sensors

MDPI

Review

Intrusion Detection in Internet of Things Systems: A Review on Design Approaches Leveraging Multi-Access Edge Computing, Machine Learning, and Datasets

Eric Gyamfi * and Anca Jurcut

School of Computer Science, University College Dublin, D04 V1W8 Dublin, Ireland; anca.jurcut@ucd.ie
* Correspondence: eric.gyamfi@ucdconnect.ie

Abstract: The explosive growth of the Internet of Things (IoT) applications has imposed a dramatic increase of network data and placed a high computation complexity across various connected devices. The IoT devices capture valuable information, which allows the industries or individual users to make critical live dependent decisions. Most of these IoT devices have resource constraints such as low CPU, limited memory, and low energy storage. Hence, these devices are vulnerable to cyber-attacks due to the lack of capacity to run existing general-purpose security software. It creates an inherent risk in IoT networks. The multi-access edge computing (MEC) platform has emerged to mitigate these constraints by relocating complex computing tasks from the IoT devices to the edge. Most of the existing related works are focusing on finding the optimized security solutions to protect the IoT devices. We believe distributed solutions leveraging MEC should draw more attention. This paper presents a comprehensive review of state-of-the-art network intrusion detection systems (NIDS) and security practices for IoT networks. We have analyzed the approaches based on MEC platforms and utilizing machine learning (ML) techniques. The paper also performs a comparative analysis on the public available datasets, evaluation metrics, and deployment strategies employed in the NIDS design. Finally, we propose an NIDS framework for IoT networks leveraging MEC.

Keywords: Internet of Things; multi-access edge computing; network intrusion detection system; machine learning; anomaly systems; IoT device; offloading; distributed NIDS

check for updates

Citation: Gyamfi, E.; Jurcut, A. Intrusion Detection in Internet of Things Systems: A Review on Design Approaches Leveraging Multi-Access Edge Computing, Machine Learning, and Datasets. *Sensors* **2022**, *22*, 3744. https://doi.org/10.3390/s22103744

Academic Editor: Claudia Campolo

Received: 18 March 2022
Accepted: 7 May 2022
Published: 14 May 2022

1. Introduction

The Internet of Things has experienced tremendous growth in area-specific applications such as healthcare, smart transportation systems, smart agriculture, and industries to improve socio-economic development in recent years [1]. These IoT systems are composed of many interconnected sensors, actuators, and varieties of network-enabled devices [2] that exchange different types of data through both the Internet infrastructure and the private networks. The cisco research team predicted an average of 75.3 billion actively connected IoT devices by 2025 [3,4]. The absence of human intervention in data exchange between IoT systems makes it unique from traditional Internet technology. Growth in IoT devices has also increased the data network bandwidth demands. However, most IoT devices have resource constraints, making it challenging to execute the traditional security methods for system protection against cyberattacks. Critical concerns about the IoT device arise when there is a need to process sensitive information. Hence, it is essential to introduce the MEC platform that permits computation to be performed at the network end to address the resource-constraint problems in IoT systems [5,6]. MEC allows IoTs to offload high computational-intensive tasks to the proximal edge server [7].

Since the IoT has become the driving force of the current industrial revolution and the system for collecting live dependent data, it is vital to take cyber-security seriously [8,9]. Hence, there is the need for a Network Intrusion Detection System (NIDS) that can detect current and future attacks to protect the IoT network and the systems built on it.

1.1. IoT Network Intrusion Detection Systems

NIDS monitors the internet traffic across the devices in an IoT network. It acts as a defense line [10], which can identify risks and protect the network from intruders and malicious attacks.

NIDS is the primary tool used to combat network intrusion and various attacks in current computer network systems. NIDS examines and investigates the hosted device's network, user actions, and discovers signatures of known threats and unknown malicious attacks within a network. Another goal of the NIDS is to monitor the IoT network, discover unauthorized intrusions within the network, and enable context-awareness to other devices connected to the network and the execution of necessary defense (firewall rules). NIDS also generates an alert or creates attack flags when it discovers internal and external attacks. Cyber-attackers initiate internal attacks through the compromised IoT devices connected to the network. Third parties outside the leading network initiate external attacks. There are principally three general elements of NIDS [11], namely Observation, Analysis and Detection. The Observation module monitors the network traffics, patterns, and resources. Analysis and Detection are normally the core part of NIDS. They can detect intrusions based on given algorithmic rules. The alert module raised attack flags if an intrusion is detected [11]. Considering that the development of NIDSs for IoT systems represents a significant challenge for information security researchers, this survey has covered the following topics: detection method, NIDS placement strategy, security threats, and validation strategy.

1.2. Motivation and Contribution

Several surveys have been published in diverse research areas related to IoT security, including security frameworks [12], security in context of eHealth [12], privacy issues [13], state of the art and security challenges [14], models, techniques, and tools [15], and attacks [16]. Some of these papers were published during the early stage of IoT system evolution. This paper focuses on the use of ML techniques to enhance security in IoT. Additionally, we investigate how the emergence of MEC can aid to develop a sophisticated security system (NIDS) for IoT systems. Moreover, this paper adds to knowledge of the comprehensive reviews of state-of-the-art design of NIDS for the resource-constraint IoT using the MEC, the implementation strategies, and the IoT dataset used. The study extends the design approaches used by researchers and how the proposed methods fit into NIDS design for IoT systems and MEC environment. We also proposed an NIDS framework for the IoT utilizing MEC architecture and demonstrated the possible ways to choose NIDS for the IoT devices based on some conditions. After reviewing the related research publications in this field, this work serves the audience from the following perspectives:

- We examine some of the leading IoT challenges presented in the recent research trends. Moreover, we analyze the feasibility of emerging technologies such as MEC to design security applications for the IoT.
- We investigate the ML-based NIDS designed for IoT and their implementation strategies.
- We examine the placement strategies used in MEC to develop NIDS for the IoT systems.
- We review various datasets and metrics used to develop NIDS for IoT systems.
- We propose a security framework for the resource-constrained IoT systems by utilizing the modern MEC architecture.

1.3. Paper Outlines

We have organized the rest of this paper as follows: Section 2 provides a summarized explanation of the key terminologies and security technologies used in IoT NIDS design. We classify the IoT security threat classes in Section 2 and introduce the fundamental security routines with attention to NIDS diversities. Section 3 describes the background of NIDS in IoT systems. This section also elaborates on the various types or categories of NIDS. In Section 4, we discuss how MEC are used to create NIDS, its design process, deployment strategies, and implications toward the heterogeneous IoT systems. Section 5 examines the mode of implementation of NIDS in IoT, while Section 6 examines the NIDS design strategies. Then, we compare and evaluate the various implementations of ML-based NIDS, datasets, and metrics used in NIDS design for IoT systems in Section 7. Finally, the survey concludes with an outline and future directions in Section 8. Figure 1 shows the brief taxonomy of the survey. Table 1 also provides some of the commonly used acronyms in this survey.

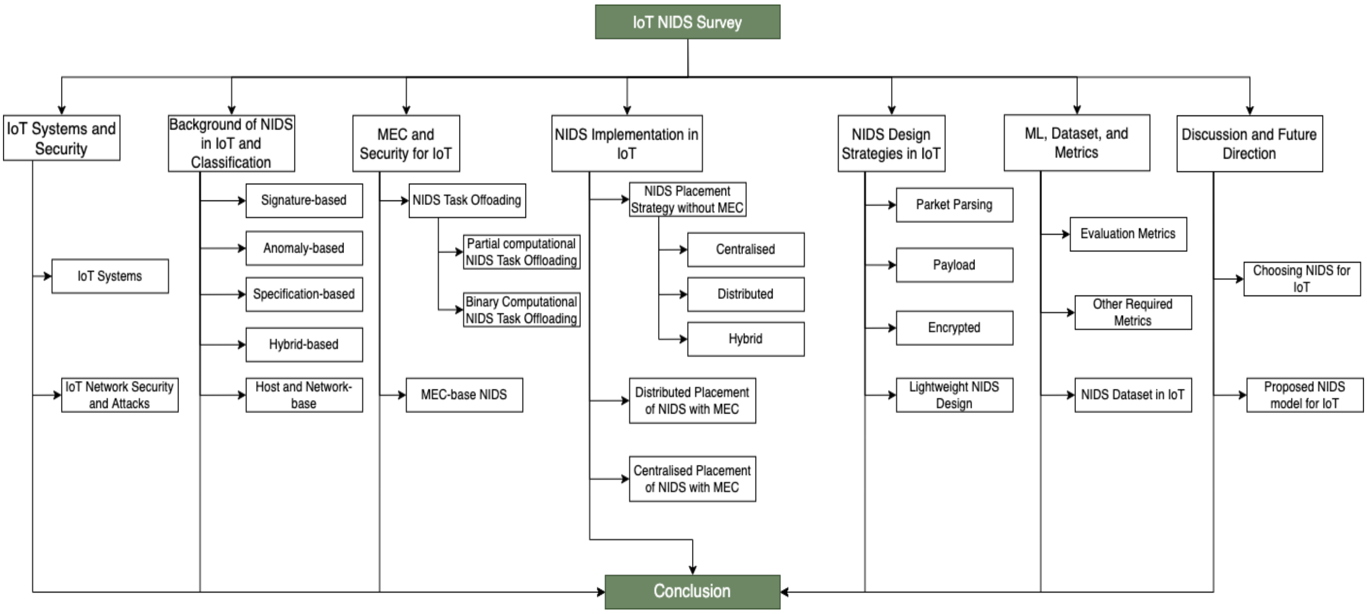
IoT NIDS Survey
IoT Systems and Security
Background of NIDS in IoT and Classification
MEC and Security for IoT
NIDS Implementation in IoT
NIDS Design Strategies in IoT
ML, Dataset, and Metrics
Discussion and Future Direction
IoT Systems
IoT Network Security and Attacks
Signature-based
Anomaly-based
Specification-based
Hybrid-based
Host and Network-base
NIDS Task Offoading
Partial computational NIDS Task Offloading
Binary Computational NIDS Task Offloading
MEC-base NIDS
NIDS Placement Strategy without MEC
Centralised
Distributed
Hybrid
Distributed Placement of NIDS with MEC
Centralised Placement of NIDS with MEC
Parket Parsing
Payload
Encrypted
Lightweight NIDS Design
Evaluation Metrics
Other Required Metrics
NIDS Dataset in IoT
Choosing NIDS for IoT
Proposed NIDS model for IoT
Conclusion

Figure 1. Taxonomy of Survey.

Table 1. Summary of Some Important Acronyms.

Acronym	Short Definition
IoT	Internet of Things
NIDS	Network Intrusion Detection System
ML	Machine Learning
NIDS	Intrusion Detection System
MEMS	Micro-electromechanical System
OT	Operational Technology
VHD	Virtual Honeypot Device
IEEE	Electrical and Electronics Engineers
6LoWPAN	IPv6 Low-Power Wireless Personal Area Network
LPWAN	Low Power Wide Area Network
DoS	Denial-of-Service
DDoS	Distributed Denial-of-Service
MITM	Man-In-The-Middle
WSN	Wireless Sensor Network
RPL	Routing Protocol for Low Power and Lossy Network

Table 1. *Cont.*

Acronym	Short Definition
NFV	Network Functions Virtualisation
ICN	Information-Centric Network
SDN	Software-Defined Network
MEC	Mobile Edge Computing
VM	Virtual Machine
RL	Reinforcement Learning
LA	Learning Automata
MLP	Multilayer Perceptron
R2L	Remote-to-Local
U2R	User-to-Route
CNN	Convolutional Neural Network
DIDS	Distributed Denial of Service
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
AIS	Artificial Immune System
PCA	Principal Component Analysis
QoS	Quality of Service

2. IoT System and Security Issues

The IoT system has evolved with many use-cases. Each of the applications of the IoT system comes with its own adverse security issues. This section elaborates on the IoT system and the related security problems.

2.1. IoT System

IoT systems consist of interconnected computing devices [17,18], mechanical or any object equipped with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction [19]. IoT systems use distinctive network address schemes to connect to various objects or things [20]. Most IoT devices connected to the Internet are exposed to cyber-attacks due to the weak security systems as a result of resource constraints. Most IoT systems however operate autonomously via untrusted network connections and the Internet, which is also a contributing factor that exposes the network to cyber-attacks [21]. Considering cyber threats and network attacks versus the promising future of IoT systems, the security issues should be addressed with urgency.

2.2. IoT-MEC Architecture

There is no one defined architecture that is generally accepted by the IoT community. Different architectures are designed based on the use case, the technology used, and the size of the IoT network. In smart health use cases, refs. [22–25] proposed a smart architecture to monitor and track patients' health records and transmit physiological parameters to control centers where the advance interpretation of the data is performed. Wang et al. [26] proposed a novel IoT access architecture based on field-programmable gate array (FPGA) and system on chip (SoC), which provides a unified approach to the IoT for a wide assortment of low-speed and high-speed devices with associated extendibility and configurability [27]. In technology and communication, Shang et al. [28] analyzed the security and challenges in implementing TCP/IP technology for IoT systems architectures. In [29–31], the authors also created different routing protocol for IoT communication. These architectures can be implemented in Autonomous Systems of Things (ASoT) and legacy Autonomous Systems (ASs). Different configuration techniques were applied for IoT routing architectures. Notwithstanding, different architectures have been implemented for assorted use scenarios such as in [32], where the authors proposed an IoT-based architecture for sport (football). Their aims were to embed sensing devices (e.g., sensors and RFID), telecommunication technologies (e.g., ZigBee) and cloud computing with the mission to monitor the health of footballers and eradicate the occurrence of diverse health problems in football. In all the above-mentioned architectures, they have different use-cases; therefore, the sensitivity of data in each scenario differs. Creating security for each system requires special knowledge in the field of application. However, the IoT-MEC architecture follows the level such as Sensor Level (IoT end-device), MEC/Fog Layer, and the Cloud. Detailed the structure of the IoT devices, MEC, and the cloud is shown in Section 3.

2.2.1. Sensor Level (IoT End-Device)

Mobile devices (e.g., smartphones, tablets, and laptops) and IoT devices (e.g., industrial actuators, wearable devices, and smart sensors) make up the bottom of the IoT or sensor layer [33]. Based on their ownership, co-location, and use cases, these devices can be grouped into clusters.

2.2.2. MEC/Fog

The MEC is a layer that lies between mobile devices and the cloud [19]. The MEC system directly connects to the IoT gateway. The MEC is usually equipped with high-performance computational resources [34] to support the IoT devices connected at the network edge. In an IoT MEC-enabled environment, security is vital. Hence, the computational resources of the MEC provide an opportunity to design a sophisticated security system to protect the IoT system.

2.2.3. Cloud

The cloud layer is part of the core network, and it consists of various cloud servers and data centers that can process and store massive quantities of data [33].

2.3. IoT Network Security and Attacks

IoT applications and technologies are expected to step farther ahead than anyone could possibly imagine [35]. However, the development of IoT technologies is still in transition and has not fully matured in terms of security protection. The IoT systems have several security challenges [16], such as non-uniform manufacturing standards and update management problems caused by IoT software developers. Physical handling of the security issues and users' ignorance due to their lack of awareness of security problems associated with the IoT systems are also critical problems. Moreover, there is no standard defined security architecture accepted by the IoT community. Different security architectures are adopted based on the use case, system requirement, the available technology, and the IoT network size. For example, in the intelligent health use cases discussed in the research works in [22], the authors proposed different intelligent security architectures to monitor and track patients' health records. The sensitivity of data in IoT use cases varies. Hence, the need to create security for each scenario requires special knowledge in the field of application. It is therefore convincing that NIDS designed to target the various architecture and use case in IoT requires customization [36]. Apart from adopting encryption methods to safeguard the IoT system data transmission, the network and environment of the IoT systems must also be protected. Unfortunately, traditional network security systems are not applicable in IoT systems due to the nature of the resource constraints. Furthermore, different network attacks have also emerged due to the rapid development and wide application of the IoT system applications [16,37]. The amount of attacks will continue to rise as the IoT use cases expands. Being able to recognize and comprehend the intense rise in cyber-threats in the IoT system drastically decreases the risk of a network security and data breach. Detailed discussions of these attacks are found in [4,16]. Recently, some of the most frequent attacks initiated to target the IoT systems include:

- Spoofing: Attackers impersonate a legitimate IoT system in a network to gain control or illegal access to the network. When the attacker obtains access, they initiate DoS and Man-In-the-Middle attacks against targeted devices [38–40].
- Denial of Service (DoS): A cyber-attack makes IoT systems or resources on the network unavailable to the intended legitimate users. These attacks are launched by the perpetrators to temporarily or permanently disrupt a host IoT system's services [41–43].
- A distributed denial-of-service (DDoS): The attack is a malicious network attack that can disrupt the regular traffic and the network's services. DDoS floods the target or surrounding infrastructure with highly intense network traffic [44]. DDoS attacks become successful when attackers apply multiple compromised computing systems as the sources to generate a large amount of network traffic. The targets can be either the IoT system or other networked devices [45,46].
- Jamming: Most IoT devices communicate to other devices through wireless networks. The perpetrators attack the targeted IoT system and send a fake signal to interrupt the radio transmission, thereby depleting the bandwidth, processing power, and memories [47,48].
- Man-in-the-Middle: Figure 2 shows a diagram of Man-In-the-Middle (MITM) attacks. MITM attackers secretly relay and manipulate the communication between two IoT systems, the remote devices, and eavesdrop the private communications among the parties [49,50].

