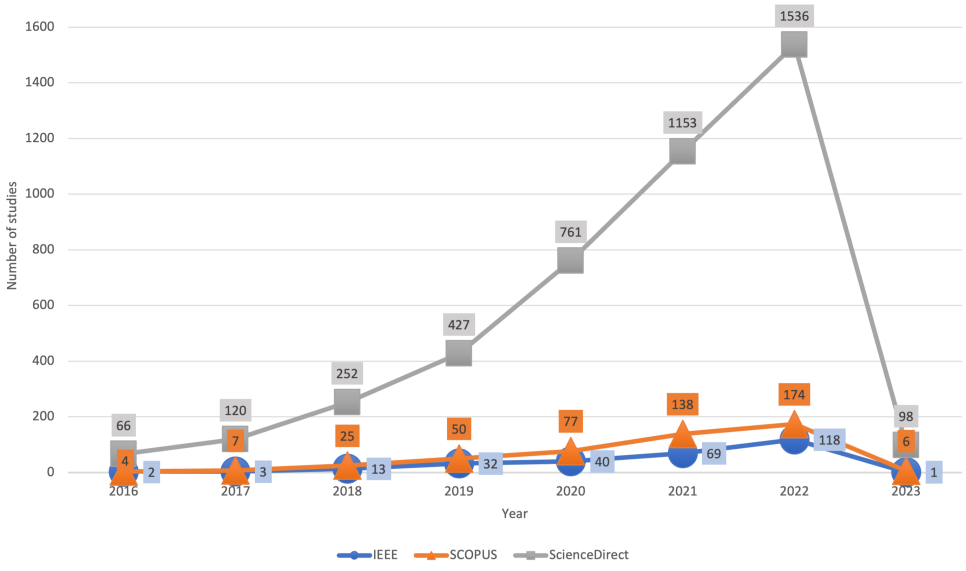

Figure 1. The studies in ML and DL for secure IoT.

ML is a subset of AI that makes the machine or device perform its tasks automatically. DL is a subset of machine learning that consists of three techniques, supervised, semi-supervised, and unsupervised. It involves many layers of artificial neural networks and each layer has multiple neurons. Neurons in each layer have activation functions that can be exploited to produce non-linear responses. The structures of brain neurons are reflected in this methodology [10,11]. It has been shown that ML algorithms are effective in detecting security attacks due to these technologies supporting security requirements

in IoT environments. Thus, focusing on using these techniques to detect IoT vulnerability could efficiently improve security and avoid zero-day attacks. Vulnerability detection and classification in the IoT ecosystem are dependent on defining characteristics that need to be meticulously examined. A comprehensive understanding of IoT architecture and advanced ML techniques is required, which is explained in this paper. Using ML algorithms, it is possible to detect complex vulnerability within diverse IoT devices, enabling potential threats to be avoided. Meanwhile, the classification characteristic involves a structured taxonomy based on severity and impacted devices, thereby facilitating targeted mitigation strategies, which will be discussed later.

This work aims to provide a comprehensive summary of IoT vulnerability detection and classification using emerging techniques, such as ML and DL. The main contributions can be summarized as follows:

- A comprehensive survey has been undertaken to focus on recent research studies using ML and DL to detect and classify vulnerabilities and attacks in IoT ecosystems.
- This work investigates the common potential vulnerabilities at each layer in the IoT architecture and summarises ML workflow to detect vulnerabilities in IoT devices.
- A framework is proposed to detect and mitigate the vulnerabilities in IoT environments, and recent research trends on machine-learning-based vulnerability detection are summarized.

2. Related Works

Numerous prior surveys have reviewed IoT vulnerability detection by focusing on particular areas such as firmware and IDS. The most related ones are summarized, which inspires forward for our contributions in this paper. Neshenko *et al.* addressed IoT vulnerabilities that are constantly evolving and explained a comprehensive approach in order to categorize state-of-the-art surveys in [12]. In addition, the works in [13,14] focused only on detecting IoT firmware vulnerability, whereas [15] described the common IoT communication protocols and how they implemented specific security mechanisms to make a comparison of the considered IoT technologies. In contrast, a paper described the IoT vulnerabilities and reviewed state-of-the-art articles but without seeking deeper into ML and DL techniques that are used to improve IoT security [16]. The latest published survey proposed a state of the art to using AI to enhance the IoT security by focusing on specific algorithms [17].

This paper added a crucial field of using ML and DL to enhance IoT security depending on each layer of the IoT architecture. The IoT systems, devices, users, and applications suffer from various vulnerabilities that an attacker can potentially exploit. Due to limited resources of IoT devices, the AI technology links high-security levels with low computational complexity. Recently, ML and DL technologies have been increasingly used to enhance the security of IoT environments, and a number of new solutions have developed to improve the security of IoT systems. Due to the IoT ecosystems consisting of a hardware layer, network layers, a middle-ware layer, such as intrusion detection systems (IDS), intrusion prevention systems (IPS), and firewall, and, finally, a software layer, each vulnerability in these layers could be exploited to undertake several attacks.

3. Vulnerabilities in IoT Environments

This work highlights the most recent studies aiming to use ML and DL to detect and classify vulnerabilities and attacks in the hardware layer, middle-ware, network and software layer. Figure 2 shows the potential vulnerabilities in each layer in the IoT architecture.

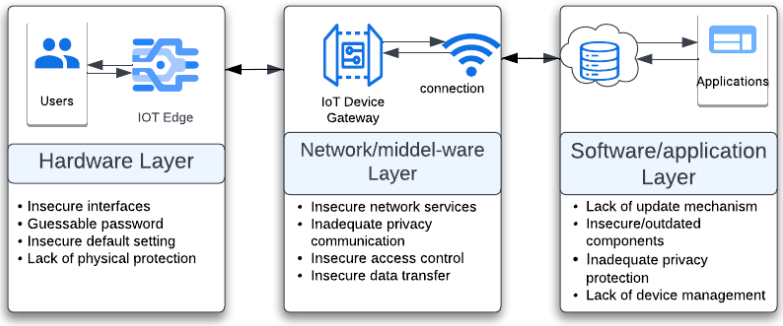

Figure 2. Potential IoT vulnerabilities on the IoT architecture.

Table 1 presents the top 10 IoT vulnerabilities and their impact identified by the Open Web Application Security Project (OWASP) [18].

Table 1. Top 10 IoT vulnerabilities and consequences.

Vulnerabilities	Impact (Consequences)
Weak password	This vulnerability usually was performed to carry out attacks against the authorization of the IoT devices which allows the attacker to gain permission to control and access its ecosystem [19–23].
Insecure network services	Hampers both the confidentiality, integrity, and availability of the system [24–26].
Insecure Interfaces	Alteration and injection could be implemented against authorization and authentication. The authentication process identifies the device, whereas authorization grants permissions and both of these must exist in IoT devices to perform their roles. The result of these types of attacks could be unauthorized access and access to the network boundaries that connect within the device [27–29].
Lack of update mechanism	Consists of unconfigured firmware, unencrypted delivery, no anti-rollback mechanisms, and no notifications of security updates, which leads to exploit the threat on the IoT software. For instance, the software updates can be replaced with malicious code by an attacker if there is an unsecured update mechanism [27,30,31].
Insecure/outdated components	Using insecure components such as operating system platforms or third-party hardware could compromise the device, such as using hardware, like a server, which needs to be secured to prevent an attacker from gaining access to it or to the network for launching an attack or using it to initiate a botnet attack [32–34].
Inadequate privacy protection	Most of the IoT ecosystems store personal information for their users and any failure to protect these data could lead to this information being stolen or compromised. In addition, the users must be aware of the privacy methods that are used in their devices, such as where and how their data will be stored using the regulation to protect their privacy [35,36].
Insecure data transfer/storage	It is necessary to protect the data in the whole IoT for both transferred data and saved data in order to protect the consumers' privacy. As the IoT ecosystem uses sensitive data in transit and storage or when processing, these data must be protected by using an efficient technique such as encryption to ensure the integrity and confidentiality of the ecosystem [20,35,37].

Table 1. *Cont.*

Vulnerabilities	Impact (Consequences)
Lack of device management	IoT device management issues occur when a company fails to adequately protect and secure connected devices. In an IoT environment, IoT device management is responsible for configuring, monitoring, and maintaining connected devices. Keeping the system updated and monitored, asset management, and updating the response capabilities protect the system against cyber attacks and data breaches [38–40].
Insecure default setting	By restricting operators from altering configurations, the system cannot be secured, such as a lack of file system permissions could be exploited by running the services as root. It is possible to interrupt secret keys that are utilized to launch connections within a restricted network during the on-boarding process for an IoT device, thereby allowing the attackers to initiate an attack from the deepest physical layer such as the motherboard [41–43].
Lack of physical protection	Poor physical hardening could make the IoT-connected network vulnerable. It may allow attackers to gain sensitive information that can be used in future remote attacks or to gain control of the device [44–48].

It can be seen from Table 1 that most IoT vulnerabilities are caused by the weak built-in security strategy. In our previous work, we have proposed a framework to test and evaluate the security in an IoT device using TP-Link Tapo P100 smart plug as a use case. In the framework, manual scanning was conducted to detect vulnerability that resulted in the most critical vulnerability, some of which also has different sub-vulnerability, e.g., Treck Transmission Control Protocol (TCP/IP) Stack (Ripple20). Actually, lots of work to detect vulnerabilities has been undertaken at the device level [49]. Researchers at Joint Special Operations Forces (JSOF) discovered a number of zero-day vulnerabilities, and a low-level TCP/IP library introduced by Treck was used to exploit these vulnerabilities. It consists of 19 vulnerabilities, 4 of which are classified as critical, in the high risk of conducting remote code injections, enabling external access to the network and allowing malicious code to be embedded in the devices which results in potentially dangerous effects on these systems,that are reported in public corpora of Common Vulnerabilities and Exposures (CVE), namely:

- *CVE-2020-11896* could be triggered by sending multiple malformed IPv4 packets to a device supporting IPv4 tunneling to cause a Denial of Service (DoS) attack.
- *CVE-2020-11897* can be prompted by sending multiple malformed IPv6 packets to a device that supports IPv6 to enable the execution of remote code.
- *CVE-2020-11901* allows remote code execution by answering a single DNS request made from the device, so it allows an attacker to gain control of the device via DNS cache poisoning.
- *CVE-2020-11898* could expose sensitive information when an unauthorized network attacker send a handling packet.