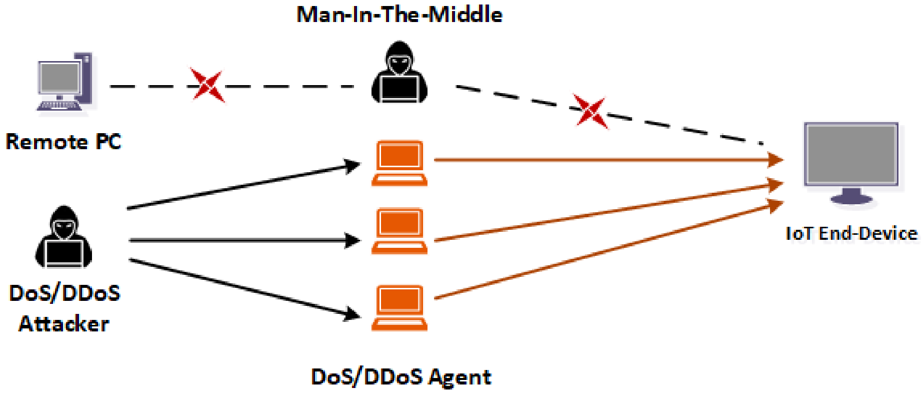

Figure 2. Network Attack.

- Privacy leakage: Data protection and privacy should be a priority in IoT system communication. Most IoT system manufacturers and users pay minimal attention to the contents of information stored on their devices and how third-party companies analyze and process such IoT privacy information. IoT systems such as wearable devices that collect user's information about their location, situation, and health information have contributed immensely to the high rise in the risk of personal privacy leakage [51,52].
- Marai Botnet Attack: Mirai is malware that turns networked devices into remotely controlled bots that cyber-attackers use as part of a botnet in a large-scale network. It primarily targets online consumer devices such as IP cameras and home routers. Attacks such as DoS/DDoS also used Mirai as a prevalent initiator [53,54].
- Sybil Attack: Sybil attacks are found in peer-to-peer networks. A Sybil attack subverts the IoT device's identity to create many anonymous identities and consume a disproportionate power. It is named after Sybil, the author of the book Sybil, which is a case study of a woman dealing with a dissociative identity disorder. An IoT device in a network that operates multiple identities often undermines the authorized network access in reputation systems [55,56]. Sybil attacks capitalize on this weakness in the IoT system network to initial attacks.

- AI-Based Attacks: AI attacks have been around since 2007 [57,58], and the high threats they pose to the IoT are becoming more prevalent. Hackers now create AI-powered tools that are quicker, simpler to scale, and more efficient than human-centric attacks. This is a significant danger to the IoT ecosystem. While the techniques and features of conventional IoT risks offered by cybercriminals may appear to be the same, the scale, automation, and customisation of AI-powered attacks will make them increasingly difficult to eradicate.

  Table 2 summarizes some common attacks in IoT systems and their modes of initiation.

Table 2. Common Attacks in IoT-MEC.

Source	Attacks	Mode of Attack Initiation
[38–40]	Spoofing	Impersonation
[41–43]	Denial of Service (DoS)	Network Flooding
[44–46]	Distributed Denial-of-Service (DDoS)	Network Flooding
[47,48]	Jamming	Fake Signaling
[49,50]	Man-In-the-Middle	Eavesdropping Packets
[51,52]	Privacy Leakage	Attack Authentication Storage
[53,54]	Marai Botnet Attack	Malware Implant on Devices
[55,56]	Sybil Attack	Creates Anonymous Identities
[57,58]	AI-Based Attacks	Creates AI-powered Tools

3. Background of NIDS in IoT and Their Classification

The IoT security problems are similar to the early genealogy of computers in the mid-1990s [59] when the unreliability of personal computers (PC) security required emergency attention. Many computer networks were vulnerable to threats, and there were no direct approaches to fix them. Organizations kept vulnerabilities as a mystery and did not discharge security overhauls rapidly. Moreover, it was hard, if not incomprehensible, to motivate clients to introduce new security systems. In recent years, this has changed because of a mix of full divulgence distributed vulnerabilities to organizations' constraints. The skills of attackers, the developing technologies, and the enormous growth of Internet traffic have made it difficult for any existing NIDS to offer a reliable service to the IoT system. However, a close examination of the IoT system's cyberattack shows that there usually exists a behavioral pattern in the attacks [60], which can be learned when ML and MEC are combined to design NIDS more effectively.

This section provides a comprehensive review of the various classifications of the state-of-the-art NIDS used in IoT systems. Since IoT networks operate similar to traditional general purpose computer networks, researchers have adopted the traditional techniques to design NIDS and propose numerous models. We based our classification in this section on the framework, implementation, and operation to categorized NIDS into Signature, Anomaly, Specification, and Hybrid based [61].

3.1. Signature-Based NIDS in IoT Systems

IoT devices have limited resources, such as the memory, the processing ability, and the energy constraints of the physical IoT devices [62]. Therefore, the use of traditional signature-based intrusion detection systems on these devices is practically impossible. To construct robust detection systems for IoT, signature-based NIDS often requires large datasets. However, conventional signature-based NIDS must be restructured in accordance with the resources of the IoT devices. Various attempts have been made to design signature-based NIDS for IoT devices. Kasinathan et al. [63] proposed a signature-based NIDS framework. Their framework identifies DoS attacks in 6LoWPAN-based [64] networks.

The authors used signature-based NIDS known as Suricata (an open-source NIDS software) [65] to test their proposed framework. Suricata was designed to detect intrusion in general computer networks, which does not specifically target IoT networks. Moreover, there were no clear evidence of the effects of Suricata on their IoT devices use in their research. In [66–69], the authors presented different strategies to design NIDS to protect the IoT devices. The authors used ML methods to design signature-based NIDS to detect network attacks in IoT devices. In [70,71], the authors applied deep reinforcement learning to create NIDS for industrial IoT. This approach combines the observational capabilities of deep learning with the decision-making capabilities of reinforcement learning to allow the effective detection of various types of cyber-attacks on the Industrial Internet of Things. However, in each of the research works, the experimental results presented are promising, but the authors failed to demonstrate how their method could work in a real-world IoT network environment.

3.2. Anomaly-Based NIDS for IoT Systems

Anomaly-based NIDS compare an IoT system's activities instantly against the standard behavioral profile and generate notifications whenever a deviation from such operation exceeds a threshold. This approach is efficient in detecting new attacks on IoT systems. In particular, those attacks initiated misuse of the IoT devices' resources. Unlike signature-based NIDS, anomaly-based NIDS are designed with either normal or anomaly data (one-class of data) [72]. Several NIDS developed to secure IoT devices use anomaly-based techniques since they can be modeled with the intention of being lightweight. In [73], the authors highlighted the necessity to obtain custom anomaly-based security in an IoT system. They also emphasized that vulnerabilities in IoT systems are due to insecure web interfaces, insufficient authentication and authorization, insufficient transport layer protection, broken cryptography, insecure software/firmware updates, or poor physical security. Their research presented an efficient hierarchical anomaly-based intrusion detection system. In their experiments, the proposed framework applied a resilient policy that enables the IoT system to detect malicious attacks. In [74–77], the authors also used ANN to detect intrusion in the IoT system's gateway. They used ANN to detect anomalies in the data sent from the edge devices. The researchers connected multiple IoT devices and a high-resourced network device to operate as the gateway. Their results were immaculate and promising. In [78–80], the authors presented an NIDS algorithm that detects attacks in IoT systems, which uses anomaly detection systems based on different ML methods. The authors explained that an IoT network attack typically leaves its traces on the system. Using this approach, the authors proposed three techniques to identify such anomalies in their network. However, they did not demonstrate any experimental results of false-positive rates, which is a major problem with anomaly-based NIDS. The authors also performed a further study on the power and memory consumption of IoT devices. The authors of [81,82] proposed a smart home NIDS that alters the decision function of its underlying anomaly categorization models autonomously in response to changing conditions in the smart home.

The research works in this subsection demonstrate that anomaly-based NIDS often start by establishing a baseline of the network's normal traffic and activity. They compare the current condition of network traffic to this baseline in order to find patterns that deviate from the normal traffic. Anomaly-based NIDS are suitable for IoT systems due to the ability to scale them into a lightweight but are prone to false-negative rates. The false-alarm rate is due to two main reasons: (1) researchers failed to obtain a credible training dataset that holds all the necessary intrusion signatures; (2) not all normal or intrusions behaviors were captured during training.

3.3. Specification-Based NIDS for IoT Systems

The Specification-Based NIDS defines a set of rules that can be manually updated by the IoT systems administrator to detect intrusion on a network. The specification-based approach identifies threats by the definition of high-level rules in the IoT network environment, operational features, and mechanisms. Due to the resource constraint problems in IoT systems, specification-based NIDS is recommended. However, specification-based NIDS detects only specific kinds of attacks that fall within the rules defined in IoT systems. The critical difference between specification-based and anomaly-based NIDS is their manually defined rules for each attack [83]. The authors argued in [84,85] that specification-based NIDS can identify attacks that disturb the optimal protocol structure of IoT networks. The authors presented an RPL specification to be used as a reference to evaluate the IoT node's behaviors, which was created using a semi-auto profiling approach that creates a high-level abstract of activities using network simulation traces. This specification, which includes all valid protocol states and transitions as well as statistics, is implemented as a collection of rules in the NIDS agents, which are transmitted in the form of cluster heads to monitor the whole network. Instead of allowing the cluster head (usually an MEC server) to monitor everything, the members of the cluster report relevant information about themselves and their neighbors to the cluster head to conserve resources. Moreover, the authors in [86–88] applied a specification-based method to detect attacks in the IoT network. All the research works in this subsection demonstrates that specific attacks such as those that target RPL protocols in IoT devices can efficiently be detected by applying specification-based NIDS.

3.4. Hybrid-Based NIDS for IoT Systems

The hybrid strategies use two or more signature-based, anomaly-based, and specification-based NIDS to create one system. Hybrid-based NIDS for IoT systems capitalize on the strength of the combined methods and minimize the impact of their setbacks. To eradicate false alarm rate flops in anomaly-based NIDS and keep the advantages of signature and specification-based detection, some researchers proposed hybrid-based NIDS for IoT systems. Based on this combined approach, we categorize the hybrid NIDS into two categories:

- Sequence-based: either anomaly or misuse detection is applied first, and a different technique is applied next.
- Parallel-based: multiple detectors are concurrently applied, and the final decision is made based on multiple output sources.

The most common type of hybrid NIDS combines signature-based detection and anomaly detection. In such a hybrid system, the signature-base detection technique detects known attacks, and the anomaly detection technique detects novel or unknown attacks. Due to the resource constraint nature of the IoT systems, applying hybrid-based NIDS in the same IoT systems is practically difficult and not recommended. However, a lot of research works demonstrate the feasibility of hybrid-based NIDS on IoT systems.

According to Ning and Lui [89], human activities also cause security problems through the architecture differences and the standards used in IoT design. The authors introduced a hybrid security framework called IPM (that deals with information, physical, and management security perspectives). Cervantes et al. [90] combined Watchdog, Reputation, and the Trust Strategy to detect intrusion in IoT systems. Their approach detected the sinkhole attack. INTI (Intrusion detection of Sinkhole attacks on 6LoWPAN for Internet of Things) investigates NIDS's effects that cause problems with the performance. According to their study, the sinkhole has advanced effects on networks such as mobile ad hoc networks (MANETs), wireless sensor networks (WSNs), and vehicular ad hoc networks (VANETs). They emphasized that sinkholes cause a high positive rate, false negatives, high energy consumption rates and slow down the IoT system's performance. Their system analyzed each IoT device's behavior in the network and checked for the existence of sinkholes. They also identified attacks on IoT devices through inter and intra-cluster communication channels. Constrained Application Protocol (CoAP), designed by Krimmling and Peter [91], used hybrid NIDS to combat intrusion in IoT. Their system was tested with the OMNET++ simulator. The authors also evaluated their framework with an intelligent transport application system that uses CoAP. The heterogeneous nature of the communication network poses a critical challenge to IoT systems. Therefore, they employed signature-based and anomaly-based NIDS in their research. They detected known attacks with signature-based NIDS, while the anomaly detected unknown attacks that they introduced during their experiment.

The application of hybrid-based NIDS in IoT systems solves the problems associated with a single NIDS. However, hybrid-based NIDS require more computational resources, storage capacity, and energy, which makes them an imperfect fit for the IoT system. Recently, researchers have utilized MEC computing to resolve the resource constraint problems associated with the IoT system. Researchers apply MEC technology to design NIDS for the IoT to offload the most resource-intensive part of the NIDS to a proximal MEC server [92,93]. A detailed discussion of this approach is provided in Section 4 of this paper.

3.5. Host and Network-Based NIDS for IoT Systems

NIDS design for IoT can also be classified based on their mode of operation. There are two main mode of NIDS operation: namely, Host-based and Network-based.

Host-based NIDS identifies the threats and attacks on the same system or devices in which they have been installed. Host-based NIDSs use system data from normal routine processes, the same subnet as the monitored machines, and system call on a particular device [94]. Wang et al. [95] designed a host-based NIDS that detects cyber attacks in IoT. The authors experimentally demonstrated the existence of the mimicry attacks and demonstrated the need to prevent them. In [96], the authors propose a host-based intrusion detection system that was created and prototyped to safeguard IoT devices, that make up IoT network backbones. In their approach, a set of suggested NIDS performs traditional security verification, and it engages with a controller from the host-based NIDS to enable the coordination of intrusion detection activities in response to IoT devices affected by DDoS attacks throughout the network. The of authors [97] argued that network-based NIDS required a large amount of data to make meaningful decisions. Table 3 shows some differences between Host-based and Network-based NIDS.

Table 3. Different between Host-Based and Network-Based IDS.

	Host-Based NIDS	**Network-Based IDS**
Data Source	System call logs	Captured network traffic
Placement Strategy	Locally on the hosted device or machine	Specific IoT devices on the same subnet
Detection Rate	Low, difficult to detect new attacks	High, can detect new attacks in real time
Threats Trace-ability	Based on system calls	Trace intrusion based on network addresses and timestamps
Limitations	Cannot analyse network attacks, rules created can be obsolete, depends on the operating system (OS)	Monitor only network traffic within a specific subnet

4. Multi-Access Edge Computing as a Resource to Provide Security for IoT

The interest in applying MEC over cloud computing to combat network attacks in IoT systems has risen dramatically recently. Hence, MEC standardization has become of pivotal interest to major telecommunication and network stakeholders under the supervision of institutions such as the European Telecommunications Standards Institute (ETSI) and Open Edge Computing Initiative (OEC [98]) [7,99,100]. Moreover, it is known that NIDS designed for IoT systems use case scenarios based on MEC require high quality of service, reduced latency, high throughput, and real-time operation. Researchers desist from the use of cloud computing due to its crucial drawback of low propagation delay (high latency) [101,102]. The most influential features that make IoT system developers select MEC over cloud computing for NIDS design include:

1. Providing security context-awareness (the capability of the MEC server to disseminate real-time security information to the IoT device) [103];

2. Energy-saving during and after data transfer;
3. Improvement of privacy/security in IoT (users are deprived of the total ownership of their data in cloud computing, which results in private data leakage and loss) [104];
4. Optimal resource allocation by the MEC for the IoT system [105].

Figure 3 shows the structural layout of a cloud or remote server, the MEC, and the IoT.

MEC-based NIDS for the IoT system is implemented based on virtualized platforms that use modern technology improvement in network functions virtualization (NFV) [106], information-centric networks (ICN) [107], and software-defined networks (SDN) [108–110]. Specifically, NFV enhances an edge device to supply computing services to numerous connected IoT systems to create multiple virtual machines (VMs) to execute their NIDS tasks spontaneously or operating different network functions [111]. The IoT systems offload computationally intensive tasks to the MEC platform for execution. The security task offloading takes place in different forms. The next sections examine the various possible offloading techniques for NIDS implementation in IoT systems.

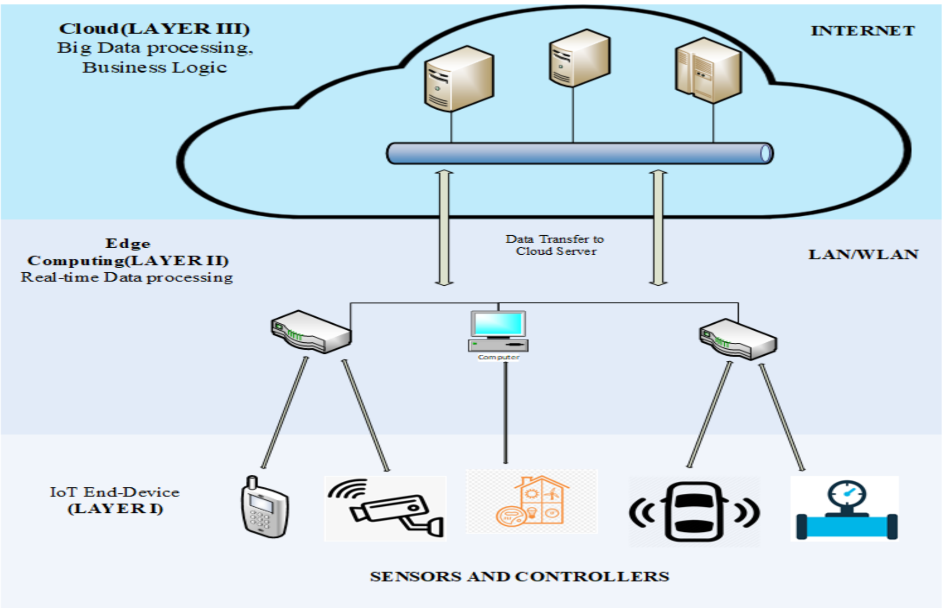

Figure 3. MEC Architecture.

4.1. NIDS Task Offloading in IoT-MEC Environment

This section elaborates the mechanism of slicing the NIDS (tasks) and transferring part or all to the MEC platform. Task offloading is the primary technology used in MEC and Fog for such kind of NIDS design to secure the IoT system from cyber attacks. Task offloading is employed as a mechanism to reduce energy consumption, latency, and the resource-constraints problems in IoT system [112–114]. Hence, the proximity of the IoT system from the MEC server during NIDS task offloading is significant. It determines the intensity of tradeoffs (bandwidth, amount of energy, scheduling algorithms, etc.) required for a complete execution cycle of the NIDS in IoT systems [115,116]. NIDS tasks offloading can be categorized as follows: Binary NIDS Tasks Offloading; Partial Computational NIDS Tasks Offloading.

4.1.1. Partial Computational NIDS Task Offloading

Partial Computational Offloading dividends the underlying NIDS into sub-tasks. The IoT system offloads sections of the sub-tasks to a proximal edge computational platform for execution, while less computational tasks are executed on the IoT system. Wang et al. [117] proposed a distributed framework in their NIDS design for IoT systems. Their approach uses the partial computational offloading technique to distribute the NIDS on the IoT system and MEC platform. They also believed that the NIDS sub-task offloading to the MEC platform reduced workload and energy consumption. In [80,118,119], the authors proposed an internal distributed NIDS for IoT. They offloaded sections of the NIDS system across the various IoT systems. Each IoT device in the network monitors its internal network but provides computational support to other devices whenever there is an intrusion. Their analysis showed that the above approach has higher accuracy of detection compared to a standalone NIDS. Figure 3 shows the structure of an *MEC* design. Different partial computational offloading algorithms have been designed for IoT systems in a 5 G heterogeneous network. A review of these research works shows that the algorithms implemented are directed toward minimization of energy consumption and low latency [120–122]. However, the researchers pay less attention to the use of MEC offloading to secure the IoT. In [123], the authors implemented a multi-user partial computation offloading. To this aim, Alladi et al. [123] described a Deep Learning Engines (DLEs)-based artificial intelligence (AI)-based intrusion detection architecture to identify and classify vehicle traffic within internet of vehicles (IoV) networks into probable cyber-attacks. These DLEs were also implemented on MEC servers rather than remote cloud, which takes into account the mobility of the vehicles and the real-time requirements of the IoV networks. The following conditions must be considered to apply partial computational offloading to design NIDS for IoT systems:

- First, the sequence of execution functions or routines cannot be discretionarily selected because the outputs of some components of the NIDS model are the inputs of others.
- Second, due to resource constraints and security considerations, some algorithms or routines of the NIDS must be offloaded to the MEC for execution, while others have to be executed locally.

4.1.2. Binary NIDS Task Offloading

The binary NIDS task offloading allows the IoT device to execute the entire NIDS either on the local IoT device or remotely in the edge platform [124]. An integrated NIDS cannot be divided into sub-tasks. Hence, those kinds of NIDS are executed fully in the edge platform or locally in the IoT system. In [125], the authors proposed a Markov decision process as a means to enhance tasks offloading to the edge. They first proposed a double deep Q-network (DQN)-based strategic computational offloading algorithm to study the optimal policies before they implemented their original technique.

4.2. MEC-Based NIDS Design for IoT Systems

In NIDS design, the MEC provides the hosting infrastructure to support the computational intensive security services that manage the real-time detection of threats within and outside the IoTs' network. Table 4 summarizes the various research works that used MEC technology to design NIDSs for IoT systems. In this section, we review the use of MEC platforms to design NIDS for IoT systems and discuss their strengths and weaknesses. Since there are limited MEC-based NIDS for IoT systems, we provide a comprehensive survey of the state-of-the-art related designs and their application.

In [105,126], the authors provided closely related findings on MEC-based NIDS for IoT systems. Their research provided a security end-to-end protocol for resource-constrained devices, especially in the context of healthcare sensors. The authors also applied security functionality and reduced computationally intensive operations to satisfy the IoT system's resources constraints problem. The authors offloaded the computationally intensive algorithm of the NIDS to an edge platform in neighboring trusted devices. They introduce a technique to select the proximal MEC platform with unique selection criteria. Ranaweera et al. [127] highlighted some security flaws in 5 G-based IoT use cases that have been implemented in the MEC setting. They suggested solutions to mitigate the security flaws that they identified. In [128,129], the authors proposed security as a service (SECaaS) architecture that protects the IoT from cyber-attacks. They test their suggested architecture by creating a virtualized infrastructure that combines lightweight and hypervisor-based virtualization technologies to design a security system for the IoT using MEC. Sha et al. [130] proposed a new security system deployed at the MEC platform to secure connected IoT systems. Their security system, which was known as EdgeSec, comprised six major modules that systematically handle specific security challenges in IoT systems. The modules include a security profile manager, security analysis, protocol mapping, a security simulation, communication interface management, and request handling. In their experiment, each IoT device is registered to the security profile to manage the device-specific information collected and the device-specific security requirements. An NIDS deployed to an MEC platform controls the security of the various IoT systems. They also utilized protocol mapping to choose the appropriate security protocols from the protocol library for each IoT device in their setup. In [131], the authors proposed a privacy-aware scheduling algorithm based on MEC. The research's primary purpose is to execute tasks from different IoT systems with different privacy requirements connected to different MEC platforms. The proposed scheduling algorithm also aims to satisfy the real-time requirements of IoT system operation. The authors of [105] in their paper examine ways to improve the security of MEC systems by utilizing the cooperative mechanism amongst non-orthogonal multiple access (NOMA) user pairs. They presented a two-slot hybrid cooperative NOMA security (THCNS) system that uses cooperative interference between NOMA user pairs to improve the security of offloading, taking into account the varying latency needs of IoT users.

The research works in this section demonstrate that the use of MEC to create NIDS for the resource-constraint IoT devices is still in the early stage. Hence, researchers must provide robust frameworks for the offloading process, detection models, and implementation strategies to design sophisticated NIDS to secure the IoT system and their network environments.

5. Mode of NIDS Implementation in IoT Systems

Additionally, we categorize the NIDS based on their mode of implementation. This category of NIDS depends on the location, position, and part of the network where it is hosted. This subsection discusses the NIDS placement strategies in IoT systems.

5.1. IoT NIDS Based on Placement Strategies without MEC

There are different techniques to implement NIDS in IoT systems. Most of the implementation strategies depend on the structure and the type of network technology available. Although different IoT systems have different network architecture, however, we classified the placement strategy under the following:

- Physical domain: interfaces the sensors to the IoT board.
- Network domain: gives the IoT device the capabilities to transfer data.
- Application domain: gives the user power to interact with the IoT device.

Usually, NIDSs are placed in network equipment such as routers, installed in between the IoT device network and the Internet connectivity to the Internet Service Provider (ISP). These routers are known as border routers [132]. The following subsections explain some of the NIDSs' placement strategies in IoT systems.

5.1.1. Centralized Placement

In the centralized placement, NIDS analyzes all the traffic that goes in and out of the boarder router from the connected IoT devices. The centralized placement strategy has the disadvantage of not identifying attacks within the internal IoT networks. In [63], the authors used the centralized placement method to deploy their proposed NIDS. Their research focused on prevention of the DoS attack against the IoT system. Therefore, they used a dedicated host that sniffs network data and analyzed it. They connected the host NIDS through a wired network connection, leaving the remaining IoT devices connections in a wireless mode. The technique helps the NIDS to identify the DoS attack when the network is compromised. Wallgren et al. [133] designed a centralized NIDS placement system by setting up their NIDS in a border router. Instead of analyzing the network traffic, echo messages were sent to the various IoTs regularly to determine whether the network has been under attack.

5.1.2. Distributed Placement

In the distributed placement strategy, the NIDSs are spread across the various IoT devices in the network. Due to the IoT system's resource-constrained nature, each system must be optimized independently, and the NIDS must be lightweight.

IoT devices that are configured to keep track of threats in nearby connected devices are known as watchdogs. Cernantes et al. [90] used this placement strategy in their work. The INTI used the watchdog to detect and mitigate attacks. In their experiment, an IoT device within the network clusters was classified as the leader or the head node. Each connected IoT device's role can change over time due to the network reconfiguration policy they introduced. Whenever an IoT device detects an attack, it broadcasts a message to the other connected devices to protect themselves from the attacker.

5.1.3. Hybrid Placement

The hybrid NIDS placement integrates the technique of centralized and distributed placement strategy. It dwells on the advantages of both methods and avoids their pitfalls. In Pongle and Chavan [134], the authors stated that IoT devices are responsible for detecting changes in their neighborhood and sending information about neighbours to centralized modules.

Their experimental results showed that the energy, computational, and memory usage overhead matched the constrained IoT system's resources. In [118], the authors proposed an NIDS that allocates different responsibilities to the border router and the connected IoT systems. The NIDS module, in their experiment, monitored the neighbors to identify possible intrusions. When an intrusion is detected, the IoT sends an alert to the NIDS model placed within the border router. In [135,136], the authors used a distributed placement strategy to implement their proposed security system for the resource-constraint IoT system. Thanigaivelan et al. [118] classified their system as a distributed NIDS. However, the border router's central role to make the final decision about intrusion detection makes the proposed NIDS a hybrid approach.

5.2. Distributed Placement Strategies of NIDS in IoT with MEC

Distributed NIDS (DIDS) consists of two or more NIDS places over a wide span of a network(s), which interact with each other. Some distributed systems have a centralized NIDS that supervises the other security systems placed in other IoT systems in the same network [137]. DIDS operates by the process of cooperative intelligent systems spread across the network in a distributed IoT MEC environment. Due to the resource constraints in IoT, distributed data-flow programming models are used to build IoT applications utilizing the MEC platform [138,139].

In [140], the authors proposed a novel NIDS architecture model for the IoT system. The model operates based on the MapReduce approach in the context of distributed detection. The model also incorporated a multi-faceted detection technique based on anomaly-based and misuse-based NIDS agents. Farhoud et al. [141] designed a distributed and lightweight NIDS based on an Artificial Immune System (AIS). They distributed their AIS across the cloud, the edge, and the IoT devices. In [142], the authors developed a distributed NIDS that detected attacks, where part of the detection model is placed on the IoT device and hosted the remaining model on the resources of the Internet Service Provider (ISP). They designed the NIDS to secure home gateway devices that connect smart home IoT devices to the Internet. Zeeshan et al. [143] designed and evaluated an NIDS system suitable for small IoT devices. They used a trust management process that allows IoT devices to handle vital information about connected neighbors. This mechanism allows the IoT device to identify malicious patterns in the network. [144] proposed an NIDS that utilizes information flow processing to obtain event data from distributed sources as soon as relevant data arrive. Their system also was equipped with capabilities to detect attacks in real time. In [145], a collaborative NIDS was proposed by distributing the NIDS algorithm across the various IoT devices to detect intrusions. In this way, the IoT devices share the cost of running the NIDS, which reduced the energy, processing power, and storage capacity required for the detection.

5.3. Centralized NIDS Placement Strategies Based on IoT with MEC

In the centralized strategy based on the MEC platform, the NIDS is placed in the central part (proximal MEC platform). Usually, most network system administrators adopt centralized placement strategies and install NIDS in the border router. However, analyzing the network traffic that crosses the border router is insufficient to detect attacks that involve IoT devices within the network. Then, researchers must create NIDS that can monitor the traffic transferred between IoT devices. Cho et al. [146] proposed NIDS for IoT systems to analyze the packets that traverse through the border router (proximal MEC platform) to connected physical IoT devices and other connected network devices. The work focused on botnet attacks that pass through the border router. Wallgren et al. [133] proposed a centralized NIDS IDS that was placed in the border router. Their main aim was to detect network attacks within the IoT network. The authors proposed a heartbeat protocol placed in the border router that transmits ICMPv6 echo requests to all the IoT devices in the network to detect attacks. Jun et al. [147] also proposed Complex Event-Processing (CEP) techniques for network intrusion detection in IoT systems. The authors adopted a centralized approach and placed the NIDS in the border router to monitor network traffics. The main advantage of the proposed system is the use of features of the events flows to detect the intrusions, which can reduce the false alarm rate. Table 4 summarizes some of the NIDS designed with MEC by the research community.

Table 4. Analysis of NIDS in IoT systems using MEC and Placement Strategies.

Reference	Detection Methods	Threats Identified	Placement Strategies
Thanigaivelan et al. [118]	Network fingerprinting	Network Anomalies	Distributed locally in the IoT Devices
Ferdowsi and W. Saad [119]	Artificial Neural Network (ANN)	Data Intrusion	Distributed locally in the IoT Devices
R. Chen, C. M. Liu, and C. Chen [148]	Artificial Immune-Base (ANN)	Network Intrusion	Distributed locally in the IoT Devices
Wang et al. [117]	ZeroR, KNN, SVM, NaiveBayes, Neural Network	Network Intrusion	Distributed with MEC
Hosseinpour et al. [141]	Negative selection algorithm	Network Intrusion (DoS/DDoS)	Distributed with cloud, Fog, and MEC
D. A. Abeshu and C. Naveen [149]	Stacked Autoencoder Deep Learning	Network Intrusion Attacks	Distributed with MEC
A. Abeshu and C. Naveen [150]	Multi-Layer Deep Network	Network Intrusion Attacks	Distributed Locally and Fog
Diro Abebe and Chilamkurti Naveen [151]	LSTM Deep Learning	Network Intrusion Attacks	Distributed Locally and Fog
A. S. Sohal et al. [152]	Markov Model and Virtual Honeypot Device	Network Intrusion Attacks	Distributed with Fog to MEC
J. Stolfo et al. [153]	Offensive Decoy technology	Cloud Data attacks	Distributed with cloud and Fog
B. K. Sudqi et al. [154]	Multilayer Perceptron (MLP)	Network Intrusion Attacks	Centralized on Fog

Table 4. *Cont.*

Reference	Detection Methods	Threats Identified	Placement Strategies
Sheikhan Mansour and Bostani Hamid [140]	MapReduce approach	anomaly-based and misuse-based attacks	Distributed locally on IoT
Gajewski M et al. [142]	Co-responsible distributed NIDS	Network Intrusion Attacks	Distributed between IoT and ISP's Server
Xingshuo et al. [155]	Sample extreme learning Machine	Network Intrusion Attacks	Distributed between IoT, MEC, and Fog

6. NIDS Design Strategies for IoT Systems

NIDS design strategies can be categorized into Packet Parsing-based, Payload Analysis based, and encrypted traffic analysis [156]. NIDS examines the payload information of transmitted packets in the IoT network. However, this strategy sometimes fails because the payload information is inaccessible in encrypted transmission. To solve the limitations, researchers apply encryption traffic analysis. Due to the resource constraint problems in IoT, most NIDS designed to protect IoT systems adopt lightweight design principles. This section examines the various design strategies and the technique of reducing the NIDS into a lightweight to match the resources in the IoT system.