Many IoT device manufacturers, including *HP, Schneider Electric, Intel, Rockwell Automation, Caterpillar, and Baxter*, are affected by these vulnerabilities, as are companies operating in various fields such as medicine, transportation, industry, enterprise, and energy (oil/gas) sectors. As the low-level TCP/IP library spreads widely, tracking it becomes very difficult. JSOF discovered that this software has expanded worldwide through direct and indirect use [50]. Efforts to exploit this vulnerability to conduct attacks in each layer of the IoT architecture are summarized in the following subsections.

3.1. Vulnerabilities in the IoT Hardware Layer

IoT hardware includes smart sensors, gateways, switches, and some network infrastructure. The IoT hardware layer consists of all the physical devices and connectivity

protocols which are responsible for collecting the actual data and transforming them into the middle layer. It could consist of many sub-layers to increase IoT security, as proposed in a framework to protect IoT chips using multi-layer hardware [51].

Making IoT devices incapable of performing their main task is one of the core aims of attackers, so there are many attacks that target specific protocols such as TCP to switch the device off or move it away from the network to make it unavailable. For example, de-authentication attacks and DoS attacks which target the IoT device itself. Moreover, from the previous work on the TP-Link Tapo P100 smart plug, there were two attacks that targeted the Tapo device: de-authentication attack and TCP SYN flooding–DoS attack. Both of these attacks were performed successfully by exploiting the plug vulnerabilities.

- De-authentication attack on the smart plug. It points to making a DoS attack on the communication between the Wi-Fi access point and the targeted device and aims to disconnect the device from the network, even if the network password is unknown.
- TCP SYN flooding–DoS attack. This involves sending a lot of lawful SYN requests to the targeted IP address, and the victim's system will never receive the final ACK message; as is known, the TCP protocol uses a three-way handshake mechanism. Therefore, the number of open connections will increase, causing the targeted machine to break down. It could be conducted by sending the SYN packets from one specific IP or from random IPs.

3.2. Vulnerabilities and Attacks in the IoT Network Layer/Middle-Layer

The IoT network layer acts as a bridge between the hardware layer and the application layer (software layer). Due to the importance of routing and forwarding performed by the network layer, attacks on the network layer could disrupt these services by causing packets to be delayed or dropped [52].

This layer works with different connections in each system and is responsible for transferring data to distributed applications. Network attacks in IoT systems are the most common attacks because each IoT system uses different schemes with the connection such as Bluetooth, Wi-Fi and mobile networks. The attacks could be internal, or external, and the most common network attacks are Distributed DoS (DDoS) and Botnet. During the evaluation of the IoT device, it was found that the most critical vulnerability was Ripple20, which is a set of 19 vulnerabilities found on the Treck TCP/IP stack. These network vulnerabilities could cause many attacks, and protecting the network from these attacks is difficult because the solution for this vulnerability must be embedded within the system. Therefore, exploiting this vulnerability leads to successful Dynamic Host Configuration Protocol (DHCP) attacks, DHCP starvation, and Rogue DHCP server man-in-the-middle (MITM) attacks. DHCP starvation aims to make all the IPs in a pool reserved to prevent any device connecting to the network, and it involves a DHCP server on the router. Then, a fake server is created after all the IP addresses have been starved, making the attacker machine a DHCP server offering IP addresses for other devices that are going to connect within the network. At the last stage, the attacked device takes a new IP address from the Rogue DHCP server, which is the attacker's machine, and then the attacker can capture the traffic and control the attacked device.

3.3. Vulnerabilities and Attacks in the IoT Application Layer

The IoT application layer in the IoT ecosystem contains of all the applications, software, and programming libraries, such as applications, web platforms, services, and the operating system for the IoT ecosystem which plays an important core in the security of IoT environments. Each component of this software has its protocols that manage and transfer data through the IoT ecosystems [53]. In addition, it is responsible for transferring data from the end devices, such as sensors or passing commands and gateways which are the bridge between the end device, the storage and applications [54].

Due to the end-requirement users for privacy protection, it may take into account the central system protection for machine-to-machine communication depending on the

privacy level [55]. Regarding IoT web application attacks, there have been several exploitations in the past with Android systems which were released in 2008 [56]. As a result, data storage should be secured to avoid any threats that may cause attacks. Furthermore, any vulnerable component of the software layer or web applications could lead to many types of attacks, such as stealing data or credential information, and, therefore, securing such software would require different methods depending on the task of the IoT systems; dealing with video or voice will be different to dealing with text.

4. Vulnerability Analysis Using ML and DL

After exploring the potential vulnerabilities and attacks in each layer in the IoT architecture in the previous section, this section will explain the existing studies that use ML and DL to detect these threats.

Vulnerability in the IoT is the weak points in the ecosystem that make it easy for attacks to be conducted, such as poor saving of data, which could lead to sensitive data being stolen [57]. There are many studies that advocate securing the IoT by using ML and DL techniques. Figure 3 shows the recent ML and DL algorithms, classified depending on the type of ML algorithms, that are used to secure IoT environments by detecting attacks and threats as well as enhancing IDS security in IoT environments; also, it shows the methods with the best result's study that used them. This study presents the recent vulnerabilities and attack detection over the layered structure to help future researchers resolve the issues and make any further implementation to improve their security in a way that is easy to understand and conduct. In addition, layering makes it easier to distinguish between the duties allocated to each layer, making it simple to identify a task utilizing the layering structure. Moreover, it facilitates scalability because knowing the assets and the objects that need to be protected in each layer could help to implement security measures and increase the security in each layer. Detecting vulnerabilities, attacks, and threats by using ML and DL will be discussed in the next sections based on the IoT layers.

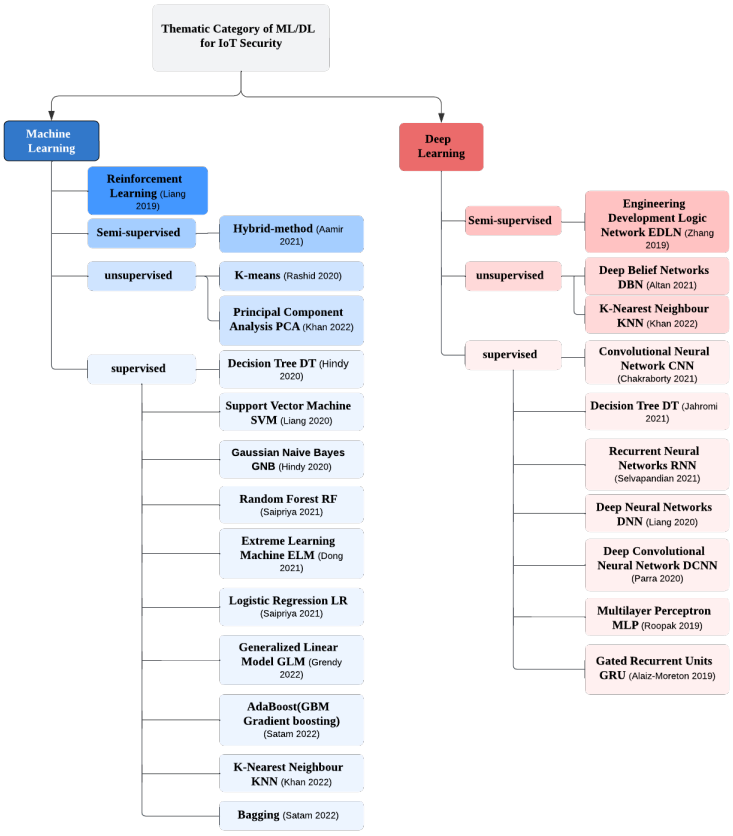

Figure 3. IoT security taxonomy using machine learning and deep learning [58–75].

4.1. Machine Learning Techniques in Detecting Attacks in the IoT Hardware Layer

Recently, there has been considerable research attention focusing on the use of ML and DL empowered intrusion detection and prevention analysis. The recent research works [76] focused on securing agricultural IoT using a federated-learning (FL)-based intrusion detection system. ML and DL techniques show great potential in automated vulnerability detection because both techniques are able to determine patterns and learn from them to identify similar vulnerabilities or threats and respond to changing behaviors [58]. The ML and DL techniques are able to enhance IoT security by providing proactive insights into the detection, classification, and prevention from threats and active attacks. Specifically, DL techniques are able to detect and categorize potential vulnerabilities from large amounts of data over IoT.

Moreover, DL-based methods are able to scan for vulnerabilities in the whole IoT system at an early stage. Numerous DL-based solutions have recently been developed by researchers [58,59,77]. They implemented their DL-based solutions by using three classifier DL algorithms: *deep neural networks (DNNs), convolutional neural network (CNNs), recurrent neural networks (RNNs)* . These methods were validated by two datasets, *CSE-CIC-IDS2018* dataset and *InSDN* dataset. The results indicated that the RNN achieved the highest accuracy on the *CSE-CIC-IDS2018*, whereas CNN had the highest accuracy on the second dataset. In addition, Jain et al. used an intrusion detection system based on a neural network to discover and restrict suspect devices in the IoT healthcare systems [77]. This study achieved an accuracy of 99.4% by implementing real data [77].

Other studies have used the same NSL-KDD dataset, which was collected in the wired network environment. One demonstrated a deep-learning-based IDS model to solve the restrictions of NN-based IDS, which involved implementing on multi-cloud IoT systems [59]. This model exists between the IoT gateway and the IoT cloud, which results in the IoT and the cloud networks being secured. In addition, it achieved 96.28% in terms of detection accuracy. In 2019, a study confirmed the efficiency of using ML for IDS, and it showed that it can improve the system's performance by using a multi-agent reinforcement algorithm [60]. They used the same dataset, and it achieved the highest accuracy of DNN on the transport layer of the IoT environment, which was 98%. This was followed in 2020 by research which demonstrated how the DL could be used beside the blockchain and multi-agent system to build an intrusion detection system [61]. They reached the same conclusion with more than performance and arrived at the best method offering the highest performance in terms of detection, which is DNN within a simulation with the blockchain technique in the transport layer.

A number of ML-based IoT IDSs have been developed that performed on collected data from MQTT protocol (MQTT-IoT-IDS2020 Dataset) to classify the attacks on the MQTT network, including the algorithms: Logistic Regression (LR), k-Nearest Neighbors (KNNs), Gaussian Naive Bayes (GNB), Decision Trees (DTs), Random Forests (RFs) and Support Vector Machine (SVM). The result from this study demonstrated that the common network attacks are easy to identify due to their behaviors and modes, whereas MQTT-based attacks are more complicated to categorize and could be simply imitating benign operations. Nevertheless, using the flow-based features is more appropriate to classify benign and MQTT attacks because they have similar characteristics [62]. Another study that aims to detect attacks on MQTT-IoT Protocol by creating multi-classification models that can feed an IDS on collected data. The classification methods consist of two ensemble methods and deep learning models, such as LSTM, GRU, and XGBoost. The result indicates that ensemble methods achieved better results than DL models [63].

Saipriya et al. designed a machine-learning-empowered IDS to secure IoT nodes [64], in which a number of ML algorithms were used, including RF, DT, GNB, SVM, and LR on the Kddcup99 dataset. This IDS achieved an accuracy of approximately 99%, the highest accuracy being RF with 99.96%, whereas GNB achieved the lowest with 95.05%.

4.2. Machine Learning and Deep Learning in Detecting Attacks in the Middle Layer

Regarding the latest review of the previous studies, the most common attacks affecting IoT environments are DoS attacks, DDoS attacks, Botnet, malware, and malicious data [78]. However, there are many solutions that could be implemented by using DL and ML, which are summarized below:

Detecting attacks in the IoT network , such as Dos, DDoS, and Botnet. Table 2 shows the previous studies that focused on detecting network attacks with the relevant details.

Table 2. ML/DL techniques against IoT network attacks.

Works	Attacks	Methods and Algorithms	Dataset and Results
Aliandy et al. [65]	DoS in MQTT-based IoT systems	Generalized linear model (GLM), RF, GBM, DL, Stack Ensemble	Data from Indonesia oil services company and the result was that the Stack Ensemble offers the best prediction accuracy and achieved the best results, followed by RF technique.
Satam et al. [66]	DoS attacks, Random Signal Attacks, and Replay Attacks	DT, AdaBoost, SVM, Naive Bayes, Ripper, and Bagging	Two datasets: the first was collected from Bluetooth traffic and the second from three temperature sensors. In the sensor datasets, the best performing technique was AdaBoost, whereas the bagging-based model was the best on the Bluetooth dataset.
Touga et al. [79]	DDoS attacks, Botnet	Probabilistic (BNN) Bayesian neural networks and normal Bayesian neural networks	Two datasets: Botnet-IoT and UNSWNB-15; 100% accuracy on Botnet-IoT 99.99% accuracy on UNSWNB-15.
Khempetch et al. [80]	DDoS attack (Syn Flood, UDP, and UDP-Lag)	DNN and long short-term memory (LSTM) algorithm	CICDDoS2019 dataset and the results indicate that 99.90–9.97% of all three types of DDoS attacks were identified.
Roopak et al. [67]	DDoS attack	Four different DL models: multilayer perceptron (MLP), 1d-CNN, Long Short-Term Memory (LSTM), CNN + LSTM	CICIDS2017 dataset and the highest accuracy was 97.16% using the CNN + LSTN model, whereas the lowest accuracy was with the MLP.
Brun et al. [81]	TCP, SYN attacks on IoT gateways	Dense RNN	Used a constructed dataset and it was able to predict with high-performance.
Dong et al. [68]	Botnet detection	Classify Botnet traffic and normal traffic based on ELM algorithm	ISCX-Bot-2014 dataset and the accuracy of ELM model was 98.67%.
Parra et al. [69]	Phishing and Botnet attack	Using both methods: DCNN with cloud-based LSTM	Two data sets: created a dataset and N_BaIoT . CNN models are efficient at detecting phishing attacks with an accuracy of 94.3% and detect Botnet attacks with an accuracy of 94.80%.
Li et al. [82]	Vulnerability detection	Automatically VulDeeLocator	An automatic software vulnerability detector was developed using deep learning techniques, with accuracy up to 80.0%.

Table 2. *Cont.*

Works	Attacks	Methods and Algorithms	Dataset and Results
Aamir et al. [70]	DDoS attack	It provides a clustering-based approach for identifying network traffic flows, including normal and DDoS traffic by using kNN, SVM and RF models	OPNET Modeler 14.5 simulator resulted in an accuracy of 95% in kNN, 92% in SVM and 96.66% in RF.
Roy et al. [83]	Network attacks	Proposed a two-layer hierarchical approach using fog layer resources by employing a multi-layered feedforward NNs	The proposed approach for both datasets CICIDS2017 and NSL-KD showed better result than existing IDs and using the fog-cloud design reduced power and time consumption.

4.3. Machine Learning and Deep Learning in Detecting Attacks in the Software Application Layer

The latest study that sought to predict and identify anomalous and malicious data used collected data from four sensors and then applied the following supervised ML algorithms: DT, RF, LR, SVM, and KNN. The results from this study demonstrate that the K-NN is the best-performing algorithm with an accuracy of 96.5% and a shorter execution time [71]. Another study focused on detecting malware attacks on the Internet of Medical Things (IoMT); they designed a hybrid architecture software-defined networking (SDN) model (CNN-LSTM) driven by DL techniques and then applied this method to an IoT malware dataset, giving an accuracy of 99.83% when using a high-capacity CPU and memory [84].

There are many studies that proposed methods and frameworks to secure IoT environments using ML and DL that have been applied to various datasets. One of these proposed a new method consisting of a time window and classification of transformer-based traffic [85]. The experiment was implemented on the IoT23 dataset and resulted in the best performance in all of the tested scenarios. Another study proposed a framework to detect and attribute cyber attacks on IoT-Enabled cyber-physical systems (CPSs). The framework consisted of various DNNs and two unsupervised stacked autoencoders, principal component analysis (PCA), and a DT classifier to detect the samples of attacks. This framework was tested using two datasets: the first was collected at the Mississippi State University from a gas pipeline system, and the second was an SWaT data set collected at the Singapore University of Technology from a water treatment system and resulted in better results for detecting and attributing than had been achieved in prior works [72].

One study proposed SecureDeepNet-IoT, a DL application for detecting invasive attacks in industrial IoT sensing systems. It used many DL algorithms, such as DBL, and deep autoencoders and was tested using the UNSW-NB15 dataset, achieving accuracy of up to 95.05% for DBNs and 94.39% for DNNs [73]. In [74], they revealed the result of using ML algorithms to secure smart city applications, such as LR, SVM, DT, RF, an artificial neural network (ANN), and KNN to detect and mitigate the attacks [86,87]. The study used various methods to improve the detection systems of bagging, boosting and stacking. They used two datasets, UNSW-NB15 and CICIDS2017, to evaluate the performance of these methods. The study used clustering methods to identify the attacks in IoT-Based Traffic Signal Systems and implemented various ML methods by using two real-time datasets that were taken from data.gov. Utilizing clustering approaches, such as *K-Means, K-Medoids, RF, and linear discriminant analysis (LDA) methods*, the results show that RF offered the best algorithm to verify the attacks, whereas the K-Medoids was the worst [75].

4.4. Machine Learning and Deep Learning in Detecting Threats and Vulnerabilities across All Layers

ML and DL have recently been used to detect threats and vulnerabilities. There are a few studies that focused on the detection of vulnerabilities. The first of these aimed to support the monitoring tool for detecting vulnerabilities in critical infrastructure by

using DL techniques. It works by taking snapshots of many control panels of industrial control systems and then classifying the snapshots by choosing the best CNN architecture depending on the result of comparing the transfer learning and fine-tuning on ImageNet. The experiment was applied by using a critical infrastructure dataset (CRINF-300), and the result with the best architecture was MobileNet-V1, which had the highest accuracy and F1-score [88]. The second study aimed to identify the vulnerabilities in IoT applications by using DL algorithms to analyze the flaw flow. The method entails analyzing the source code and then transferring the code token and taint flow into vectors. This method was applied using two datasets, Corpus1 and Corpus2, and the resulting in accuracy of the prediction models was 77.78–92.59% for Corpus1 and 61.11–87.03% for Corpus2 [89]. Another study focused on detecting vulnerable IoT devices to protect telecommunication service providers. This study sought to detect specific IoT devices on a domestic NAT network to protect the telcos' infrastructure. The ML-based method identifies the IoT devices from the list and then classifies them in a centered training. The result from this study indicated that the long short-term memory networks (LGBM) algorithm offered the best detection result [90].

Ullah et al. developed a combination of TensorFlow deep neural networks to detect plagiarized software and a CNN to detect files that are infected within malware depending on the color image visualization through the IoT network. It was evaluated on Google Code Jam (GCJ) and Mailing datasets and resulted in it having the best performance compared with other methods in this period [91].

5. Vulnerability Detection and Classification

This section focuses on detecting the vulnerabilities in IoT ecosystems by using ML and DL based on the vulnerability's location and the impacted severity in IoT devices.

5.1. Procedure

Detecting vulnerability in IoT environments offers the means to detect existing vulnerabilities by designing and training a model to detect the common pattern of these vulnerabilities, which is similar to the instance-based models such as the vulnerabilities from CVEs and CWEs. However, it could be developed to detect new vulnerabilities that relate to the IoT ecosystems by designing a learning-based model that analyses the packets on the IoT ecosystems to extract the main features and analyze them to detect the vulnerabilities and finally report them. It may develop the model that proposed in [92] to be effective for IoT ecosystems. Figure 4 illustrates that the workflow for detecting vulnerabilities in IoT devices, which is similarly followed by some previous studies. The main steps that ML should follow can be summarized in three main steps: *preparing data, modeling* and *deployment* with some sub-process in order to detect and classify the vulnerabilities and attacks as:

- **Data collection**: collected data from real-life scenarios from IoT networks which connect to other devices such as mobiles and computers; it consists of normal use traffic and suspected traffic trying to conduct some attacks. The data type could be network traffic, logs, or other related data.
- **Data preparation**: data cleaning or data preparation acts as a critical step in ML methods, especially when the data is collected from heterogeneous sources. It aims to identify errors and correct them by using different types of ML algorithms. In some complex cases, the data need to be cleaned and transformed. This stage could affect the result of the whole procedure. The quality of data can be measured in terms of its accuracy, balance, and completeness [93]. Any errors in this stage could result in enormous mistakes in the predictive models. The problems with the data could be simple, such as empty columns and duplicate rows or, in some cases, different types of data. In this stage, there are many sub-steps that should be followed, such as cleaning data, which focuses on identifying and correcting the faults in the dataset. Then, it is necessary to select features which aim to find the most important inputs. Next, transform data, which aims to change the size or allocate the variables. Then,

the features that work to use the existing data are engineered to gain new variables. Finally, reduce the dimensional, which aims to generate the compacted predictions of the data [94].