6.1. Packet Parsing-Based NIDS Design for IoT

The network packets form the fundamental structure of network communication and are the data that are used for NIDS. They contain different forms of binary data that need first to be parsed [157]. Data transmitted through the network contain the header and the application data. The header consists of IP addresses, ports, and other essential fields specific to various protocols. The main advantages of using packets parsing to design NIDS data sources include:

- Packets consist of communication contents and can effectively be employed to detect U2L and R2L attacks.
- Packets contain TCP/IP data content and timestamps that precisely provide the NIDS information about attack sources.
- Packets analysis promote real-time data processing without content caching within the IoTs' network.

Meanwhile, each of the packets do not indicate the complete communication state or every packet's contextual information. Hence, it is difficult to detect some attacks, such as DDoS [158]. The packet analysis detection methods support packets that mainly include packet parsing methods and payload analysis methods.

Packet parsing-based detection utilizes different sorts of protocols in network communications such as HTTP and DNS. These protocols operate in different forms; the packet parsing-based detection methods primarily concentrate on the protocol header section. The standard practice is to extract the header fields using parsing tools (i.e., Wireshark, Scapy, or the Bro) and convert the most vital fields' values into feature vectors. The header section also extracts necessary features from packets using classification algorithms to design attack detection systems. In [159,160], the authors proposed models to captured packets from a trustworthy enterprise network and parsed them with different tools. In packet parsing-based NIDS design, the packets are grouped into protocol types. Different clusters are then formed based on the information with the ML cluster algorithm using the protocol datasets. Thus, the first dataset contains many clusters, where the information from any given cluster is homologous. Finally, a new classified dataset is created for the NIDS model with an ML algorithm to detect IoT system attacks.

6.2. Payload Analysis-Based NIDS Design for IoT

Payload analysis-based detection is also another approach to design data-centric NIDS. The payload of the application layer protocols forms the data section of the packet sent through the IoT network. The payload analysis-based methods work well in a multi-protocol network environment because the packet headers do not need to be analyzed [161,162]. As a type of unstructured data, payloads can be processed directly by deep learning models [163]. However, this method does not include encrypted payloads.

6.3. Encrypted Traffic Analysis-Based NIDS Design for IoT

The accessibility of dataset features in encryption data is hardly limited. As a result, it is tempting to concentrate on assaults that are ostensibly apparent due to increased traffic. Scanning, brute force, and DoS/DDoS attacks are all examples of these types of attacks. Many research studies have demonstrated that such attacks may be seen in network traffic [164,165]. Other sorts of attacks on encrypted traffic that do not involve decryption, on the other hand, have received far less attention. The attack must meet a few characteristics in order to be apparent without decryption. To begin with, attacks can only be detected if traffic passes through the NIDS. Some Nattacks can be carried out locally or via a different path to the target. In certain instances, NIDS may not be able to detect the direct attack, but it may be able to detect the indirect attack. Second, the attack must alter some aspects of network flow. Without depending on existing attack signatures, zero-day and targeted cyber-attacks may be detected by concentrating on network traffic flow and modeling different detectable attack aspects [156].

6.4. Lightweight NIDS Design in IoT Systems

Due to the aforementioned resource constraints in IoT systems, the idea of lightweight NIDS is coming into the limelight in the research domain to conform to the IoT devices architecture. Based on Tiburski et al. [166] research, lightweight NIDS does not mean simplicity, but it should be able to deliver its mandated security duties without compromising the resource constraints. Sedjelmaci et al. [167] reflected on a lightweight NIDS in their research as a security system that is more resilient and robust but modest in size to enhance the ease of implementation on the IoT devices in a network. In [168], the authors also defined a lightweight NIDS as having the ability to save energy and to require less computational resources. Therefore, we define NIDS as lightweight when it can fully perform the required security operations regardless of the IoT device's resource constraints. One way of achieving lightweight NIDS is to use feature selection and reduction models. Feature reduction such as PCA, auto-encoder, Pearson correlation, etc. [169] are used to reduce the dimensionality of datasets, which affect the model size, computation, storage capacity, and complexity of the NIDS model on the IoT device. Xingshuo et al. [155] proposed a lightweight ML algorithm called sample selected extreme learning machine (SS-ELM). They assumed that the IoT device could not host a large size of data from their analysis. Therefore, the data selection occurs on the server remotely before being submitted to the MEC for training and detection. To create lightweight NIDS for IoT systems, researchers use one or a combination of the following principles:

- Applying new protocols to decrease the number of computational operations required by a traditional NIDS;
- Utilizing optimization techniques to design a security system such that the scalar multiplication, addition, and doubling are reduced.
- Offloading of all or parts of the computationally intensive algorithms to a proximal device (MEC platform) in the same network that has more resources than the IoT device;
- Using modern algorithms that require less computational resources instead of the use of classical, alternative methods;
- Utilizing data feature dimensionality reduction. Different algorithms such as sparse auto-encoder, the Ranker method,principal component analysis (PCA), etc., are used

by researchers to reduce the dimension of the dataset to lightweight ML-based NIDS for IoT devices.

All the above methods have effects on the performance of the NIDS. Therefore, extra care is required to minimize the loopholes for cyber-attackers to mount attacks against the IoT device.

7. Machine Learning, Dataset, and Metrics for IoT Systems NIDS

Regardless of the number of NIDS development years, current NIDSs face challenges in enhancing detection accuracy, reducing the false alarm rate, and detecting newly created attacks. To eradicate the above problems, modern researchers focused on developing NIDS that utilize ML methods. ML algorithms automatically identify the vital differences in abnormal data existing within standard data with high accuracy. Moreover, ML algorithms have strong generalization potentials; therefore, they can detect unknown attacks [170]. This section of the survey focuses on some of the standard ML algorithms that are frequently used in NIDS, the metrics, and benchmark datasets. ML-based NIDS is designed to monitor the host and its environment, analyze the systems' behavior, generate alerts, and respond to all suspected attacks [171]. We consider ML methods under two main headings: supervised and unsupervised learning methods. There are other schools of thought in which there are more modern types classified as semi-supervised learning methods.

Supervised learning methods require classified and labeled data. In contrast, unsupervised learning extracts valuable information from an unlabeled dataset. To implement ML for any activity, three main important facets—time, space, and output—must be considered.

The development of anomaly detection systems using ML tools creates a generalization platform to generate futuristic normal or anomalous data to develop detection models. This generalization approach consists of generative and discriminative methods. Generative approaches construct ML-based models based on standard training data samples and perform several tests to determine how well they fit such a model [172]. Discriminative methods learn the difference between normal or anomalous data instances to create a model based on the results [173]. NIDS implemented in IoTs can be categorized based on the type of traffic and network structure. The detection is classified based on packet analysis, packet parsing,and payload analysis. Table 5 provides a summary of some of the machine learning-based NIDS and their metrics. In [174], the authors used recurrent neural network (RNN) deep learning to detect the intrusion. Their research focused on the analysis of ARM-Based IoT devices against malware with 98 % efficiency using 10-fold cross-validation consisting of two-layer neurons. As described in Saeed et al. [175], the energy-efficiency IoT devices operating with a batteries source cannot hold highly computationally intensive NIDS algorithms. Hence, choosing the right ML to create an intelligent security model for IoT systems is critical. The researcher must include critical evaluation processes such as the resources, power consumption, and performance of the IoT system. There are numerous ML-based NIDS for IoT in the research domain with different techniques. Azmoodeh et al. [176] used a deep Eigenspace learning approach to detect malware in IoT devices. Their proposed NIDS identifies malware that is located in the device's Operational Code (OpCode) sequence. They generated the dataset for this experiment, making it difficult to benchmark their results with existing systems. In [177,178], the authors designed an intrusion detection system using a one-class support vector machine (SVM). Their one-class SVM is effective in providing good analysis with well-behaved feature vectors. However, the authors used deep belief networks (DBNs) for robust feature engineering. They show comparable anomaly detection while reducing the training and testing time. La et al. [179] modeled a Bayesian game theory-based NIDS to detect intrusion in the IoT system. From their analysis, attackers try to deceive security systems deployed in a network to initiate successful attacks against connected IoT devices in a network. Therefore, NIDS systems can use the honeypots technique to identify such intrusions. In [180,181], the authors developed an intrusion detection system that identifies an anomaly in IoT systems. The goal was to make an NIDS adaptive to its deployed environment and to detect intrusion based on the behavior of the attack. In [182], the authors used K-NN and Naive Bayes to identify intrusion. They tested their model using the NSL-KDD dataset. In [183,184], the authors proposed an NIDS model for detecting intrusion (BOTNET) for IoT devices using logistic regression. ML methods such as Decision Tree [185], K-Mean [186], DNN (using the NSL-KDD dataset) [187], and CNN (using the NGIDS-DS and ADFA-LD dataset for benchmark) [188] have been used by researchers to design NIDS for IoT systems. RNN, LSTM, GRU, and GAN [151,174] are also one of the most common neural network ML-based methods used by the research community to create NIDS for IoT. All the above models provide promising results but lack the analysis of the effects they pose on resources of the IoT devices.

Table 5. Analysis of ML-Based NIDS in IoT.

Reference	ML Methods	Precision	Recall	Accuracy
Mayhew et al. [189]	Behavior-Based Access Control	99.1%	-	-
Haddadpajouh et al. [174]	RNN, LSTM	-	-	98.18%
Saeed et al [175]	RNN	-	-	97.23%
Azmoodeh et. al. [176]	Deep Eigenspace learning	98.59%	98.37%	99.68%
Erfani et al [177]	One-Class SVM	-	-	-
La et al. [179]	Bayesian Game Theory	-	-	-
Arrington et al. [181]	Behavioral Modeling	-	-	-
Li et al. [190]	KNN	98.5%	-	-
Pajouh et al. [182]	Naïve Bayes	84.86%	-	-
Ghosh and Mitra. [183]	Logistic Regression	-	-	93.26%
Prokofiev et al. [184]	Logistic Regression	94.0%	98.0%	97.30%
Singh and Neetesh [191]	Self-Organizing Map	64%	-	-

The “Applicable with Edge” column was established based on our observation on various system implementation.

7.1. Evaluation Metrics for NIDS in IoT

This subsection investigates the various metrics that are mostly used and also other metrics that are not widely considered in NIDS design and implementation for IoTs. The fundamental purposes of evaluating NIDS are:

- Compare two or more NIDS on a similar scale.
- Evaluate the required performances.
- Determine the best configuration of the NIDS.

7.1.1. Performance and Evaluation Metrics Used in ML-Based NIDS for IoT Systems

Cardenas et al. [192] proposed that the Bayesian detection rates, the cost effects of failure detection rates [193], and the detection capabilities of any NIDS must be critical metrics to study. In [194], NIDS were evaluated based on the scorecard metrics in real-time and distributed systems. They argued that the data payload of the packets under analysis by an NIDS must contain realistic content. From their conclusion, the best way to evaluate any NIDS is by using real-time data from the deployed devices’ network rather than modeling or simulation. Data packets classified as an attack in a network using some ML-based NIDS algorithm may be regarded as legitimate packets by other classification models. Researchers use many metrics to evaluate NIDS. No single metric seems adequate to measure the capability and performance of NIDS. According to [195], the performance of an NIDS can be determined based on the Detection Rate (DR) and the False Alarm Rate (FAR). All the above analyses indicate that the performance of the NIDS designed for an IoT use case may differ. The implementation process, type of attack, IoT device’s resource, and the ML used during the NIDS contribute to the type of metrics.

However, the most common metrics used by researchers include Accuracy, Recall/ Sensitivity, and F1-Score. Metrics such as false negative rate (FNR) (the number of negative samples predicted expressed as a ratio to the available, total positive samples) and false positive rate (FPR) (the ratio of positive samples predicted to the total predicted positive samples) are used to measure the performance of an NIDS for IoT systems.

7.1.2. Other Required Metrics

When designing NIDS for massive IoT devices, researchers must consider some critical issues. Conventional IoT devices are energy resource-constrained. It is hard to recharge or replace the batteries, especially in a dynamic-isolated environment [196]. Some IoT devices are also deployed for specific jobs and therefore equipped with limited computational power and limited storage capacities. Hence, the power consumption and the IoT devices' storage capacities are both critical factors in designing NIDS.

For instance, an IoT device equipped with low operating rate and limited storage capacity will require a constant power supply in order to measure the environment condition in a remote area. The IoT device has to transmit the measured weather data in a timely manner to a proximal MEC platform for processing [197]. Therefore, any NIDS design for such a system must be evaluated with these hardware settings. Most current research works ignored the importance of accommodating these critical considerations in their designs and performance evaluations.

7.2. Datasets Used in IoT Systems NIDS Design

The dataset is an essential part of an ML-based NIDS design. It consists of features observed from normal and abnormal operations of the targeted systems. For IoT networks, innovative methods and detecting algorithms necessitated a well-designed dataset. The most common data generation sources for NIDS for IoT are network packet extraction flows, system logs, and sessions. Building a dataset specifically for IoT NIDS can be complicated and time consuming. Table 6 shows an overview of several selected public datasets widely used by the research community for NIDS design.

Most researchers choose to create their own datasets to train the ML-based NIDS. Regardless of the difficulty in dataset construction, a common and well-established benchmark dataset is required in order to evaluate and to compare models. Researchers have used several datasets to train NIDS. The most commonly used datasets in the literature are the KDD-Cup'99 [198], which was created for the KDD competition, and it contains 41 attributes similar to a NetFlow dataset.

The UNB-ISCX 2012 [199], CICIDS2017 [200], and the AWS(CSE-CIC-IDS2018) [201] datasets were created by the Canadian Institute of Cybersecurity. According to the authors [200], the dataset was based on five days of normal and attacks traffic data. The CICIDS2017 contains most of the necessary modern and updated attack criteria such as DoS, DDoS, Brute Force, XSS, SQL Injection, Infiltration, Portscan, and Botnet. The dataset has 80 features extracted using the flow meter [202]. Another popular dataset currently used as a benchmark for NIDS in IoT is the $UNSW-NB_{15}$, which was created by the defense force academy of University of New South Wales Australia [203,204]. The $UNSW-NB_{15}$ was created based on the current attack categories through realistic network operations. The $UNSW-NB_{15}$ dataset consists of ten (10) categories of attacks: Analysis, Backdoor, DoS, Exploit, Fizzers, Generic, Reconnaissance, Shellcode, and Worm. Recently, there are several open source datasets that target different IoT attacks. In [205], the authors simulated a variety of network attacks in an IoT environment. Their dataset consists of benign, mitm-arpspoofing, DoS-synflooding, scan-hostport, scan-portos, and Mirai Botnet. Ullah et al. [206] created a dataset for IoT NIDS known as IoTID20. The authors proposed the IoT botnet dataset, which is suitable for DoS attacks. Moustafa [207] proposed a new dataset called TON_IoT, which contains IoT/IIoT service telemetry, as well as Operating System logs and IoT network traffic, obtained from a realistic approximation of a medium-scale network at the UNSW Canberra Cyber Range and IoT Labs.