- **Feature extraction**: It aims to identify the most useful features in the data to feed the model, which could be conducted using a DL technique and creating some useful features which help the classifying model to make a decision. In some cases, in a DL model, this step does not need to be performed manually due to the DL model's ability to learn to extract them during the training phase, which is the biggest advantage of DL [95].
- **Model training**: A model can be trained to learn (define) the best values for all the weights and the bias from trained data. It could differ depending on the type of algorithms; for example, in supervised learning, an ML algorithm builds a model by investigating many examples and attempting to find a good model which reduces cost; this method is called practical risk minimization [94]. The model is usually trained in part of the main dataset by splitting them into training data and excused (tested) data which will be training them in the predictive model.
- **Model evaluation**: This aims to evaluate the trained model using various metrics with predicted models to assess the quality of the ML model, such as calculating the accuracy, precision, recall, and F1 in these algorithms to ensure the effectiveness [96].
- **Tuning/hyperparameterizing the model**: This is an optional step for some algorithms, such as DT and SVM. The tuning of hyperparameters is the process of selecting the best hyperparameters for an algorithm with the aim of increasing the effectiveness of its task [97].
- **Detection**: This stage aims to detect vulnerabilities, attacks, or threats in the IoT ecosystems; it uses the tested data from the main dataset. The result from this step differs depending on the dataset and algorithms used; it could be supervised, unsupervised or semi-supervised. In addition, this step may utilize another dataset to approve the performance of the detection model. The output from this step is forwarded to the classification step to determine the type of detected data.
- **Classification**: This is the final stage which is deployed in order to classify and analyze the upcoming data to determine whether or not they are normal using various types of ML algorithms and classifiers. In some cases, these data are categorized into numerous types of non-normal data, which is called multi-classification. If the data is non-normal, it will be passed to the mitigation framework.

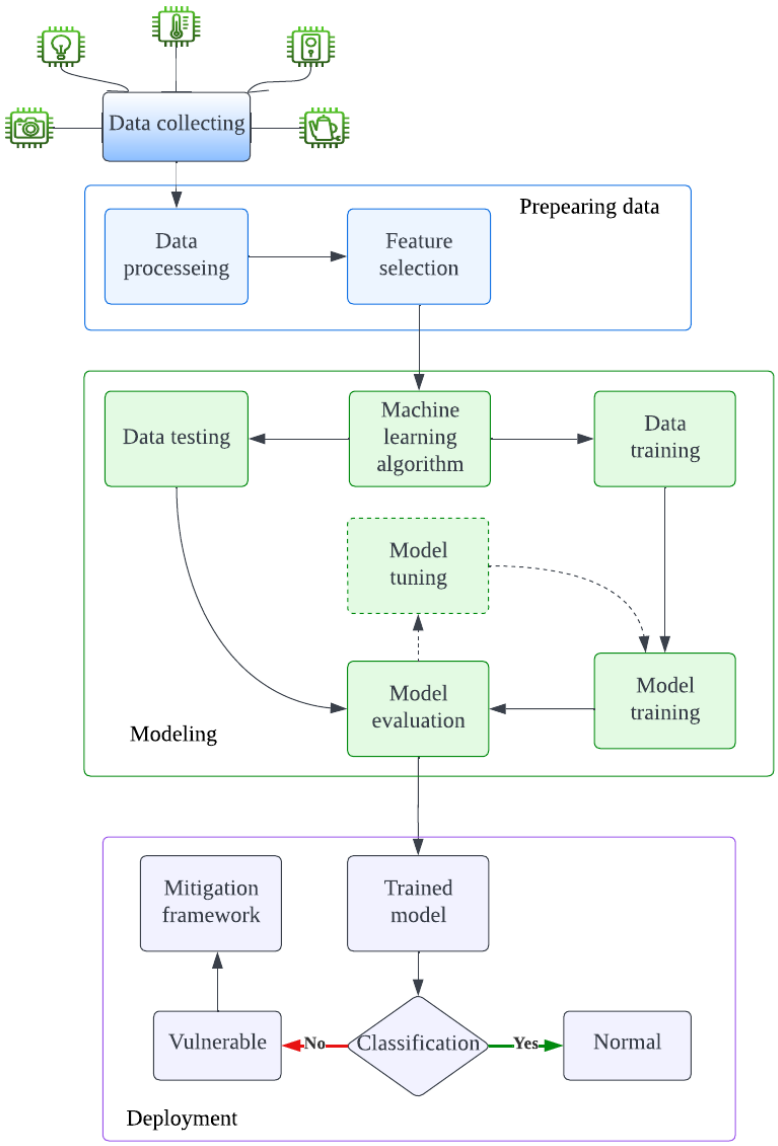

Figure 4. Machine learning workflow for detecting vulnerabilities in IoT devices.

5.2. Research Challenges

According to the common vulnerability scoring system (CVSS v3) developed by NIST, the vulnerability severity can be rated as *{ low, medium, high, critical }*. It is noted that the 'medium' vulnerability increased significantly. In the results from intensive research of the previous studies from 2018 to 2022 using ML and DL to detect attacks, threats, and vulnerabilities, there are some critical challenges that still need to be addressed:

- Incomplete data can bias the detection result of machine learning models or compromise the accuracy of the model. Missing certain details may make it difficult for a classifier to learn representative vulnerability features.
- Most existing works focused on detecting attacks as soon as they start to be implemented; actually, a previously undiscovered vulnerability can be triggered by specific events or conditions, which makes it challenging to predict.
- Resource issues in IoT devices, which are memory and time issues, play an important role in implementing ML and DL models in IoT systems. In addition, implementing these models usually occurs offline, and applying these techniques in real life remains an issue.

- Another challenge is found that the IoT ecosystems work in different ways. For example, some of them work with Bluetooth and others work with sensors which results in different types of data, which means no one method is generally better than others. Therefore, standardization of the IoT ecosystems is important to make improvements in IoT security easier and more effective. In addition, secure by design could also be beneficial for IoT device security.
- IoT devices deal with various types of data with different scales. In addition, all previous research studies were implemented with different datasets. Some of them are collected from wired networks and others from wireless networks. As a result, the accuracy of implementing ML and DL methods is different depending on the type of the datasets and their features.

For instance, the latest study [98] sought to analyze the need for standardized features and the type of attacks in IoT datasets, which was implemented on collected data from a realistic and large-scale testbed network designed at the IoT Lab of UNSW Canberra Cyber and called novel ToN-IoT with a specific focus on its heterogeneity, and combined other datasets were organized from four sources: pcap files, evt logs, sensor data, and operating system logs. Then, various algorithms were applied including GBM, RF, and NN. The results from this experiment indicated that the accuracy from training on the combined dataset, which is standardized, is somewhat better than training on separate datasets.

Other research studies that used the IoT 23 dataset arrived at different results. The authors of [99] aim to find the best solution with high accuracy and less time. They proposed an anomaly detection system model using ML and DL algorithms NB, SVM, DT, and CNN and found that the DL method offered the highest accuracy and shortest implementing time, whereas NB was the least accurate. On the other hand, another study [100] used the same dataset and aimed to explain the need for engineering the feature by using the featureless 1D-CNN ML method. This experiment resulted in 100% accuracy for the proposed model, and it has low memory time-series consumption for analyzing the network traffic.

Meanwhile, another study [101] compared NSL-KDD and NaBIoT datasets to identify the best dataset for wireless IoT IDS and sought to detect the common attacks on IoT systems Dos, probing, U2R, and R2L. It worked by developing an IDS system using different sources of the data and then implemented SVM, RF, NB, and DT methods, resulting in the NaBIoT dataset being the best for attack detection and the SVM classifier having the highest accuracy of 95%. However, the datasets were collected from the wireless network where their IoT devices work by Bluetooth sensors.

The last study used heterogeneous datasets to network intrusion detection by proposing a stacking ensemble framework which used stacked generalization LR, KNN, RF, SVM for two datasets: UNSW NB-15 a packet-based dataset and UGR'16 a flow-based dataset. The result from this ensemble model was that the best predictions of attacks were made using a real-time dataset which offered 97% of accuracy, whereas the accuracy of the simulated datasets was 94% [102].

As a result, due to the characteristics of the IoT systems, network technologies are more heterogeneous than traditional networks, posing new cybersecurity challenges. In addition, the type of the datasets and engineering features play a critical role in research that aims to determine efficient methods.

6. Vulnerability Mitigation

6.1. Vulnerability Mitigation Strategies in IoT

In practice, the most reported vulnerability could be mitigated simply by reducing the attack surface. Vulnerability management and mitigation are cyclical process to prevent malicious attackers from exploiting vulnerabilities in IoT ecosystems.

Mitigating vulnerabilities is essential in order to reduce the risk of an exploit being carried out on the vulnerability and it focuses on implementing internal and external procedures to protect the system from being attacked. Vulnerability mitigation is crucial, especially in IoT environments, because it plays a critical role in avoiding any exploitation

that could be occurred. The main objective of the vulnerability mitigation is ensuring the cybersecurity properties and cyber resilience are involved in the IoT ecosystems. Some studies focused on vulnerability mitigation for a specific method of the IoT ecosystems; for instance, Ref. [103] aims to mitigate the vulnerability of Bluetooth devices in IoT systems by applying countermeasures and mitigating the risk. Mitigating the issues with the Bluetooth system is significantly different from other systems, and it is more complex in upgrading the system because it requires a patching system in its device firmware which is not available for the users. Therefore, this study recommended applying countermeasures to reduce the risk of exploiting the vulnerability, such as increasing the awareness among Bluetooth users and encouraging them to enhance Bluetooth security by taking some steps in their system such as ensuring the encryption for the entire system and increasing the key size, preventing MITM attacks by using combination keys in linking keys and encrypting the link when transmitting data, applying multi-authentication for the device connection, and finally enhancing Bluetooth security by implementing certain applications on the device, such as firewall and file transfer.

Vulnerabilities in IoT could be attributable to weaknesses in software, insecure IoT device configuration or IoT network configuration, depending on the vulnerability category. Three strategies could be followed:

- **Avoid**: Avoid mitigation strategies allow a vulnerable node to be protected permanently by patching its weaknesses or using a secure protocol and encryption.
- **Reduce**: Reduce mitigation strategies stop the spread of weaknesses or stop these vulnerabilities from being used as a privilege to deep with more attacks. Due to the limitation of processing power and remote location in IoT environments, reduction strategies are considered safe for IoT devices.
- **Manage**: Manage strategies involve accepting the risk and implementing another solution for the device, such as setting the access control or allowing trusted communication within the system.
- **Mitigate**: Risk mitigate involves the mitigation of responsibility to a third party who is willing to take on the risk. In order to minimize the organization's exposure to that risk, this procedure is undertaken. External entities that specialize in certain areas of risk management can be delegated certain responsibilities and aspects of risk management.

6.2. Vulnerability Mitigation Frameworks

There are some previous studies that proposed frameworks depending on the location for the mitigation. For instance, Ref. [104] proposed a lightweight innovative cloud-based framework to protect IoT security from Botnet attacks. It consists of a cloud service and IoT security hardware. The security machine monitors the networks to and from the IoT device and checks that traffic it commits to a set of rules, based on a vulnerability mitigation policy. The cloud service stemmed and produced this policy depending on the public corpora of CVEs. There have been a number of studies involving detecting DDoS attacks that have released frameworks in different ways. Ref. [105] designed a framework to detect DDoS in IoT devices. It proposed a private network that is connected to a certain network which is connected to the internet via a border router that works based on resource exhaustion. The border router works in two phases, the first one for analyses and the second one for mentoring. In the first stage, it examines the network traffic and then determines if it is suspicious or not. In the next stage, it works by monitoring the suspicious follow and then classifies if it is DoS or DDoS. Another study that focuses on detecting and mitigating DDoS proposed a detection and mitigation framework based on the software-defined Internet of Things (SD-IoT).

Among the components of the framework are controller pools that involve SD-IoT controllers, SD-IoT switches, and IoT devices. This framework contains an algorithm In the switches, a cosine similarity value is calculated by taking the vectors of packet arrivals and multiplying by their cosine similarities, then using the value to decide if a DDoS

attack has happened, and then blocking the source of the attack [106]. In addition, the latest study focuses on mitigating DDoS attacks in IIoT [107], which releases a multi-level DDoS mitigation framework (MLDMF) to mitigate the DDoS attack at the edge computing stage, fog computing stage, and cloud computing stage. It works by collaboration between different nodes in different locations on the whole of the system and prevents software-defined networking (SDN) controllers from being overwhelmed by massive flow; network functions are split between edge and fog computing levels. IoT gateways based on SDN are the main component of the edge computing level. Also, the SDN controllers are the main component along with SDN application servers. Additionally, the cloud is the primary location for storing and analyzing big data. All these procedures require powerful devices to process and store.

In contrast, Ref. [108] defined an intrusion detection and mitigation framework (IoT-IDM) that works by using ML in smart home networks for devices that use OpenFlow software. It starts with monitoring the IoT device's traffic and then checking if there is any suspicious activity, and then if there is malicious activity, it will block the source of these traffic. The ML model in this framework works by learning the signature patterns of the exciting attacks.

Most of the previous frameworks focused on detecting and mitigating specific vulnerabilities. In addition, most of those frameworks need a specifically designed appliance, which is costly for using one device to protect the devices from one specific attack.

6.3. The Proposed Vulnerability Mitigation Framework

This framework aims to use ML to identify and mitigate IoT vulnerabilities. Figure 5 details the vulnerability mitigation framework in IoT systems, in which ML techniques can help with the identification and classification to detect the existing vulnerabilities by using CVEs and CWEs datasets and identify unknown vulnerabilities by learning the patterns of the existing vulnerabilities such as learning the patterns from previous scenarios. The three strategies: *avoid, reduce, management* can help deploy IoT end security devices, and a continuous monitoring strategy can help detect and monitor new threats in real time.

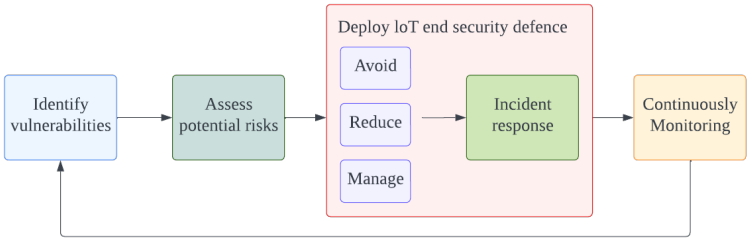

Figure 5. Vulnerability mitigation framework in IoT devices.

Turning to more details:

(1) *Identify vulnerabilities.* Scanning the network to identify the vulnerabilities in the whole IoT ecosystem using the developed ML and DL-based model to detect the vulnerable points such as the proposed model in [90] based on LGBM algorithm which produced an excellent detection model. Then, using an ML-based model to classify these vulnerabilities to determine the threats level and layer(place) of them such as feed-forward neural network (FNN), which is a combined model used for classification and achieves great classification results [109].

(2) *Assess potential risks.* Assess the risk level of the detected vulnerabilities to choose the mitigation strategies depending on the risk rate and move the output to the next stage. The risk rate could be predicted by a training-based learning model depending on the previous dataset attacks, such as IoT23.

(3) *Deploy the security defense.* Deploy the defense by determining the type of incident response, such as performing security controls—*access control, procedure controls, technical controls, compliance controls, etc.*

- If the mitigation strategy is 'Avoid', implement the security defense and the incident response immediately, such as isolating the device and then patching the system or implementing encryption protocol.
- If the strategy is 'Reduce', reduce the risk by performing the security control, such as updating the system and re-configuring the security control.
- If the strategy is 'Manage', accept the incident and then make updates for the security control and security configuration.

All of these procedures could be implemented using multi-classification in DL algorithms, such as deep neural networks and deep decision trees.

(4) *Continuous Monitoring.* Monitoring the traffic in the whole system by using certain techniques and procedures to prevent security issues and reduce the risk in IoT environments.

One of the most effective algorithms for this framework is DL due to its characteristics which provide capabilities for learning more abstract features, reducing the complexity of training for the model, providing high accuracy, handling huge datasets, and providing support for transfer of learning [110]. For more specialties, Bayesian neural networks (BNN) are recommended due to their effectiveness for use in systems that have time and memory limitations such as IoT devices.

6.4. Mitigating Exploitation of Vulnerability in IoT

In an IoT ecosystem, most vulnerabilities can be categorized into three groups: *memory management vulnerability, API vulnerability, and side-channel vulnerability* [111–113]. For different vulnerabilities, there are different ways to mitigate the potential exploitation.

- **Memory management vulnerability**. Managing the memory and controlling the followed traffic and allocation present the critical resources in the IoT environment, so any vulnerabilities in memory management could significantly affect the entire system. These vulnerabilities could be exploited to evade security controls in order to inject malicious code and smash an IoT system.
  As exploiting these vulnerabilities could cause critical potential risks for the IoT ecosystem, the potential threats must be avoided by immediately patching and updating the system. However, patching IoT devices may be complex, so if the device cannot be patched, it could be worthwhile applying a different strategy, such as reducing the exploitation impacts by keeping vulnerable devices off of the internet in order to reduce attack sides and then monitor the system to reveal any behavioral signs of compromise. Moreover, it is necessary to ensure that critical assets are protected by network segmentation. Another study that proposed a framework to detect memory corruption, named FIoT, which works using code execution and fuzzing, resulted in defining 35 IoT devices that have a zero-day vulnerability [114,115].
- **API vulnerability**. The Application Programming Interface (API) is a set of functions and procedures which are used to build and integrate by other software applications usually based on web service applications, with roles for the important platform that collects input from users and connects with the backend services. The security of IoT applications is compromised by several security vulnerabilities, including weak or hardcoded passwords, such as using guessable passwords or using the default manufacturing passwords. Therefore, the most important procedure in IoT security is confirming that only the authorized users can communicate with APIs [116,117].
  Mitigating the exploitation of API vulnerability could be resolved by using many procedures that could be implemented to ensure the security of APIs. Mitigating the insecure backend API requires strong authentication and authorization methods. ML and DL could be used to detect and determine potential threats in API along with a

strong encryption infrastructure and access controls such as using a robust primary key on the IoT infrastructure [118].

- **Side channel vulnerability**. This is a type of security attack which targets the indirect effects of a system's hardware or software rather than directly targeting the code to gather sensitive data such as cryptography keys. It could be exploited by computing or investigating several parameters from a chip or a system, such as execution time and electromagnetic radiation, which are commonly used in IoT environments. It can be categorized into three types: memory cache attacks, which use a shared physical system to monitor the cache accesses.; timing attacks aim to establish patterns by observing the computing time; and power-monitoring attacks aim to track the hardware's power consumption during computation [119,120].

Mitigating the exploitation from the channel side is difficult because these types of attacks are difficult to detect in action owing to the fact they do not leave any impact or any changes on the systems. However, it could be reduced by implementing certain countermeasures. Due to the side channel attacks in IoT usually occurring based on the data leaked over the side channel, countermeasures can be applied such as saving the implementation details of the system, restricting the physical access, and controlling the logged permissions [121]. In some cases, using ML against these attacks could be useful; for example, designing an algorithm that works to confuse the monitoring, such as running random needless processes and running some components in order to generate additional power and computing which are not actually used in the realrandomization, as [122] proposed a framework that aims to produce a continuous stack of randomization. However, these schemes could be insufficient in IoT systems because they have a limitation in terms of memory and computing consumption, so these methods could be used in some cases from an external source within the IoT system, and these sources work by generating electromagnetic radiation in order to hide the actual signals.

7. Research Trends and Directions

The incidence of cyber crime is rising and automated vulnerability management has become a critical lifeline for IoT security against cyber threats. ML-empowered smarter vulnerability management has become a trend that is expected to manage both known and unknown vulnerabilities facing IoT ecosystems. The key research trends fall into the following aspects:

- Making IoT systems more intelligent to detect their vulnerabilities by using ML and DL techniques could be more efficient to enhance security. Many studies focused on detecting attacks and malicious data, but detecting the vulnerabilities would be more efficient. Making the IoT device detect known and unknown vulnerabilities in their environment and sending alerts to the consumer would improve the whole IoT ecosystem's security.
- Time consumption and memory capacity in IoT devices pose the biggest challenge as has been noticed from the previous sections; most of the proposed solutions take time to process these huge data starting with analysis the incoming traffic then decide whether a malicious or not. However, they could be addressed by utilizing compression ML and DL models and reducing the use of cache memory by sharing the data with other devices such as external tools or resources, as well as avoiding unnecessary computations.
- Standardization of the IoT structure, infrastructure, and scale of the data used is a significant issue regarding developing techniques for improving security in IoT environments due to the diversity of the IoT and its data. However, it could be addressed by using algorithms in IoT devices to transform and standardize the data before processing them. However, it leads to the challenges associated with IoT ecosystems which are memory and time consumption.
- Creating cyber resilience in IoT involves not only preventing cyber attacks, but also managing adaptability, recovery, and preserving critical functions even in the face of

difficulty. Although the best efforts are made, breaches may occur, so the key is to ensure that the IoT ecosystem can recover and remain to deliver essential functions while saving its critical assets.