The OPCUA dataset [208] was created by developing and injecting several attacks on a CPPS testbed based on OPC UA to allow users to assess the efficiency of various strategies for built Intrusion Detection Systems (IDS) in the industrial environment. Table 6 provides detailed information about NIDS in IoT systems. Datasets such as the ELEGANT dataset [209–211] targets DoS/DDoS attacks in IoT and SDN-based IoT networks, [212] detects anomalies in industrial IoT, Mqtt-iot-ids2020 dataset [213] detects network intrusion in MQTT protocol IoT network, Edge-IIoTset dataset [214] was also created for industrial IoT to detect DoS/DDoS attacks, Information gathering, Man in the middle attacks, Injection attacks, and Malware attacks. The X-IIoTID dataset [215] is a dataset for fitting the heterogeneity and interoperability of industrial IoT systems that is network and device agnostic. The final dataset on the NIDS IoT system radar is the IoT-BDA Botnet Analysis Dataset [216], which was created to make it easier to refine and create host and network-based IoT botnet detection systems. All the above datasets are publicly available for researchers to design robust NIDS to secure the IoT system.

Table 6. Some Popular Public Datasets for NIDS for IoT.

Dataset Description	Telemetry	Attack Types	Year Created
NSL-KDD [217]	General Purpose		2009
UNB-ISCX 2012 [199]	General Purpose	DoS, DDoS, Brute Force, XSS, SQL Injection, Infiltration, Portscan, and Botnet	2012
$UNSW-NB_{15}$ [203,204]	General Purpose	Analysis, Backdoor, DoS, Exploit, Fizzers, Generic, Reconnaissance, Shellcode, and Worm	2015
CICIDS2017 [200]	General Purpose	DoS, DDoS, Brute Force, XSS, SQL Injection, Infiltration, Portscan, and Botnet	2017
AWS(CSE-CIC-IDS2018) [201]	General Purpose	DoS, DDoS, Brute Force, XSS, SQL Injection, Infiltration, Portscan, and Botnet	2018
IoT Network Intrusion Dataset [205]	IoT	Benign, mitm-arpspoofing, DoS-synflooding, scan-hostport, scan-portos, mirai-udpflooding, mirai-ackflooding, mirai-httpflooding, mirai-hostbruteforce	2019
IoTID20 Dataset [206]	IoT	DoS/DDoS attacks	2020
TON_IoT Dataset [207]	IoT, IIoT, MEC, Fog	DoS, DDoS and ransomware	2019
OPCUA dataset [208]	IoT	DoS, Eavesdropping, Man-in-the-middle, Impersonation, Spoofing attacks	2020
IOT DOS AND DDOS ATTACK DATASET [209–211,218]	IoT	DoS and DDoS attacks	2021
IOT HEALTHCARE SECURITY DATASET [219]	IoT, SDN, IIoT	Normal and IoT attack traffic	2021

Table 6. *Cont.*

Dataset Description	Telemetry	Attack Types	Year Created
Mqtt-iot-ids2020 DATASET [213]	IoI, IIoT	Normal operation, aggressive scan, UDP scan, Sparta SSH brute-force, and MQTT brute-force attack.	2020
Edge-IIoTset DATASET [214]	IoT, IIoT, MEC	DoS/DDoS attacks, Information gathering, Man in the middle attacks, Injection attacks, and Malware attacks	2022
NSS MIRAI DATASET [220]	IoT	Mirai type botnet attack	2020
WUSTL-IIOT-2021 [221]	IoT	Command Injection, DoS, Reconnaissance, Backdoor	2021
IOT-BDA BOTNET ANALYSIS DATASET [216]	IoT	Port scanning, exploitation, C2 communications and DDoS	2021
THE BOT-IOT DATASET [222]	IoT	DDoS, DoS, OS and Service Scan, Keylogging and Data exfiltration attacks	2019
X-IIoTID dataset [215]	IoT, IIoT	Brute force attack, dictionary attack, and the malicious insider, reverse shell and Man-in-the-Middle	2021

Note: Network were captured as a Pcap. Users can extract features and records based on their preference.

8. Discussion and Future Trends

With an extensive increase in different IoT architectures ranging from well-defined documentation to unstructured standards, the IoT is confronted with new security issues. Due to the increase in cyber-attacks and the diversified capabilities of the IoT devices, manufacturers and developers need to adopt to a common standard when designing NIDS for these cyber-physical devices. The utilization of ML and the concept of MEC should be considered as a significant component in the NIDS design for IoT. This section examines a systematic process for choosing NIDS for IoTs and proposes an NIDS framework using MEC.

8.1. Choosing NIDS for IoT Systems

NIDS developers must consider some major factors when choosing the NIDSs for IoT systems. Firstly, the available resources of the IoT system must be analyzed, especially when the edge or fog computing platform can be utilized. The next factor to examine is how to implement a proper NIDS. Figure 4 summarizes the process of choosing an NIDS for the IoT systems. The specific IoT application also should be considered to choose the type of NIDS. Assuming a network consists of several IoT systems connecting to an MEC platform, a distributed anomaly-based ML model will be adequate.

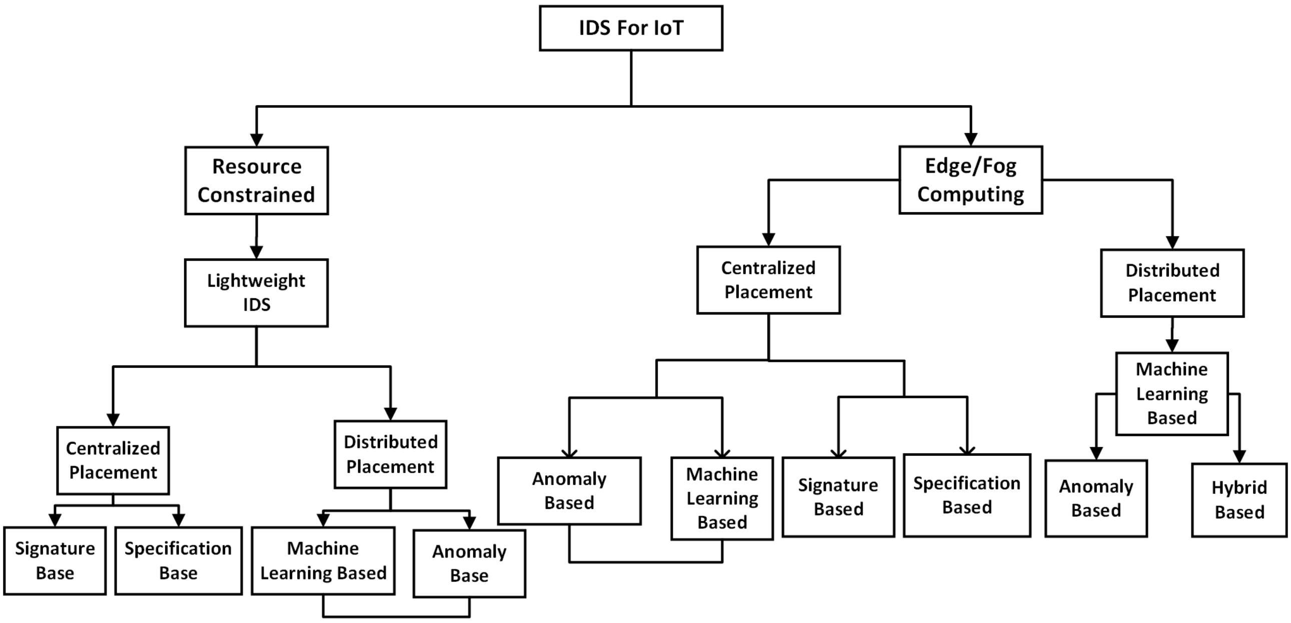

Figure 4. Structure of Choosing NIDS for IoT.

If a single IoT device is installed remotely with limited resources, a good choice of NIDS for such a system will be a signature-based with centralized placement. Such a system should be updated frequently to identify new attacks. A hybrid-based NIDS will work well in an IoT MEC environment with equipped resources. In Figure 4, a link joining the anomaly-based and the ML-based NIDS demonstrates that combining the two approaches will be a suitable approach. The single arrow pointing to a method indicates that such a method alone will be a suitable choice. It is important to consider all the above scenarios in order for the manufacturers and developers to select the right NIDS for a certain IoT system.

8.2. Proposed NIDS Model for IoT Systems

A proposed model of distributed NIDS comprising of the IoT end-dive, edge, and the cloud has been elaborated in Figure 5. As explained in previous sections, IoT MEC has emerged to help resolve the resource constraints in IoT. Therefore, designing NIDS utilizing MEC will allow the implementation of a sophisticated high-end security system for IoT devices. Moreover, using the proposed model will help to control the amount of traffic that needs to be analyzed by the NIDS model on the MEC platform. The model does not require extensive resources on the edge platform to control the wide-range connected IoT devices.

In our proposed model, the training and retraining are located in the cloud, since it contains enough resources. Two different models, including a highly in-depth NIDS and a lightweight binary NIDS, can be created using the same dataset in the cloud. The lightweight model will be implemented in the IoT devices. The lightweight NIDS performs binary classification to check the availability of intrusion in the IoT end device. If an intrusion is detected, it then forwards the captured data to the edge platform for in-depth intrusion analysis. The results of the two models are compared and the decision is made. This will help reduce the false positive rates, which is a significant problem in the ML-based NIDS. Retraining is scheduled when there are new intrusions detected. Since MEC platforms can monitor and provide resources for a large number of IoT devices, the lightweight model will detect and control whether in-depth analysis should be performed on edge.

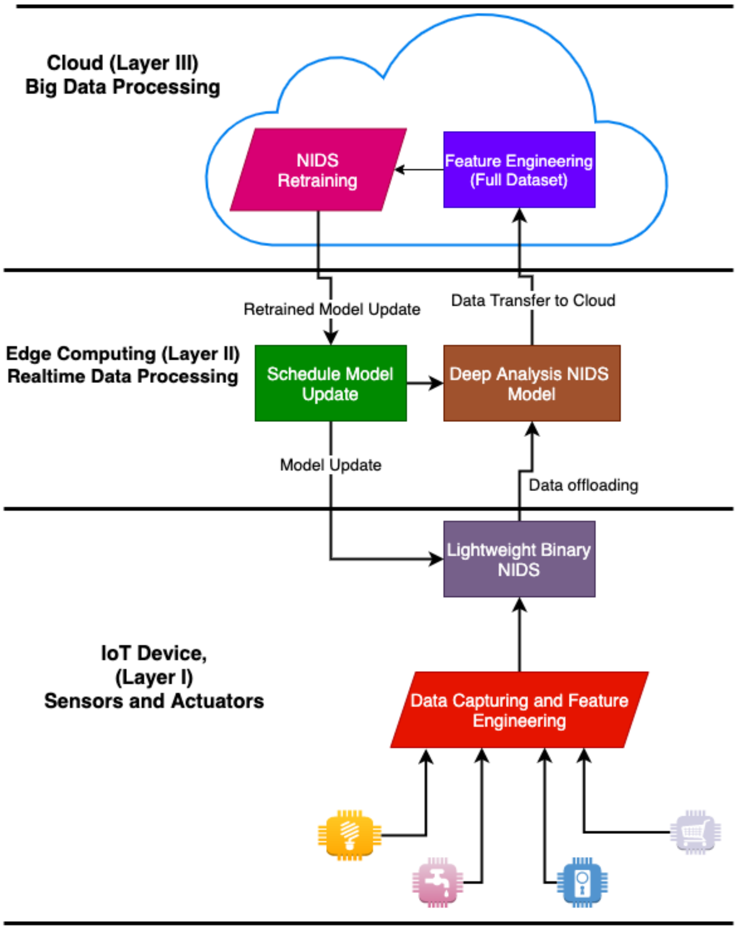

Figure 5. Proposed NIDS Model.

9. Conclusions

The necessity to secure IoT systems has stimulated many novel solutions to NIDS design. This paper has conducted a comprehensive review on NIDS that leverages MEC and ML. A taxonomy and tabular classification of detection methods, NIDS placement strategies, security threats, and validation methods were illustrated. We have discovered that there is a large body of theoretical studies. However, real-world NIDS development has not been fully testified yet. There are no specific detection techniques and deployment methods that have been accepted as standards to secure IoT systems. It still requires more effort to design a practical NIDS solution that effectively detects cyber-attacks in real IoT systems. Furthermore, critical IoT system evaluation metrics such as energy consumption, processing, and storage efficiency are not considered in most of the related studies.

Through our thoughtful review, we have identified several potential research directions. Firstly, future NIDS for IoT systems should focus on addressing the following issues: (1) improve the effectiveness of NIDS; and (2) demonstrate how their proposed system can be implemented in realistic IoT-MEC infrastructure. Secondly, researchers need to consider the critical metrics discussed in Section 7 for future NIDS design. We are confident that this survey and our proposed framework will serve as references and guidelines for researchers developing NIDS for IoT systems.

Author Contributions: E.G. prepared the paper while A.J. also reviewed and provided extensive ideas. All authors have read and agreed to the published version of the manuscript.

Funding: UCD School of Computer Science.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Not Applicable.

Conflicts of Interest: The authors declare that there were no commercial or financial ties that might be considered as a possible conflict of interest during the research.

References

1. Li, H.; Ota, K.; Dong, M. Learning IoT in edge: Deep learning for the Internet of Things with edge computing. *IEEE Netw.* **2018**, *32*, 96–101. [CrossRef]
2. Stoyanova, M.; Nikoloudakis, Y.; Panagiotakis, S.; Pallis, E.; Markakis, E.K. A survey on the internet of things (IoT) forensics: Challenges, approaches, and open issues. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1191–1221. [CrossRef]
3. Yastrebova, A.; Kirichek, R.; Koucheryavy, Y.; Borodin, A.; Koucheryavy, A. Future networks 2030: Architecture & requirements. In Proceedings of the 2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Moscow, Russia, 5–9 November 2018; pp. 1–8.
4. Jurcut, A.; Niculcea, T.; Ranaweera, P.; Le-Khac, N.A. Security considerations for Internet of Things: A survey. *SN Comput. Sci.* **2020**, *1*, 1–19. [CrossRef]
5. Shi, W.; Dustdar, S. The promise of edge computing. *Computer* **2016**, *49*, 78–81. [CrossRef]
6. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge computing: Vision and challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [CrossRef]
7. Ranaweera, P.; Jurcut, A.D.; Liyanage, M. Survey on multi-access edge computing security and privacy. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1078–1124. [CrossRef]
8. Pacheco, J.; Hariri, S. IoT security framework for smart cyber infrastructures. In Proceedings of the 2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS* W), Augsburg, Germany, 12–16 September 2016; pp. 242–247.
9. Borhani, M.; Liyanage, M.; Sodhro, A.H.; Kumar, P.; Jurcut, A.D.; Gurtov, A. Secure and resilient communications in the industrial internet. In *Guide to Disaster-Resilient Communication Networks*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 219–242.
10. Butun, I.; Morgera, S.D.; Sankar, R. A survey of intrusion detection systems in wireless sensor networks. *IEEE Commun. Surv. Tutor.* **2013**, *16*, 266–282. [CrossRef]
11. Alrajeh, N.A.; Khan, S.; Shams, B. Intrusion detection systems in wireless sensor networks: A review. *Distrib. Sens. Netw.* **2013**, *9*, 167575. [CrossRef]
12. Ammar, M.; Russello, G.; Crispo, B. Internet of Things: A survey on the security of IoT frameworks. *J. Inf. Secur. Appl.* **2018**, *38*, 8–27. [CrossRef]
13. Yang, Y.; Wu, L.; Yin, G.; Li, L.; Zhao, H. A survey on security and privacy issues in Internet-of-Things. *IEEE Internet Things J.* **2017**, *4*, 1250–1258. [CrossRef]
14. Sain, M.; Kang, Y.J.; Lee, H.J. Survey on security in Internet of Things: State of the art and challenges. In Proceedings of the 2017 19th International conference on advanced communication technology (ICACT), Pyeongchang, Korea, 19–22 February 2017; pp. 699–704.
15. Benabdessalem, R.; Hamdi, M.; Kim, T.H. A survey on security models, techniques, and tools for the internet of things. In Proceedings of the 2014 7th International Conference on Advanced Software Engineering and Its Applications, Hainan Island, China, 20–23 December 2014; pp. 44–48.
16. Jurcut, A.D.; Ranaweera, P.; Xu, L. Introduction to IoT security. In *IoT Security: Advances in Authentication*; John Wiley & Sons: Hoboken, NJ, USA, 2020; pp. 27–64. [CrossRef]
17. Pawar, P.; Trivedi, A. Device-to-device communication based IoT system: Benefits and challenges. *IETE Tech. Rev.* **2019**, *36*, 362–374. [CrossRef]
18. Ashouri, M.; Lorig, F.; Davidsson, P.; Spalazzese, R. Edge computing simulators for iot system design: An analysis of qualities and metrics. *Future Internet* **2019**, *11*, 235. [CrossRef]
19. Mahbub, M.; Gazi, M.S.A.; Provar, S.A.A.; Islam, M.S. Multi-Access Edge Computing-Aware Internet of Things: MEC-IoT. In Proceedings of the 2020 Emerging Technology in Computing, Communication and Electronics (ETCCE), London, UK, 19–20 August 2020; pp. 1–6. [CrossRef]
20. Xu, L.; Jurcut, A.D.; Ahmadi, H. Emerging Challenges and Requirements for internet of things in 5G. In *5G Enabled Internet Things*; CRC Press: Boca Raton, FL, USA, 2019; pp. 29–48.
21. Said, O.; Tolba, A. Accurate performance prediction of IoT communication systems for smart cities: An efficient deep learning based solution. *Sustain. Cities Soc.* **2021**, *69*, 102830. [CrossRef]
22. Catarinucci, L.; De Donno, D.; Mainetti, L.; Palano, L.; Patrono, L.; Stefanizzi, M.L.; Tarricone, L. An IoT-aware architecture for smart healthcare systems. *IEEE Internet Things J.* **2015**, *2*, 515–526. [CrossRef]
23. Almotiri, S.H.; Khan, M.A.; Alghamdi, M.A. Mobile health (m-health) system in the context of IoT. In Proceedings of the 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), Vienna, Austria, 22–24 August 2016; pp. 39–42.
24. Khoi, N.M.; Saguna, S.; Mitra, K.; Ahlund, C. Irehmo: An efficient iot-based remote health monitoring system for smart regions. In Proceedings of the 2015 17th International Conference on E-health Networking, Application and Services (HealthCom), Boston, MA, USA, 14–17 October 2015; pp. 563–568.