8. Conclusions

Unlike traditional security computer systems, ML techniques have become a critical aspect of securing the IoT. This work focused on ML-empowered IoT vulnerability detection and identification solutions which are expected to automatically manage both known and unknown vulnerabilities in IoT ecosystems based on recent research using ML and DL in a comprehensive manner. It investigates potential IoT vulnerabilities at each layer in the architecture and summarizes how machine learning can be employed to detect vulnerabilities in IoT devices. In addition, recent research trends on machine learning-based vulnerability detection are summarized and analyzed in each IoT layer in order to detect and mitigate vulnerabilities in IoT environments. It also analyzed the available strategies to mitigate these vulnerabilities. The vulnerability mitigation strategies framework was proposed to enhance IoT security from potential threats. The result from this study indicated that due to the advantages of ML and DL, it is crucially needed for using ML and DL techniques to detect and mitigate IoT vulnerabilities in all fields in order to enhance the security and ensure the integrity, availability, authentication, and authorization are applied for these devices.

Author Contributions: Methodology, S.B.H.; Validation, L.X.; Formal analysis, S.B.H.; Investigation, S.L.; Resources, S.H.; Writing—original draft, S.B.H.; Writing—review & editing, S.L. and L.X.; Supervision, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. Li, S.; Zhao, S.; Min, G.; Qi, L.; Liu, G. Lightweight privacy-preserving scheme using homomorphic encryption in industrial Internet of Things. *IEEE Internet Things J.* **2021**, *9*, 14542–14550. [CrossRef]
2. Zhao, S.; Li, S.; Qi, L.; Xu, L.D. Computational Intelligence Enabled Cybersecurity for the Internet of Things. *IEEE Trans. Emerg. Top. Comput. Intell.* **2020**, *4*, 666–674. [CrossRef]
3. Arshad, J.; Azad, M.A.; Amad, R.; Salah, K.; Alazab, M.; Iqbal, R. A review of performance, energy and privacy of intrusion detection systems for IoT. *Electronics* **2020**, *9*, 629. [CrossRef]
4. Mercer, D. Smart Home Will Drive Internet of Things To 50 Billion Devices. Available online: https://www.strategyanalytics.com/strategy-analytics/news/strategy-analytics-press-releases/strategy-analytics-press-release/2017/10/26/smart-home-will-drive-Internet-of-things-to-50-billion-devices-says-strategy-analytics (accessed on 1 January 2023)
5. Ashton, K. Making sense of IoT. In *How the Internet of Things Became Humanity's Nervous System*; Hewlett Packard Enterprise: Spring, TX, USA, 2017.
6. Jabraeil Jamali, M.A.; Bahrami, B.; Heidari, A.; Allahverdizadeh, P.; Norouzi, F. IoT architecture. In *Towards the Internet of Things*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 9–31.
7. Honar Pajooh, H.; Rashid, M.; Alam, F.; Demidenko, S. Multi-layer blockchain-based security architecture for internet of things. *Sensors* **2021**, *21*, 772. [CrossRef] [PubMed]
8. Rana, M.; Shafiq, A.; Altaf, I.; Alazab, M.; Mahmood, K.; Chaudhry, S.A.; Zikria, Y.B. A secure and lightweight authentication scheme for next generation IoT infrastructure. *Comput. Commun.* **2021**, *165*, 85–96. [CrossRef]
9. Azrour, M.; Mabrouki, J.; Guezzaz, A.; Kanwal, A. Internet of things security: Challenges and key issues. *Secur. Commun. Netw.* **2021**, *2021*, 5533843. [CrossRef]
10. Wang, C.; Dong, S.; Zhao, X.; Papanastasiou, G.; Zhang, H.; Yang, G. SaliencyGAN: Deep learning semisupervised salient object detection in the fog of IoT. *IEEE Trans. Ind. Inform.* **2019**, *16*, 2667–2676. [CrossRef]
11. Zhou, Y.; Han, M.; Liu, L.; He, J.S.; Wang, Y. Deep learning approach for cyberattack detection. In Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Honolulu, HI, USA, 15–19 April 2018; IEEE: New York, NY, USA, 2018; pp. 262–267.
12. Neshenko, N.; Bou-Harb, E.; Crichigno, J.; Kaddoum, G.; Ghani, N. Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on Internet-scale IoT exploitations. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2702–2733. [CrossRef]

13. Xie, W.; Jiang, Y.; Tang, Y.; Ding, N.; Gao, Y. Vulnerability detection in iot firmware: A survey. In Proceedings of the 2017 IEEE 23rd International Conference on Parallel and dIstributed Systems (ICPADS), Shenzhen, China, 15–17 December 2017; IEEE: New York, NY, USA, 2017; pp. 769–772.
14. Feng, X.; Zhu, X.; Han, Q.L.; Zhou, W.; Wen, S.; Xiang, Y. Detecting vulnerability on IoT device firmware: A survey. *IEEE/CAA J. Autom. Sin.* **2022**, *10*, 25–41. [CrossRef]
15. Meneghello, F.; Calore, M.; Zucchetto, D.; Polese, M.; Zanella, A. IoT: Internet of threats? A survey of practical security vulnerabilities in real IoT devices. *IEEE Internet Things J.* **2019**, *6*, 8182–8201. [CrossRef]
16. Yu, M.; Zhuge, J.; Cao, M.; Shi, Z.; Jiang, L. A survey of security vulnerability analysis, discovery, detection, and mitigation on IoT devices. *Future Internet* **2020**, *12*, 27. [CrossRef]
17. Ahanger, T.A.; Aljumah, A.; Atiquzzaman, M. State-of-the-art survey of artificial intelligent techniques for IoT security. *Comput. Netw.* **2022**, *206*, 108771. [CrossRef]
18. OWASP. *Internet of Things*; OWASP Foundation: Bel Air, MA, USA, 2022.
19. Qu, J. Research on Password Detection Technology of IoT Equipment Based on Wide Area Network. *ICT Express* **2021**, *8*, 213–219. [CrossRef]
20. Verma, R.S.; Chandavarkar, B.R.; Nazareth, P. Mitigation of hard-coded credentials related attacks using QR code and secured web service for IoT. In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019; IEEE: New York, NY, USA, 2019; pp. 1–5.
21. Sun, H.M.; Chen, Y.H.; Lin, Y.H. oPass: A User Authentication Protocol Resistant to Password Stealing and Password Reuse Attacks. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 651–663. [CrossRef]
22. Mouris, D.; Tsoutsos, N.G. Zilch: A Framework for Deploying Transparent Zero-Knowledge Proofs. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 3269–3284. [CrossRef]
23. Erendor, M.E.; Yildirim, M. Cybersecurity Awareness in Online Education: A Case Study Analysis. *IEEE Access* **2022**, *10*, 52319–52335. [CrossRef]
24. Alladi, T.; Chamola, V.; Sikdar, B.; Choo, K.K.R. Consumer IoT: Security vulnerability case studies and solutions. *IEEE Consum. Electron. Mag.* **2020**, *9*, 17–25. [CrossRef]
25. Chatterjee, D.; Boyapally, H.; Patranabis, S.; Chatterjee, U.; Hazra, A.; Mukhopadhyay, D. Physically Related Functions: Exploiting Related Inputs of PUFs for Authenticated-Key Exchange. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 3847–3862. [CrossRef]
26. Meng, Q.; Nian, X.; Chen, Y.; Chen, Z. Attack-Resilient Distributed Nash Equilibrium Seeking of Uncertain Multiagent Systems Over Unreliable Communication Networks. In *IEEE Transactions on Neural Networks and Learning Systems*; IEEE: New York, NY, USA, 2022; pp. 1–15. [CrossRef]
27. Nadir, I.; Mahmood, H.; Asadullah, G. A taxonomy of IoT firmware security and principal firmware analysis techniques. *Int. J. Crit. Infrastruct. Prot.* **2022**, *38*, 100552. [CrossRef]
28. Morgner, P.; Mai, C.; Koschate-Fischer, N.; Freiling, F.; Benenson, Z. Security Update Labels: Establishing Economic Incentives for Security Patching of IoT Consumer Products. In Proceedings of the 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 18–21 May 2020; pp. 429–446. [CrossRef]
29. Li, S. Zero trust based internet of things. *EAI Endorsed Trans. Internet Things* **2019**, *5*, e1.
30. Arthi, R.; Krishnaveni, S. Design and Development of IOT Testbed with DDoS Attack for Cyber Security Research. In Proceedings of the 2021 3rd International Conference on Signal Processing and Communication (ICPSC), Coimbatore, India, 13–14 May 2021; pp. 586–590. [CrossRef]
31. Cao, H.; Brown, M.; Chen, L.; Smith, R.; Wachowicz, M. Lessons Learned from Integrating Batch and Stream Processing using IoT Data. In Proceedings of the 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), Granada, Spain, 22–25 October 2019; pp. 32–34. [CrossRef]
32. Alrawi, O. Security Evaluation of Home-Based IoT Deployments. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019.
33. Thapaliya, B.; Mursi, K.T.; Zhuang, Y. Machine Learning-based Vulnerability Study of Interpose PUFs as Security Primitives for IoT Networks. In Proceedings of the 2021 IEEE International Conference on Networking, Architecture and Storage (NAS), Riverside, CA, USA, 24–26 October 2021; pp. 1–7. [CrossRef]
34. Islam, M.J.; Rahman, A.; Kabir, S.; Karim, M.R.; Acharjee, U.K.; Nasir, M.K.; Band, S.S.; Sookhak, M.; Wu, S. Blockchain-SDN-Based Energy-Aware and Distributed Secure Architecture for IoT in Smart Cities. *IEEE Internet Things J.* **2022**, *9*, 3850–3864. [CrossRef]
35. Chandavarkar, B. Hardcoded credentials and insecure data transfer in IoT: National and international status. In Proceedings of the 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 1–3 July 2020; IEEE: New York, NY, USA, 2020; pp. 1–7.
36. Shin, S.; Seto, Y. Development of iot security exercise contents for cyber security exercise system. In Proceedings of the 2020 13th International Conference on Human System Interaction (HSI), Tokyo, Japan, 6–8 June 2020; IEEE: New York, NY, USA, 2020; pp. 1–6.
37. Singh, S.K.; Park, J.H. TaLWaR: Blockchain-Based Trust Management Scheme for Smart Enterprises With Augmented Intelligence. *IEEE Trans. Ind. Inform.* **2023**, *19*, 626–634. [CrossRef]