25. Coelho, C.; Coelho, D.; Wolf, M. An IoT smart home architecture for long-term care of people with special needs. In Proceedings of the 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), Milan, Italy 14–16 December 2015; pp. 626–627.
26. Wang, S.; Hou, Y.; Gao, F.; Ji, X. A novel IoT access architecture for vehicle monitoring system. In Proceedings of the 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), Reston, VA, USA, 12–14 December 2016; pp. 639–642.
27. Wang, S.; Hou, Y.; Gao, F.; Ma, S. A novel clock synchronization architecture for IoT access system. In Proceedings of the 2016 2nd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China 14–17 October 2016; pp. 1456–1459.
28. Shang, W.; Yu, Y.; Droms, R.; Zhang, L. *Challenges in IoT Networking via TCP/IP Architecture*; Technical Report; NDN-0038; NDN Project; Named Data Networking. 2016. Available online: https://named-data.net/wp-content/uploads/2016/02/ndn-0038-1-challenges-iot.pdf (accessed on 17 March 2022)
29. Alkhabbas, F.; Spalazzese, R.; Davidsson, P. Architecting emergent configurations in the internet of things. In Proceedings of the 2017 IEEE International Conference on Software Architecture (ICSA), Gothenburg, Sweden, 3–7 April 2017; pp. 221–224.
30. Park, S.; Crespi, N.; Park, H.; Kim, S.H. IoT routing architecture with autonomous systems of things. In Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, Korea, 6–8 March 2014; pp. 442–445.
31. Amadeo, M.; Campolo, C.; Iera, A.; Molinaro, A. Named data networking for IoT: An architectural perspective. In Proceedings of the 2014 European Conference on Networks and Communications (EuCNC), Bologna, Italy, 23–26 June 2014; pp. 1–5.
32. Ikram, M.A.; Alshehri, M.D.; Hussain, F.K. Architecture of an IoT-based system for football supervision (IoT Football). In Proceedings of the 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), Milan, Italy, 14–16 December 2015; pp. 69–74.
33. Lyu, X.; Tian, H.; Jiang, L.; Vinel, A.; Maharjan, S.; Gjessing, S.; Zhang, Y. Selective offloading in mobile edge computing for the green Internet of Things. *IEEE Netw.* **2018**, *32*, 54–60. [CrossRef]
34. Ali, B.; Gregory, M.A.; Li, S. Multi-access edge computing architecture, data security and privacy: A review. *IEEE Access* **2021**, *9*, 18706–18721. [CrossRef]
35. Khan, B.U.I.; Olanrewaju, R.F.; Anwar, F.; Mir, R.N.; Najeeb, A.R. A critical insight into the effectiveness of research methods evolved to secure IoT ecosystem. *Int. J. Inf. Comput. Secur.* **2019**, *11*, 332–354. [CrossRef]
36. Wang, H.; Zhang, Z.; Taleb, T. Special issue on security and privacy of IoT. *World Wide Web* **2018**, *21*, 1–6. [CrossRef]
37. Pongle, P.; Chavan, G. A survey: Attacks on RPL and 6LoWPAN in IoT. In Proceedings of the 2015 International conference on pervasive computing (ICPC), Pune, India 8–10 January 2015; pp. 1–6.
38. Hasan, M.; Mohan, S. Protecting actuators in safety-critical IoT systems from control spoofing attacks. In Proceedings of the 2nd International ACM Workshop on Security and Privacy for the Internet-of-Things, Xi'an, China, 30 May 2019; pp. 8–14.
39. Javed, A.; Malik, K.M.; Irtaza, A.; Malik, H. Towards protecting cyber-physical and IoT systems from single-and multi-order voice spoofing attacks. *Appl. Acoust.* **2021**, *183*, 108283. [CrossRef]
40. Aldabbas, H.; Amin, R. A novel mechanism to handle address spoofing attacks in SDN based IoT. *Clust. Comput.* **2021**, *24*, 3011–3026. [CrossRef]
41. Varalakshmi, I.; Thenmozhi, M.; Sasi, R. Detection of Distributed Denial of Service Attack in an Internet of Things Environment-A Review. In Proceedings of the 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), Iasi, Romania, 20–23 October 2021; pp. 1–6.
42. Ferrag, M.A.; Shu, L.; Djallel, H.; Choo, K.K.R. Deep learning-based intrusion detection for distributed denial of service attack in Agriculture 4.0. *Electronics* **2021**, *10*, 1257. [CrossRef]
43. Gupta, B.; Chaudhary, P.; Chang, X.; Nedjah, N. Smart defense against distributed Denial of service attack in IoT networks using supervised learning classifiers. *Comput. Electr. Eng.* **2022**, *98*, 107726. [CrossRef]
44. Alyas, M.; Noor, M.I.; Hassan, H. DDOS Attack Detection Strategies in Cloud A Comparative Stud. *VFAST Trans. Softw. Eng.* **2017**, *12*, 35–42. [CrossRef]
45. Kolias, C.; Kambourakis, G.; Stavrou, A.; Voas, J. DDoS in the IoT: Mirai and other botnets. *Computer* **2017**, *50*, 80–84. [CrossRef]
46. 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]
47. Abdalzaher, M.S.; Elwekeil, M.; Wang, T.; Zhang, S. A deep autoencoder trust model for mitigating jamming attack in IoT assisted by cognitive radio. *IEEE Syst. J.* **2021**. [CrossRef]
48. Kerrakchou, I.; Chadli, S.; Kharbach, A.; Saber, M. Simulation and Analysis of Jamming Attack in IoT Networks. In Proceedings of the International Conference on Digital Technologies and Applications, Moscow, Russia, 30–31 March 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 323–333.
49. Salem, O.; Alsubhi, K.; Shaafi, A.; Gheryani, M.; Mehaoua, A.; Boutaba, R. Man-in-the-Middle Attack Mitigation in Internet of Medical Things. *IEEE Trans. Ind. Inform.* **2021**, *18*, 2053–2062. [CrossRef]
50. Pospisil, O.; Fujdiak, R.; Mikhaylov, K.; Ruotsalainen, H.; Misurec, J. Testbed for LoRaWAN Security: Design and Validation through Man-in-the-Middle Attacks Study. *Appl. Sci.* **2021**, *11*, 7642. [CrossRef]
51. Liu, J.; Sun, W. Smart attacks against intelligent wearables in people-centric internet of things. *IEEE Commun. Mag.* **2016**, *54*, 44–49. [CrossRef]
52. Ziegeldorf, J.H.; Morchon, O.G.; Wehrle, K. Privacy in the Internet of Things: Threats and challenges. *Secur. Commun. Netw.* **2014**, *7*, 2728–2742. [CrossRef]
53. Kambourakis, G.; Kolias, C.; Stavrou, A. The mirai botnet and the iot zombie armies. In Proceedings of the MILCOM 2017—2017 IEEE Military Communications Conference (MILCOM), Baltimore, MD, USA, 23–25 October 2017; pp. 267–272.

54. Meidan, Y.; Bohadana, M.; Mathov, Y.; Mirsky, Y.; Shabtai, A.; Breitenbacher, D.; Elovici, Y. N-baiot—Network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Comput.* **2018**, *17*, 12–22. [CrossRef]
55. Rajan, A.; Jithish, J.; Sankaran, S. Sybil attack in IOT: Modelling and defenses. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Chennai, India, 3–5 August 2017; pp. 2323–2327.
56. Evangelista, D.; Mezghani, F.; Nogueira, M.; Santos, A. Evaluation of Sybil attack detection approaches in the Internet of Things content dissemination. In Proceedings of the 2016 Wireless Days (WD), Toulouse, France, 23–25 March 2016; pp. 1–6.
57. Li, J.; Zhao, Z.; Li, R.; Zhang, H. Ai-based two-stage intrusion detection for software defined iot networks. *IEEE Internet Things J.* **2018**, *6*, 2093–2102. [CrossRef]
58. Falco, G.; Viswanathan, A.; Caldera, C.; Shrobe, H. A master attack methodology for an AI-based automated attack planner for smart cities. *IEEE Access* **2018**, *6*, 48360–48373. [CrossRef]
59. Schneier, B. Security Risks of Embedded Systems—Schneier on Security. 2014. Available online: https://www.schneier.com/blog/archives/2014/01/security_risks_9.html (accessed on 13 May 2021).
60. Vaca, F.D. An Ensemble Learning Based Multi-level Network Intrusion Detection System for Wi-Fi Dominant Networks. Ph.D. Thesis, Purdue University Graduate School, Lafayette, IN, USA, 2019.
61. Abduvaliyev, A.; Pathan, A.S.K.; Zhou, J.; Roman, R.; Wong, W.C. On the vital areas of intrusion detection systems in wireless sensor networks. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 1223–1237. [CrossRef]
62. Ioulianou, P.; Vasilakis, V.; Moscholios, I.; Logothetis, M. A Signature-based Intrusion Detection System for the Internet of Things. *Inf. Commun. Technol. Form* **2018**, *In press*.
63. Kasinathan, P.; Pastrone, C.; Spirito, M.A.; Vinkovits, M. Denial-of-Service detection in 6LoWPAN based Internet of Things. In Proceedings of the 2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Lyon, France, 7–9 October 2013; pp. 600–607.
64. Olsson, J. 6LoWPAN demystified. *Tex. Instrum.* **2014**, *13*, 1–13.
65. Boujrad, M.; Lazaar, S.; Hassine, M. Performance Assessment of Open Source IDS for improving IoT Architecture Security implemented on WBANs. In Proceedings of the 3rd International Conference on Networking, Information Systems & Security, Msida, Malta, 15–17 December 2020; pp. 1–4.
66. Hodo, E.; Bellekens, X.; Hamilton, A.; Dubouilh, P.L.; Iorkyase, E.; Tachtatzis, C.; Atkinson, R. Threat analysis of IoT networks using artificial neural network intrusion detection system. In Proceedings of the Networks, Computers and Communications (ISNCC), Marrakech, Morocco, 16–18 May 2016; pp. 1–6.
67. Hanif, S.; Ilyas, T.; Zeeshan, M. Intrusion detection in IoT using artificial neural networks on UNSW-15 dataset. In Proceedings of the 2019 IEEE 16th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT and AI (HONET-ICT), Charlotte, NC, USA, 6–9 October 2019; pp. 152–156.
68. Qureshi, A.U.H.; Larijani, H.; Ahmad, J.; Mtetwa, N. A heuristic intrusion detection system for Internet-of-Things (IoT). In *Proceedings of the Intelligent Computing, Proceedings of the Computing Conference, London, UK, 16–17 July 2019*; Springer: Berlin/Heidelberg, Germany. 2019; pp. 86–98.
69. Almogren, A.S. Intrusion detection in Edge-of-Things computing. *J. Parallel Distrib. Comput.* **2020**, *137*, 259–265. [CrossRef]
70. Tharewal, S.; Ashfaque, M.W.; Banu, S.S.; Uma, P.; Hassen, S.M.; Shabaz, M. Intrusion detection system for industrial Internet of Things based on deep reinforcement learning. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 9023719. [CrossRef]
71. Caminero, G.; Lopez-Martin, M.; Carro, B. Adversarial environment reinforcement learning algorithm for intrusion detection. *Comput. Netw.* **2019**, *159*, 96–109. [CrossRef]
72. Mothukuri, V.; Khare, P.; Parizi, R.M.; Pouriyeh, S.; Dehghantanha, A.; Srivastava, G. Federated learning-based anomaly detection for IoT security attacks. *IEEE Internet Things J.* **2021**, 9, 2545–2554. [CrossRef]
73. Mehta, D.; Mady, A.E.d.; Boubekeur, M. Anomaly-Based Intrusion Detection System for Embedded Devices on Internet. In Proceedings of the Tenth International Conference on Advances in Circuits, Electronics and Micro-electronics, Venice, Italy, 16–20 September 2018.
74. Canedo, J.; Skjellum, A. Using machine learning to secure IoT systems. In Proceedings of the 2016 14th Annual Conference on Privacy, Security and Trust, PST, Auckland, New Zealand, 12–14 December 2016; pp. 219–222. [CrossRef]
75. Wu, Y.; Dai, H.N.; Tang, H. Graph neural networks for anomaly detection in industrial internet of things. *IEEE Internet Things J.* **2021**. [CrossRef]
76. Pacheco, J.; Benitez, V.H.; Felix-Herran, L.C.; Satam, P. Artificial neural networks-based intrusion detection system for internet of things fog nodes. *IEEE Access* **2020**, *8*, 73907–73918. [CrossRef]
77. Alladi, T.; Agrawal, A.; Gera, B.; Chamola, V.; Sikdar, B.; Guizani, M. Deep neural networks for securing IoT enabled vehicular ad-hoc networks. In Proceedings of the ICC 2021-IEEE International Conference on Communications, Montreal, QC, Canada, 14–23 June 2021; pp. 1–6.
78. Eskandari, M.; Janjua, Z.H.; Vecchio, M.; Antonelli, F. Passban IDS: An intelligent anomaly-based intrusion detection system for IoT edge devices. *IEEE Internet Things J.* **2020**, *7*, 6882–6897. [CrossRef]
79. Alsoufi, M.A.; Razak, S.; Siraj, M.M.; Nafea, I.; Ghaleb, F.A.; Saeed, F.; Nasser, M. Anomaly-based intrusion detection systems in iot using deep learning: A systematic literature review. *Appl. Sci.* **2021**, *11*, 8383. [CrossRef]
80. Kumar, P.; Gupta, G.P.; Tripathi, R. Design of anomaly-based intrusion detection system using fog computing for IoT network. *Autom. Control Comput. Sci.* **2021**, *55*, 137–147. [CrossRef]