38. Kotenko, I.; Doynikova, E.; Fedorchenko, A.; Desnitsky, V. Automation of Asset Inventory for Cyber Security: Investigation of Event Correlation-Based Technique. *Electronics* **2022**, *11*, 2368. [CrossRef]
39. Asef, P.; Taheri, R.; Shojafar, M.; Mporas, I.; Tafazolli, R. SIEMS: A Secure Intelligent Energy Management System for Industrial IoT Applications. *IEEE Trans. Ind. Inform.* **2023**, *19*, 1039–1050. [CrossRef]
40. Adil, M.; Jan, M.A.; Liu, Y.; Abulkasim, H.; Farouk, A.; Song, H. A Systematic Survey: Security Threats to UAV-Aided IoT Applications, Taxonomy, Current Challenges and Requirements With Future Research Directions. In *IEEE Transactions on Intelligent Transportation Systems*; IEEE: New York, NY, USA, 2022; pp. 1–19. [CrossRef]
41. Choudhary, Y.; Umamaheswari, B.; Kumawat, V. A Study of Threats, Vulnerabilities and Countermeasures: An IoT Perspective. *Humanities* **2021**, *8*, 39–45. [CrossRef]
42. Pal, R.; Huang, Z.; Yin, X.; Lototsky, S.; De, S.; Tarkoma, S.; Liu, M.; Crowcroft, J.; Sastry, N. Aggregate Cyber-Risk Management in the IoT Age: Cautionary Statistics for (Re)Insurers and Likes. *IEEE Internet Things J.* **2021**, *8*, 7360–7371. [CrossRef]
43. Wang, H.; Barriga, L.; Vahidi, A.; Raza, S. Machine Learning for Security at the IoT Edge—A Feasibility Study. In Proceedings of the 2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW), Monterey, CA, USA, 4–7 November 2019; pp. 7–12. [CrossRef]
44. Tao, M.; Ota, K.; Dong, M. Locating compromised data sources in IoT-enabled smart cities: A great-alternative-region-based approach. *IEEE Trans. Ind. Inform.* **2018**, *14*, 2579–2587. [CrossRef]
45. Chen, L.; Xue, L.; Huang, H.; Wang, W.; Cao, M.; Xiao, F. Double Rainbows: A Promising Distributed Data Sharing in Augmented Intelligence of Things. *IEEE Trans. Ind. Inform.* **2023**, *19*, 653–661. [CrossRef]
46. Ryon, L.; Martintoni, D. Field Loadable Software Confidentiality Protection. In Proceedings of the 2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC), Portsmouth, VA, USA, 18–22 September 2022; pp. 1–6. [CrossRef]
47. Tong, F.; Chen, X.; Wang, K.; Zhang, Y. CCAP: A Complete Cross-Domain Authentication Based on Blockchain for Internet of Things. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 3789–3800. [CrossRef]
48. Dofe, J. Thermal Side-channel Leakage Protection in Monolithic Three Dimensional Integrated Circuits. In Proceedings of the 2022 IEEE 35th International System-on-Chip Conference (SOCC), Belfast, UK, 5–8 September 2022; pp. 1–2. [CrossRef]
49. Gourisetti, S.N.G.; Mylrea, M.; Patangia, H. Cybersecurity Vulnerability Mitigation Framework Through Empirical Paradigm (CyFEr): Prioritized Gap Analysis. *IEEE Syst. J.* **2020**, *14*, 1897–1908. [CrossRef]
50. Kol, M. JSOF Research Lab. 2020. Available online: https://www.jsof-tech.com/wp-content/uploads/2020/08/Ripple20_CVE-2020-11901-August20.pdf (accessed on 3 February 2023).
51. Dong, C.; He, G.; Liu, X.; Yang, Y.; Guo, W. A multi-layer hardware trojan protection framework for IoT chips. *IEEE Access* **2019**, *7*, 23628–23639. [CrossRef]
52. Adina, P.; Shahzad, M. A Distributed & Lightweight Framework to Secure IoT Networks Against Network Layer Attacks. In Proceedings of the 2022 International Conference on Computer Communications and Networks (ICCCN), Honolulu, HI, USA, 25–28 July 2022; IEEE: New York, NY, USA, 2022; pp. 1–9.
53. Nebbione, G.; Calzarossa, M.C. Security of IoT application layer protocols: Challenges and findings. *Future Internet* **2020**, *12*, 55. [CrossRef]
54. Mocrii, D.; Chen, Y.; Musilek, P. IoT-based smart homes: A review of system architecture, software, communications, privacy and security. *Internet Things* **2018**, *1*, 81–98. [CrossRef]
55. Burhan, M.; Rehman, R.A.; Khan, B.; Kim, B.S. IoT elements, layered architectures and security issues: A comprehensive survey. *Sensors* **2018**, *18*, 2796. [CrossRef]
56. Meng, H.; Thing, V.L.; Cheng, Y.; Dai, Z.; Zhang, L. A survey of Android exploits in the wild. *Comput. Secur.* **2018**, *76*, 71–91. [CrossRef]
57. Hosmer, C. IoT vulnerabilities. In *Defending IoT Infrastructures with the Raspberry Pi*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 1–15.
58. Chakraborty, S.; Krishna, R.; Ding, Y.; Ray, B. Deep learning based vulnerability detection: Are we there yet. *IEEE Trans. Softw. Eng.* **2021**, *48*, 9. [CrossRef]
59. Selvapandian, D.; Santhosh, R. Deep learning approach for intrusion detection in IoT-multi cloud environment. *Autom. Softw. Eng.* **2021**, *28*, 19. [CrossRef]
60. Liang, C.; Shanmugam, B.; Azam, S.; Jonkman, M.; De Boer, F.; Narayansamy, G. Intrusion detection system for Internet of Things based on a machine learning approach. In Proceedings of the 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), Vellore, India, 30–31 March 2019; IEEE: New York, NY, USA, 2019; pp. 1–6.
61. Liang, C.; Shanmugam, B.; Azam, S.; Karim, A.; Islam, A.; Zamani, M.; Kavianpour, S.; Idris, N.B. Intrusion detection system for the internet of things based on blockchain and multi-agent systems. *Electronics* **2020**, *9*, 1120. [CrossRef]
62. Hindy, H.; Bayne, E.; Bures, M.; Atkinson, R.; Tachtatzis, C.; Bellekens, X. Machine learning based IoT intrusion detection system: An MQTT case study (MQTT-IoT-IDS2020 dataset). In *Selected Papers from the 12th International Networking Conference, 16 November 2020*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 73–84.
63. Alaiz-Moreton, H.; Aveleira-Mata, J.; Ondicol-Garcia, J.; Muñoz-Castañeda, A.L.; García, I.; Benavides, C.; et al. Multiclass classification procedure for detecting attacks on MQTT-IoT protocol. *Complexity* **2019**, *2019*, 6516253. [CrossRef]

64. Saipriya, T.; Anand, M. To Secure IoT sensor nodes through Fog computing. In Proceedings of the 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 4–6 August 2021; IEEE: New York, NY, USA, 2021; pp. 836–844.
65. Grendy, E.; Aliandy, R.J. Denial of Service Classification on Message Queueing Telemetry Transport Protocol at Indonesia Oil Services Company. *J. Theor. Appl. Inf. Technol.* **2022**, *100* , 2289–2299.
66. Satam, S.; Satam, P.; Pacheco, J.; Hariri, S. Security framework for smart cyber infrastructure. *Clust. Comput.* **2022**, *25*, 2767–2778. [CrossRef]
67. Roopak, M.; Tian, G.Y.; Chambers, J. Deep learning models for cyber security in IoT networks. In Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 7–9 January 2019; IEEE: New York, NY, USA, 2019; pp. 452–457.
68. Dong, X.; Dong, C.; Chen, Z.; Cheng, Y.; Chen, B. BotDetector: An extreme learning machine-based Internet of Things botnet detection model. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e3999. [CrossRef]
69. Parra, G.D.L.T.; Rad, P.; Choo, K.K.R.; Beebe, N. Detecting Internet of Things attacks using distributed deep learning. *J. Netw. Comput. Appl.* **2020**, *163*, 102662. [CrossRef]
70. Aamir, M.; Zaidi, S.M.A. Clustering based semi-supervised machine learning for DDoS attack classification. *J. King Saud Univ. Comput. Inf. Sci.* **2021**, *33*, 436–446. [CrossRef]
71. Khan, S.H.; Arko, A.R.; Chakrabarty, A. Anomaly Detection in IoT Using Machine Learning. In *Artificial Intelligence for Cloud and Edge Computing*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 237–254.
72. Jahromi, A.N.; Karimipour, H.; Dehghantanha, A.; Choo, K.K.R. Toward Detection and Attribution of Cyber-Attacks in IoT-Enabled Cyber–Physical Systems. *IEEE Internet Things J.* **2021**, *8*, 13712–13722. [CrossRef]
73. Altan, G. SecureDeepNet-IoT: A deep learning application for invasion detection in industrial Internet of things sensing systems. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4228. [CrossRef]
74. Rashid, M.M.; Kamruzzaman, J.; Hassan, M.M.; Imam, T.; Gordon, S. Cyberattacks detection in iot-based smart city applications using machine learning techniques. *Int. J. Environ. Res. Public Health* **2020**, *17*, 9347. [CrossRef]
75. Zhang, Y.; Dukkipati, C.; Cheng, L.C. Clustering Methods for Identification of Attacks in IoT Based Traffic Signal System. In Proceedings of the 2019 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), Beijing, China, 15–17 August 2019; IEEE: New York, NY, USA, 2019; pp. 24–28.
76. Ferrag, M.A.; Friha, O.; Maglaras, L.; Janicke, H.; Shu, L. Federated deep learning for cyber security in the internet of things: Concepts, applications, and experimental analysis. *IEEE Access* **2021**, *9*, 138509–138542. [CrossRef]
77. Jain, A.; Singh, T.; Sharma, S.K. Security as a solution: An intrusion detection system using a neural network for IoT enabled healthcare ecosystem. *Interdiscip. J. Inf. Knowl. Manag.* **2021**, *16*, 331–369. [CrossRef]
78. Gao, H.; Qiu, B.; Barroso, R.J.D.; Hussain, W.; Xu, Y.; Wang, X. TSMAE: A novel anomaly detection approach for internet of things time series data using memory-augmented autoencoder. In *IEEE Transactions on Network Science and Engineering*; IEEE: New York, NY, USA, 2022.
79. Toğaçar, M. Detecting attacks on IoT devices with probabilistic Bayesian neural networks and hunger games search optimization approaches. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e4418. [CrossRef]
80. Khempetch, T.; Wuttidittachotti, P. DDoS attack detection using deep learning. *IAES Int. J. Artif. Intell.* **2021**, *10*, 382. [CrossRef]
81. Brun, O.; Yin, Y.; Gelenbe, E.; Kadioglu, Y.M.; Augusto-Gonzalez, J.; Ramos, M. Deep learning with dense random neural networks for detecting attacks against IoT-connected home environments. In Proceedings of the International ISCIS Security Workshop, London, UK, 26–27 February 2018; Springer: Cham, Switzerland, 2018; pp. 79–89.
82. Li, Z.; Zou, D.; Xu, S.; Chen, Z.; Zhu, Y.; Jin, H. VulDeeLocator: A Deep Learning-Based Fine-Grained Vulnerability Detector. *IEEE Trans. Dependable Secur. Comput.* **2022**, *19*, 2821–2837. [CrossRef]
83. Roy, S.; Li, J.; Bai, Y. A Two-layer Fog-Cloud Intrusion Detection Model for IoT Networks. *Internet Things* **2022**, *19*, 100557. [CrossRef]
84. Khan, S.; Akhunzada, A. A hybrid DL-driven intelligent SDN-enabled malware detection framework for Internet of Medical Things (IoMT). *Comput. Commun.* **2021**, *170*, 209–216. [CrossRef]
85. Kozik, R.; Pawlicki, M.; Choraś, M. A new method of hybrid time window embedding with transformer-based traffic data classification in IoT-networked environment. *Pattern Anal. Appl.* **2021**, *24*, 1441–1449. [CrossRef]
86. Gao, H.; Xiao, J.; Yin, Y.; Liu, T.; Shi, J. A Mutually Supervised Graph Attention Network for Few-Shot Segmentation: The Perspective of Fully Utilizing Limited Samples. In *IEEE Transactions on Neural Networks and Learning Systems*; IEEE: New York, NY, USA, 2022.
87. Gao, H.; Huang, W.; Liu, T.; Yin, Y.; Li, Y. PPO2: Location Privacy-Oriented Task Offloading to Edge Computing Using Reinforcement Learning for Intelligent Autonomous Transport Systems. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 7. [CrossRef]
88. Blanco-Medina, P.; Fidalgo, E.; Alegre, E.; Vasco-Carofilis, R.A.; Jañez-Martino, F.; Villar, V.F. Detecting vulnerabilities in critical infrastructures by classifying exposed industrial control systems using deep learning. *Appl. Sci.* **2021**, *11*, 367. [CrossRef]
89. Naeem, H.; Alalfi, M.H. Identifying vulnerable IoT applications using deep learning. In Proceedings of the 2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER), London, ON, Canada, 18–21 February 2020; IEEE: New York, NY, USA, 2020; pp. 582–586.

90. Meidan, Y.; Sachidananda, V.; Peng, H.; Sagron, R.; Elovici, Y.; Shabtai, A. A novel approach for detecting vulnerable IoT devices connected behind a home NAT. *Comput. Secur.* **2020**, *97*, 101968. [CrossRef]
91. Ullah, F.; Naeem, H.; Jabbar, S.; Khalid, S.; Latif, M.A.; Al-Turjman, F.; Mostarda, L. Cyber security threats detection in internet of things using deep learning approach. *IEEE Access* **2019**, *7*, 124379–124389. [CrossRef]
92. Li, Z.; Zou, D.; Xu, S.; Ou, X.; Jin, H.; Wang, S.; Deng, Z.; Zhong, Y. Vuldeepecker: A deep learning-based system for vulnerability detection. *arXiv* **2018**, arXiv:1801.01681.
93. Jesmeen, M.; Hossen, J.; Sayeed, S.; Ho, C.; Tawsif, K.; Rahman, A.; Arif, E. A survey on cleaning dirty data using machine learning paradigm for big data analytics. *Indones. J. Electr. Eng. Comput. Sci.* **2018**, *10*, 1234–1243.
94. Brownlee, J. *Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python*; Machine Learning Mastery: Vermont, Australia, 2020.
95. Kasongo, S.M.; Sun, Y. A deep learning method with wrapper based feature extraction for wireless intrusion detection system. *Comput. Secur.* **2020**, *92*, 101752. [CrossRef]
96. Zhou, J.; Gandomi, A.H.; Chen, F.; Holzinger, A. Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronics* **2021**, *10*, 593. [CrossRef]
97. Tune Model Hyperparameters—Azure Machine Learning | Microsoft Learn. 2021. Available online: https://learn.microsoft.com/en-us/azure/machine-learning/component-reference/tune-model-hyperparameters?view=azureml-api-2 (accessed on 14 February 2023).
98. Booij, T.M.; Chiscop, I.; Meeuwissen, E.; Moustafa, N.; den Hartog, F.T. ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets. *IEEE Internet Things J.* **2021**, *9*, 485–496. [CrossRef]
99. Liang, Y.; Vankayalapati, N. Machine Learning and Deep Learning Methods for Better Anomaly Detection in IoT-23 Dataset Cybersecurity. 2021. Available online: https://github.com/yliang725/Anomaly-Detection-IoT23 (accessed on 10 January 2023).
100. Khan, A.; Cotton, C. Detecting attacks on IoT devices using featureless 1D-CNN. In Proceedings of the 2021 IEEE International Conference on Cyber Security and Resilience (CSR), Rhodes, Greece, 26–28 July 2021; IEEE: New York, NY, USA, 2021; pp. 461–466.
101. Seong, T.B.; Ponnusamy, V.; Jhanjhi, N.; Annur, R.; Talib, M. A comparative analysis on traditional wired datasets and the need for wireless datasets for IoT wireless intrusion detection. *Indones. J. Electr. Eng. Comput. Sci.* **2021**, *22*, 1165–1176. [CrossRef]
102. Rajagopal, S.; Kundapur, P.P.; Hareesha, K.S. A stacking ensemble for network intrusion detection using heterogeneous datasets. *Secur. Commun. Netw.* **2020**, *2020*, 4586875. [CrossRef]
103. Lonzetta, A.M.; Cope, P.; Campbell, J.; Mohd, B.J.; Hayajneh, T. Security vulnerabilities in Bluetooth technology as used in IoT. *J. Sens. Actuator Netw.* **2018**, *7*, 28. [CrossRef]
104. Hadar, N.; Siboni, S.; Elovici, Y. A lightweight vulnerability mitigation framework for IoT devices. In Proceedings of the 2017 Workshop on Internet of Things Security and Privacy, Dallas, TX, USA, 3 November 2017; pp. 71–75.
105. Adat, V.; Gupta, B. A DDoS attack mitigation framework for internet of things. In Proceedings of the 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 6–8 April 2017; IEEE: New York, NY, USA, 2017; pp. 2036–2041.
106. Yin, D.; Zhang, L.; Yang, K. A DDoS attack detection and mitigation with software-defined Internet of Things framework. *IEEE Access* **2018**, *6*, 24694–24705. [CrossRef]
107. Yan, Q.; Huang, W.; Luo, X.; Gong, Q.; Yu, F.R. A multi-level DDoS mitigation framework for the industrial Internet of Things. *IEEE Commun. Mag.* **2018**, *56*, 30–36. [CrossRef]
108. Nobakht, M.; Sivaraman, V.; Boreli, R. A host-based intrusion detection and mitigation framework for smart home IoT using OpenFlow. In Proceedings of the 2016 11th International Conference on Availability, Reliability and Security (ARES), Salzburg, Austria, 31 August–2 September 2016; IEEE: New York, NY, USA, 2016; pp. 147–156.
109. Ibitoye, O.; Shafiq, O.; Matrawy, A. Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; IEEE: New York, NY, USA, 2019; pp. 1–6.
110. Wang, J.; Chen, Y.; Hao, S.; Peng, X.; Hu, L. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognit. Lett.* **2019**, *119*, 3–11. [CrossRef]
111. Hore, S.; Moomtaheen, F.; Shah, A.; Ou, X. Towards Optimal Triage and Mitigation of Context-sensitive Cyber Vulnerabilities. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 1270–1285. [CrossRef]
112. *IEEE P11073-40101/D5, July 2020*; IEEE Approved Draft Standard—Health Informatics—Device Interoperability— Part 40101: Cybersecurity—Processes for Vulnerability Assessment. IEEE: New York, NY, USA, 2020; pp. 1–46.
113. Aurisch, T.; Jacke, A. Replication Strategies of Mobile Agents for Autonomous Vulnerability Mitigation. In Proceedings of the 2019 International Conference on Military Communications and Information Systems (ICMCIS), Budva, Montenegro, 14–15 May 2019; pp. 1–6. [CrossRef]
114. Zhu, L.; Fu, X.; Yao, Y.; Zhang, Y.; Wang, H. FIoT: Detecting the memory corruption in lightweight IoT device firmware. In Proceedings of the 2019 18th IEEE International Conference On Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), Rotorua, New Zealand , 5–8 August 2019; IEEE: New York, NY, USA, 2019; pp. 248–255.

115. Nelson, C.; Izraelevitz, J.; Bahar, R.I.; Lehman, T.S. Eliminating Micro-Architectural Side-Channel Attacks using Near Memory Processing. In Proceedings of the 2022 IEEE International Symposium on Secure and Private Execution Environment Design (SEED), Storrs, CT, USA, 26-27 September 2022; pp. 179–189. [CrossRef]
116. Education, I.C. *What is an Application Programming Interface (API)*? IBM: Armonk, NY, USA, 2020.
117. Lee, J.; Jung, S.; Suh, T.; Oh, Y.; Yoon, M.K.; Koo, G. GhostLeg: Selective Memory Coalescing for Secure GPU Architecture. *IEEE Access* **2022**, *10*, 111449–111462. [CrossRef]
118. Siriwardena, P. *Advanced API Security: OAuth 2.0 and Beyond*; Springer: Berlin/Heidelberg, Germany, 2020.
119. Bhunia, S.; Tehranipoor, M. *Hardware Security: A Hands-On Learning Approach*; Morgan Kaufmann: Burlington, MA, USA, 2018.
120. Chen, Z.; Zhang, Q.; Wu, J.; Yan, J.; Xue, J. A Source-Level Instrumentation Framework for the Dynamic Analysis of Memory Safety. *IEEE Trans. Softw. Eng.* **2022**, *49* 2107–2127. [CrossRef]
121. Gavin Wright, A.S.G. What Is a Side-Channel Attack? 2021. Available online: https://www.techtarget.com/searchsecurity/definition/side-channel-attack (accessed on 10 March 2023).
122. Lyerly, R.; Wang, X.; Ravindran, B. Dynamic and Secure Memory Transformation in Userspace. In Proceedings of the European Symposium on Research in Computer Security, Guildford, UK, 14–18 September 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 237–256.