81. Heartfield, R.; Loukas, G.; Bezemskij, A.; Panaousis, E. Self-configurable cyber-physical intrusion detection for smart homes using reinforcement learning. *IEEE Trans. Inf. Forensics Secur.* **2020**, *16*, 1720–1735. [CrossRef]
82. Ma, X.; Shi, W. Aesmote: Adversarial reinforcement learning with smote for anomaly detection. *IEEE Trans. Netw. Sci. Eng.* **2020**, *8*, 943–956. [CrossRef]
83. Mitchell, R.; Chen, I.R. A survey of intrusion detection techniques for cyber-physical systems. *ACM Comput. Surv. (CSUR)* **2014**, *46*, 55. [CrossRef]
84. Le, A.; Loo, J.; Chai, K.K.; Aiash, M. A specification-based IDS for detecting attacks on RPL-based network topology. *Information* **2016**, *7*, 25. [CrossRef]
85. Althubaity, A.; Ammar, R.; Han, S. Detecting Rules-related Attacks in RPL-based Resource-Constrained Wireless Networks. In Proceedings of the 2020 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Louisville, KY, USA, 9–11 December 2020; pp. 1–8.
86. Kareem, M.A.; Tayeb, S. ML-based NIDS to secure RPL from routing attacks. In Proceedings of the 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), Virtual, 27–30 January 2021; pp. 1000–1006.
87. Sharma, V.; You, I.; Yim, K.; Chen, R.; Cho, J.H. BRIoT: Behavior rule specification-based misbehavior detection for IoT-embedded cyber-physical systems. *IEEE Access* **2019**, *7*, 118556–118580. [CrossRef]
88. Yun, K.; Astillo, P.V.; Lee, S.; Kim, J.; Kim, B.; You, I. Behavior-Rule Specification-based IDS for Safety-Related Embedded Devices in Smart Home. In Proceedings of the 2021 World Automation Congress (WAC), Virtual, 1–5 August 2021; pp. 65–70.
89. Ning, H.; Liu, H. Cyber-physical-social based security architecture for future internet of things. *Adv. Internet Things* **2012**, *2*, 1. [CrossRef]
90. Cervantes, C.; Poplade, D.; Nogueira, M.; Santos, A. Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things. In Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, Canada, 11–15 May 2015 pp. 606–611.
91. Krimmling, J.; Peter, S. Integration and evaluation of intrusion detection for CoAP in smart city applications. In Proceedings of the Communications and Network Security (CNS), San Francisco, CA, USA, 29–31 October 2014; pp. 73–78.
92. Nguyen, D.C.; Pathirana, P.N.; Ding, M.; Seneviratne, A. Privacy-preserved task offloading in mobile blockchain with deep reinforcement learning. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 2536–2549. [CrossRef]
93. Wu, H.; Wolter, K.; Jiao, P.; Deng, Y.; Zhao, Y.; Xu, M. EEDTO: An energy-efficient dynamic task offloading algorithm for blockchain-enabled IoT-edge-cloud orchestrated computing. *IEEE Internet Things J.* **2020**, *8*, 2163–2176. [CrossRef]
94. Gassais, R.; Ezzati-Jivan, N.; Fernandez, J.M.; Aloise, D.; Dagenais, M.R. Multi-level host-based intrusion detection system for Internet of things. *J. Cloud Comput.* **2020**, *9*, 1–16. [CrossRef]
95. Wang, X.; Lu, X. A host-based anomaly detection framework using XGBoost and LSTM for IoT devices. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, 8838571. [CrossRef]
96. Kfouri, G.d.O.; Gonçalves, D.G.; Dutra, B.V.; de Alencastro, J.F.; de Caldas Filho, F.L.; e Martins, L.M.; Praciano, B.J.; de Oliveira Albuquerque, R.; de Sousa, R.T., Jr. Design of a distributed HIDS for IoT backbone components. *Ann. Comput. Sci. Inf. Syst.* **2019**, 81–88. [CrossRef]
97. Smys, S.; Basar, A.; Wang, H. Hybrid intrusion detection system for internet of things (IoT). *J. ISMAC* **2020**, *2*, 190–199. [CrossRef]
98. Blakley, J. Open Edge Computing Initiative. 2021. Available online: https://www.openedgecomputing.org/ (accessed on 5 January 2022).
99. ETSI, M. Mobile edge computing (mec): Framework and reference architecture. *ETSI DGS MEC* **2016**, *3*, 1–18.
100. Ai, Y.; Peng, M.; Zhang, K. Edge computing technologies for Internet of Things: A primer. *Digit. Commun. Netw.* **2018**, *4*, 77–86. [CrossRef]
101. Gyamfi, E.; Jurcut, A. Novel Online Network Intrusion Detection System for Industrial IoT based on OI-SVDD and AS-ELM *IEEE Internet Things J.* **2022**. [CrossRef]
102. Corcoran, P.; Datta, S.K. Mobile-edge computing and the Internet of Things for consumers: Extending cloud computing and services to the edge of the network. *IEEE Consum. Electron. Mag.* **2016**, *5*, 73–74. [CrossRef]
103. Perera, C.; Zaslavsky, A.; Christen, P.; Georgakopoulos, D. Context aware computing for the internet of things: A survey. *IEEE Commun. Surv. Tutor.* **2013**, *16*, 414–454. [CrossRef]
104. Suo, H.; Liu, Z.; Wan, J.; Zhou, K. Security and privacy in mobile cloud computing. In Proceedings of the 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC), Sardinia, Italy, 1–5 July 2013; pp. 655–659.
105. Li, B.; Wu, W.; Zhao, W.; Zhang, H. Security enhancement with a hybrid cooperative noma scheme for mec system. *IEEE Trans. Veh. Technol.* **2021**, *70*, 2635–2648. [CrossRef]
106. Li, L.; Ota, K.; Dong, M. DeepNFV: A lightweight framework for intelligent edge network functions virtualization. *IEEE Netw.* **2018**, *33*, 136–141. [CrossRef]
107. Mastorakis, S.; Mtibaa, A.; Lee, J.; Misra, S. Icedge: When edge computing meets information-centric networking. *IEEE Internet Things J.* **2020**, *7*, 4203–4217. [CrossRef]
108. Kaur, K.; Garg, S.; Aujla, G.S.; Kumar, N.; Rodrigues, J.J.; Guizani, M. Edge computing in the industrial internet of things environment: Software-defined-networks-based edge-cloud interplay. *IEEE Commun. Mag.* **2018**, *56*, 44–51. [CrossRef]
109. He, Y.; Yu, F.R.; Zhao, N.; Leung, V.C.; Yin, H. Software-defined networks with mobile edge computing and caching for smart cities: A big data deep reinforcement learning approach. *IEEE Commun. Mag.* **2017**, *55*, 31–37. [CrossRef]

110. Baktir, A.C.; Ozgovde, A.; Ersoy, C. How can edge computing benefit from software-defined networking: A survey, use cases, and future directions. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2359–2391. [CrossRef]
111. Wang, Q.; Hu, H.; Sun, H.; Hu, R.Q. Secure and energy-efficient offloading and resource allocation in a NOMA-based MEC network. In Proceedings of the 2020 IEEE/ACM Symposium on Edge Computing (SEC), San Jose, CA, USA, 12–14 November 2020; pp. 420–424.
112. Ahmad, I.; Lembo, S.; Rodriguez, F.; Mehnert, S.; Vehkaperä, M. Security of Micro MEC in 6G: A Brief Overview. In Proceedings of the 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 8–11 January 2022; pp. 332–337.
113. Ansere, J.A.; Han, G.; Wang, H.; Choi, C.; Wu, C. A Reliable Energy Efficient Dynamic Spectrum Sensing for Cognitive Radio IoT Networks. *IEEE Internet Things J.* **2019**, *6*, 6748–6759. [CrossRef]
114. Liao, R.F.; Wen, H.; Wu, J.; Pan, F.; Xu, A.; Song, H.; Xie, F.; Jiang, Y.; Cao, M. Security enhancement for mobile edge computing through physical layer authentication. *IEEE Access* **2019**, *7*, 116390–116401. [CrossRef]
115. Ranaweera, P.; Jurcut, A.D.; Liyanage, M. Realizing multi-access edge computing feasibility: Security perspective. In Proceedings of the 2019 IEEE Conference on Standards for Communications and Networking (CSCN), Paris, France, 29–31 October 2019; pp. 1–7.
116. Aazam, M.; Zeadally, S.; Harras, K.A. Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities. *Future Gener. Comput. Syst.* **2018**, *87*, 278–289. [CrossRef]
117. Wang, Y.; Meng, W.; Li, W.; Liu, Z.; Liu, Y.; Xue, H. Adaptive machine learning-based alarm reduction via edge computing for distributed intrusion detection systems. *Concurr. Comput. Pract. Exp.* **2019**, *31*, e5101. [CrossRef]
118. Thanigaivelan, N.K.; Nigussie, E.; Kanth, R.K.; Virtanen, S.; Isoaho, J. Distributed internal anomaly detection system for Internet-of-Things. In Proceedings of the Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2016; pp. 319–320.
119. Ferdowsi, A.; Saad, W. Generative adversarial networks for distributed intrusion detection in the internet of things. *arXiv* **2019**, arXiv:1906.00567.
120. Singh, J.; Bello, Y.; Hussein, A.R.; Erbad, A.; Mohamed, A. Hierarchical security paradigm for iot multiaccess edge computing. *IEEE Internet Things J.* **2020**, *8*, 5794–5805. [CrossRef]
121. Yang, L.; Zhang, H.; Li, M.; Guo, J.; Ji, H. Mobile edge computing empowered energy efficient task offloading in 5 G. *IEEE Trans. Veh. Technol.* **2018**, *67*, 6398–6409. [CrossRef]
122. Abeysekara, P.; Dong, H.; Qin, A.K. Machine learning-driven trust prediction for mec-based iot services. In Proceedings of the 2019 IEEE International Conference on Web Services (ICWS), Milan, Italy, 8–13 July 2019; pp. 188–192.
123. Alladi, T.; Kohli, V.; Chamola, V.; Yu, F.R.; Guizani, M. Artificial intelligence (AI)-empowered intrusion detection architecture for the internet of vehicles. *IEEE Wirel. Commun.* **2021**, *28*, 144–149. [CrossRef]
124. Dinh, T.Q.; Tang, J.; La, Q.D.; Quek, T.Q. Offloading in mobile edge computing: Task allocation and computational frequency scaling. *IEEE Trans. Commun.* **2017**, *65*, 3571–3584.
125. Chen, X.; Zhang, H.; Wu, C.; Mao, S.; Ji, Y.; Bennis, M. Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning. *IEEE Internet Things J.* **2018**, *6*, 4005–4018. [CrossRef]
126. Iqbal, M.A.; Bayoumi, M. Secure End-to-End key establishment protocol for resource-constrained healthcare sensors in the context of IoT. In Proceedings of the 2016 International Conference on High Performance Computing & Simulation (HPCS), Pasadena, CA, USA, 3–6 April 2016; pp. 523–530.
127. Ranaweera, P.; Jurcut, A.; Liyanage, M. MEC-enabled 5G use cases: A survey on security vulnerabilities and countermeasures. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–37. [CrossRef]
128. Xu, Y.; Zhang, T.; Yang, D.; Liu, Y.; Tao, M. Joint resource and trajectory optimization for security in UAV-assisted MEC systems. *IEEE Trans. Commun.* **2020**, *69*, 573–588. [CrossRef]
129. Ranaweera, P.; Imrith, V.N.; Liyanag, M.; Jurcut, A.D. Security as a service platform leveraging multi-access edge computing infrastructure provisions. In Proceedings of the ICC 2020-2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–6.
130. Sha, K.; Errabelly, R.; Wei, W.; Yang, T.A.; Wang, Z. Edgesec: Design of an edge layer security service to enhance iot security. In Proceedings of the 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC), Madrid, Spain, 14–15 May 2017; pp. 81–88.
131. Singh, A.; Auluck, N.; Rana, O.; Jones, A.; Nepal, S. RT-SANE: Real time security aware scheduling on the network edge. In Proceedings of the10th International Conference on Utility and Cloud Computing, Austin, TX, USA, 5–8 December 2017; pp. 131–140.
132. Kamma, P.K.; Palla, C.R.; Nelakuditi, U.R.; Yarrabothu, R.S. Design and implementation of 6LoWPAN border router. In Proceedings of the Wireless and Optical Communications Networks (WOCN), Mumbai, India, 24–26 February 2016; pp. 1–5.
133. Wallgren, L.; Raza, S.; Voigt, T. Routing Attacks and Countermeasures in the RPL-based Internet of Things. *Int. J. Distrib. Sens. Netw.* **2013**, *9*, 794326. [CrossRef]
134. Pongle, P.; Chavan, G. Real time intrusion and wormhole attack detection in internet of things. *Int. J. Comput. Appl.* **2015**, *121*, 1–9. [CrossRef]

135. Gyamfi, E.; Jurcut, A. A Robust Security Task Offloading in Industrial IoT-Enabled Distributed Multi-Access Edge Computing. *Front. Signal Proc.* **2022**, *2*, 788943. [CrossRef]
136. Gyamfi, E.; Ansere, J.A.; Kamal, M.; Tariq, M.; Jurcut, A. An Adaptive Network Security System for IoT-Enabled Maritime Transportation. In Proceedings of the EEE Transactions on Intelligent Transportation Systems, Macau, China, 8–12 October 2022.
137. Abraham, A.; Thomas, J. Distributed intrusion detection systems: A computational intelligence approach. In *Applications of Information Systems to Homeland Security and Defense*; IGI Global: Hershey, PA, USA, 2006; pp. 107–137.
138. Giang, N.K.; Blackstock, M.; Lea, R.; Leung, V.C. Developing iot applications in the fog: A distributed dataflow approach. In Proceedings of the 2015 5th International Conference on the Internet of Things (IOT), Seoul, Korea 26–28 October 2015; pp. 155–162.
139. Halinen, A.; Törnroos, J.Å.; Elo, M. Network process analysis: An event-based approach to study business network dynamics. *Ind. Mark. Manag.* **2013**, *42*, 1213–1222. [CrossRef]
140. Sheikhan, M.; Bostani, H. A hybrid intrusion detection architecture for internet of things. In Proceedings of the 2016 8th International Symposium on Telecommunications (IST), Tehran, Iran, 27–28 September 2016; pp. 601–606.
141. Hosseinpour, F.; Vahdani Amoli, P.; Plosila, J.; Hämäläinen, T.; Tenhunen, H. An intrusion detection system for fog computing and IoT based logistic systems using a smart data approach. *Int. J. Digit. Content Technol. Its Appl.* **2016**, *10*, 1–14.
142. Gajewski, M.; Batalla, J.M.; Mastorakis, G.; Mavromoustakis, C.X. A distributed IDS architecture model for Smart Home systems. *Clust. Comput.* **2019**, *22*, 1739–1749. [CrossRef]
143. Khan, Z.A.; Herrmann, P. A trust based distributed intrusion detection mechanism for internet of things. In Proceedings of the 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA), Taipei, Taiwan, 27–29 March 2017; pp. 1169–1176.
144. Zhou, C.V.; Leckie, C.; Karunasekera, S. A survey of coordinated attacks and collaborative intrusion detection. *Comput. Secur.* **2010**, *29*, 124–140. [CrossRef]
145. Arshad, J.; Azad, M.A.; Abdellatif, M.M.; Rehman, M.H.U.; Salah, K. COLIDE: A collaborative intrusion detection framework for Internet of Things. *IET Netw.* **2018**, *8*, 3–14. [CrossRef]
146. Cho, E.J.; Kim, J.H.; Hong, C.S. Attack model and detection scheme for botnet on 6LoWPAN. In Proceedings of the Asia-Pacific Network Operations and Management Symposium, Jeju Island, Korea, 23–25 September 2009; Springer: Berlin/Heidelberg, Germany, 2009; pp. 515–518.
147. Jun, C.; Chi, C. Design of complex event-processing IDS in internet of things. In Proceedings of the 2014 Sixth International Conference on Measuring Technology and Mechatronics Automation, Zhangjiajie, China, 10–11 January 2014; pp. 226–229.
148. Chen, R.; Liu, C.M.; Chen, C. An artificial immune-based distributed intrusion detection model for the internet of things. Advanced materials research. *Trans. Tech. Publ.* **2012**, *366*, 165–168.
149. Abeshu, A.; Chilamkurti, N. Deep learning: The frontier for distributed attack detection in fog-to-things computing. *IEEE Commun. Mag.* **2018**, *56*, 169–175. [CrossRef]
150. Diro, A.A.; Naveen, C. Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Gener. Comput. Syst.* **2018**, *82*, 761–768. [CrossRef]
151. Diro, A.; Chilamkurti, N. Leveraging LSTM networks for attack detection in fog-to-things communications. *IEEE Commun. Mag.* **2018**, *56*, 124–130. [CrossRef]
152. Sohal, A.S.; Sandhu, R.; Sood, S.K.; Chang, V. A cybersecurity framework to identify malicious edge device in fog computing and cloud-of-things environments. *Comput. Secur.* **2018**, *74*, 340–354. [CrossRef]
153. Stolfo, S.J.; Salem, M.B.; Keromytis, A.D. Fog computing: Mitigating insider data theft attacks in the cloud. In Proceedings of the 2012 IEEE symposium on security and privacy workshops, Francisco, CA USA, 24–25 May 2012; pp. 125–128.
154. Sudqi Khater, B.; Abdul Wahab, A.; Idris, M.; Abdulla Hussain, M.; Ahmed Ibrahim, A. A Lightweight Perceptron-Based Intrusion Detection System for Fog Computing. *Appl. Sci.* **2019**, *9*, 178. [CrossRef]
155. An, X.; Zhou, X.; Lü, X.; Lin, F.; Yang, L. Sample selected extreme learning machine based intrusion detection in fog computing and MEC. *Wirel. Commun. Mob. Comput.* **2018**, *2018*. [CrossRef]
156. Kovanen, T.; David, G.; Hämäläinen, T. Survey: Intrusion detection systems in encrypted traffic. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 281–293.
157. Yin, X.C.; Liu, Z.G.; Nkenyereye, L.; Ndibanje, B. Toward an applied cyber security solution in IoT-based smart grids: An intrusion detection system approach. *Sensors* **2019**, *19*, 4952. [CrossRef] [PubMed]
158. Qiu, H.; Dong, T.; Zhang, T.; Lu, J.; Memmi, G.; Qiu, M. Adversarial attacks against network intrusion detection in iot systems. *IEEE Internet Things J.* **2020**, *8*, 10327–10335. [CrossRef]
159. Husnain, M.; Hayat, K.; Cambiaso, E.; Fayyaz, U.U.; Mongelli, M.; Akram, H.; Ghazanfar Abbas, S.; Shah, G.A. Preventing MQTT Vulnerabilities Using IoT-Enabled Intrusion Detection System. *Sensors* **2022**, *22*, 567. [CrossRef]
160. Jiang, X.; Qiu, T.; Zhou, X.; Zhang, B.; Sun, X.; Chi, J. A Text Similarity-based Protocol Parsing Scheme for Industrial Internet of Things. In Proceedings of the 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Dalian, China, 5–7 May 2021; pp. 781–787.
161. Chaabouni, N.; Mosbah, M.; Zemmari, A.; Sauvignac, C.; Faruki, P. Network intrusion detection for IoT security based on learning techniques. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2671–2701. [CrossRef]

162. Khan, M.A.; Karim, M.; Kim, Y. A scalable and hybrid intrusion detection system based on the convolutional-LSTM network. *Symmetry* **2019**, *11*, 583. [CrossRef]
163. Liu, H.; Lang, B.; Liu, M.; Yan, H. CNN and RNN based payload classification methods for attack detection. *Knowl. Based Syst.* **2019**, *163*, 332–341. [CrossRef]
164. Vieira, L.; Santos, L.; Gonçalves, R.; Rabadão, C. Identifying attack signatures for the internet of things: An IP flow based approach. In Proceedings of the 2019 14th Iberian Conference on Information Systems and Technologies (CISTI), Coimbra, Portugal, 19–22 June 2019; pp. 1–7.
165. Pontes, C.F.; de Souza, M.M.; Gondim, J.J.; Bishop, M.; Marotta, M.A. A new method for flow-based network intrusion detection using the inverse Potts model. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 1125–1136. [CrossRef]
166. Tiburski, R.T.; Moratelli, C.R.; Johann, S.F.; Neves, M.V.; de Matos, E.; Amaral, L.A.; Hessel, F. Lightweight security architecture based on embedded virtualization and trust mechanisms for IoT edge devices. *IEEE Commun. Mag.* **2019**, *57*, 67–73. [CrossRef]
167. Sedjelmaci, H.; Senouci, S.M.; Al-Bahri, M. A lightweight anomaly detection technique for low-resource IoT devices: A game-theoretic methodology. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 23–27 May 2016; pp. 1–6.
168. MALEH, Y.; Ezzati, A. Lightweight Intrusion Detection Scheme for Wireless Sensor Networks. *IAENG Int. J. Comput. Sci.* **2015**, *42*, 1–8.
169. Zhao, S.; Li, W.; Zia, T.; Zomaya, A.Y. A dimension reduction model and classifier for anomaly-based intrusion detection in internet of things. In Proceedings of the 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), Orlando, FL, USA, 6–10 November 2017; pp. 836–843.
170. Liu, H.; Lang, B. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. *Appl. Sci.* **2019**, *9*, 4396. [CrossRef]
171. Buczak, A.L.; Guven, E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 1153–1176. [CrossRef]
172. Mikolajczyk, K.; Leibe, B.; Schiele, B. Multiple object class detection with a generative model. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2006; Volume 1, pp. 26–33.
173. Lin, R.S.; Ross, D.A.; Lim, J.; Yang, M.H. Adaptive discriminative generative model and its applications. In *Advances in Neural Information Processing Systems*; 2005; pp. 801–808. Available online: https://proceedings.neurips.cc/paper/2004/file/f12f2b34a0c3174269c19e21c07dee68-Paper.pdf (accessed on 18 November 2021).
174. HaddadPajouh, H.; Dehghantanha, A.; Khayami, R.; Choo, K.K.R. A deep Recurrent Neural Network based approach for Internet of Things malware threat hunting. *Future Gener. Comput. Syst.* **2018**, *85*, 88–96. [CrossRef]
175. Saeed, A.; Ahmadinia, A.; Javed, A.; Larijani, H. Intelligent intrusion detection in low-power IoTs. *ACM Trans. Internet Technol. (TOIT)* **2016**, *16*, 27. [CrossRef]
176. Azmoodeh, A.; Dehghantanha, A.; Choo, K.K.R. Robust malware detection for internet of (battlefield) things devices using deep eigenspace learning. *IEEE Trans. Sustain. Comput.* **2018**, *4*, 88–95. [CrossRef]
177. Erfani, S.M.; Rajasegarar, S.; Karunasekera, S.; Leckie, C. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognit.* **2016**, *58*, 121–134. [CrossRef]
178. Razzak, I.; Zafar, K.; Imran, M.; Xu, G. Randomized nonlinear one-class support vector machines with bounded loss function to detect of outliers for large scale IoT data. *Future Gener. Comput. Syst.* **2020**, *112*, 715–723. [CrossRef]
179. La, Q.D.; Quek, T.Q.; Lee, J.; Jin, S.; Zhu, H. Deceptive attack and defense game in honeypot-enabled networks for the internet of things. *IEEE Internet Things J.* **2016**, *3*, 1025–1035. [CrossRef]
180. Anthi, E.; Williams, L.; Burnap, P. Pulse: An adaptive intrusion detection for the internet of things. In Proceedings of the Living in the Internet of Things: Cybersecurity of the IoT, London, UK, 28–29 March 2018.
181. Arrington, B.; Barnett, L.; Rufus, R.; Esterline, A. Behavioral modeling intrusion detection system (bmids) using internet of things (iot) behavior-based anomaly detection via immunity-inspired algorithms. In Proceedings of the 2016 25th International Conference on Computer Communication and Networks (ICCCN), Waikoloa, HI, USA, 1–4 August 2016; pp. 1–6.
182. Pajouh, H.H.; Javidan, R.; Khayami, R.; Ali, D.; Choo, K.K.R. A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks. *IEEE Trans. Emerg. Top. Comput.* **2016**, *7*, 314–323. [CrossRef]
183. Ghosh, P.; Mitra, R. Proposed GA-BFSS and logistic regression based intrusion detection system. In Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT), Hooghly, India, 7–8 February 2015; pp. 1–6.
184. Prokofiev, A.O.; Smirnova, Y.S.; Surov, V.A. A method to detect Internet of Things botnets In Proceedings of the 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), Moscow, Russia, 29 January–1 February 2018; pp. 105–108.
185. Azad, C.; Mehta, A.K.; Jha, V.K. Evolutionary Decision Tree-Based Intrusion Detection System. In *Proceedings of the Third International Conference on Microelectronics, Computing and Communication Systems*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 271–282.

186. Kumar, V.; Das, A.K.; Sinha, D. UIDS: A unified intrusion detection system for IoT environment. *Evol. Intell.* **2021**, *14*, 47–59. [CrossRef]
187. Amarasinghe, K.; Manic, M. Improving user trust on deep neural networks based intrusion detection systems. In Proceedings of the IECON 2018—44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, USA, 21–23 October 2018; pp. 3262–3268.
188. Blanco, R.; Cilla, J.J.; Malagón, P.; Penas, I.; Moya, J.M. Tuning cnn input layout for ids with genetic algorithms. In Proceedings of the International Conference on Hybrid Artificial Intelligence Systems, Oviedo, Spain, 20–22 June 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 197–209.
189. Mayhew, M.; Atighetchi, M.; Adler, A.; Greenstadt, R. Use of machine learning in big data analytics for insider threat detection. In Proceedings of the MILCOM 2015-2015 IEEE Military Communications Conference, Tampa, FL, USA, 26–28 October 2015; pp. 915–922.
190. Li, W.; Yi, P.; Wu, Y.; Pan, L.; Li, J. A new intrusion detection system based on KNN classification algorithm in wireless sensor network. *J. Electr. Comput. Eng.* **2014**, *2014*, 240217. [CrossRef]
191. Singh, R.R.; Gupta, N.; Kumar, S. To reduce the false alarm in intrusion detection system using self organizing map. *Int. J. Soft Comput. Eng. (IJSCE)* **2011**, *2231*, 2307.
192. Cárdenas, A.A.; Baras, J.S.; Seamon, K. A framework for the evaluation of intrusion detection systems. In Proceedings of the 2006 IEEE Symposium on Security and Privacy (S&P'06), Oakland, CA, USA, 21–24 May 2006; p. 15.
193. Gu, G.; Fogla, P.; Dagon, D.; Lee, W.; Skorić, B. Measuring intrusion detection capability: An information-theoretic approach. In Proceedings of the 2006 ACM Symposium on Information, computer and communications security, Taipei, Taiwan, 21–24 March 2006; pp. 90–101.
194. Fink, G.A.; Chappell, B.; Turner, T.; O'Donoghue, K. A metrics-based approach to intrusion detection system evaluation for distributed real-time systems. In Proceedings of the 16th International Parallel and Distributed Processing Symposium, Ft. Lauderdale, FL, USA, 15–19 April 2001; p. 8.
195. Kim, J.; Kim, J.; Thu, H.L.T.; Kim, H. Long short term memory recurrent neural network classifier for intrusion detection. In Proceedings of the 2016 International Conference on Platform Technology and Service (PlatCon), Jeju, Korea, 15–17 February 2016; pp. 1–5.
196. Ansere, J.A.; Han, G.; Bonsu, K.A.; Peng, Y. Energy Efficient Joint Power Allocation and User Selection Algorithm for Data Transmission in Internet of Things Networks. *IEEE Internet Things J.* **2020**, *7*, 8736–8747. [CrossRef]
197. Ansere, J.A.; Han, G.; Liu, L.; Peng, Y.; Kamal, M. Optimal Resource Allocation in Energy Efficient Internet of Things Networks with Imperfect CSI. *IEEE Internet Things J.* **2020**, *7*, 5401–5411. [CrossRef]
198. Tsai, C.F.; Lin, C.Y. A triangle area based nearest neighbors approach to intrusion detection. *Pattern Recognit.* **2010**, *43*, 222–229. [CrossRef]
199. IDS 2012 Datasets Research Canadian Institute for Cybersecurity UNB. Available online: https://www.unb.ca/cic/datasets/ids.html (accessed on 22 April 2020).
200. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the International Conference on Information Systems Security and Privacy, Funchal, Madeira, 22–24 January 2018; pp. 108–116. Available online: https://www.scitepress.org/Papers/2018/66398/66398.pdf (accessed on 17 October 2021).
201. IDS 2018 Datasets Research Canadian Institute for Cybersecurity UNB. Available online: https://www.unb.ca/cic/datasets/ids-2018.html (accessed on 22 April 2020).
202. Ullah, I.; Mahmoud, Q.H. A two-level hybrid model for anomalous activity detection in IoT networks. In Proceedings of the 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 11–14 January 2019; pp. 1–6. [CrossRef]
203. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6.
204. Moustafa, N.; Slay, J. The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Inf. Secur. J. Glob. Perspect.* **2016**, *25*, 18–31. [CrossRef]
205. Kang, H.; Ahn, D.H.; Lee, G.M.; Yoo, J.; Park, K.H.; Kim, H.K. IoT Network Intrusion Dataset. IEEE Dataport. 2019. Available online: https://ieee-dataport.org/ieee-dataport-dataset-upload-contest-entries (accessed on 10 January 2022).
206. Ullah, I.; Mahmoud, Q.H. A scheme for generating a dataset for anomalous activity detection in iot networks. In Proceedings of the Canadian Conference on Artificial Intelligence, Ottawa, ON, Canada, 13–15 May 2020; Springer: Berlin/Heidelberg, Germany, 2020, pp. 508–520.
207. Alsaedi, A.; Moustafa, N.; Tari, Z.; Mahmood, A.; Anwar, A. TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. *IEEE Access* **2020**, *8*, 165130–165150. [CrossRef]
208. Pinto, R.; Gonçalves, G.; Tovar, E.; Delsing, J. Attack detection in cyber-physical production systems using the deterministic dendritic cell algorithm. In Proceedings of the 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 8–11 September 2020; Volume 1, pp. 1552–1559.

209. Sousa, B.; Cruz, T.; Arieiro, M.; Pereira, V. An ELEGANT dataset with Denial of Service and Man in The Middle attacks. *arXiv* **2021**, arXiv:2103.09380.
210. Sambangi, S.; Gondi, L.; Aljawarneh, S.; Annaluri, S.R. SDN DDOS Attack Image Dataset. 2021. Available online: https://ieee-dataport.org/documents/sdn-ddos-attack-image-dataset (accessed on 10 January 2022)
211. Hussain, F.; Abbas, S.G.; Husnain, M.; Fayyaz, U.U.; Shahzad, F.; Shah, G.A. IoT DoS and DDoS attack detection using ResNet. In Proceedings of the 2020 IEEE 23rd International Multitopic Conference (INMIC), Bahawalpur, Pakistan, 5–7 November 2020; pp. 1–6.
212. LEE, Y. Development of Industrial IoT System for Anomaly Detection in Smart Factory. 2020. Available online: https://ieee-dataport.org/documents/development-industrial-iot-system-anomaly-detection-smart-factory (accessed on 10 January 2022).
213. Hindy, H.; Tachtatzis, C.; Atkinson, R.; Bayne, E.; Bellekens, X. Mqtt-iot-ids2020: Mqtt Internet of Things Intrusion Detection Dataset. IEEE Dataport. 2020. Available online: https://ieee-dataport.org/open-access/mqtt-iot-ids2020-mqtt-internet-things-intrusion-detection-dataset (accessed on 11 January 2022)
214. Ferrag, M.A.; Friha, O.; Hamouda, D.; Maglaras, L.; Janicke, H. Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning. *IEEE Access* **2022**, 40281–40306. [CrossRef]
215. Al-Hawawreh, M.; Sitnikova, E.; Aboutorab, N. X-IIoTID: A connectivity-and device-agnostic intrusion dataset for industrial Internet of Things. *IEEE Internet Things J.* **2021**, *9*, 3962–3977. [CrossRef]
216. Trajanovski, T.; Zhang, N. IoT-BDA Botnet Analysis Dataset. *IEEE Internet Things J.* **2021**, *9*. [CrossRef]
217. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Milano, Italy, 7–10 September 2009; pp. 1–6.
218. Hussain, F.; Abbas, S.G.; Husnain, M.; Fayyaz, U.U.; Shahzad, F.; Shah, G.A. IoT DoS and DDoS Attack Dataset. 2021. Available online: https://ieee-dataport.org/documents/iot-dos-and-ddos-attack-dataset (accessed on 11 January 2022). [CrossRef]
219. Hussain, F.; Abbas, S.G.; Shah, G.A.; Pires, I.M.; Fayyaz, U.U.; Shahzad, F.; Garcia, N.M.; Zdravevski, E. A Framework for Malicious Traffic Detection in IoT Healthcare Environment. *Sensors* **2021**, *21*, 3025. [CrossRef]
220. Kalupahana Liyanage, K.S.; Divakaran, D.M.; Singh, R.P.; Gurusamy, M. NSS Mirai Dataset. 2020. Available online: https://ieee-dataport.org/documents/nss-mirai-dataset (accessed on 10 January 2022)
221. Zolanvari, M. WUSTL-IIOT-2021. 2021. Available online: https://ieee-dataport.org/documents/wustl-iiot-2021, (accessed on 10 January 2022)
222. Moustafa, N. The Bot-IoT Dataset. 2019. Available online: https://ieee-dataport.org/documents/bot-iot-dataset (accessed on 11 January 2022)