electronics

MDPI

Review

A Review of Intrusion Detection Systems Using Machine and Deep Learning in Internet of Things: Challenges, Solutions and Future Directions

Javed Asharf [1], Nour Moustafa [2,*], Hasnat Khurshid [1], Essam Debie [2], Waqas Haider [2] and Abdul Wahab [3]

[1] Military College of Signals, National University of Sciences and Technology (NUST), H-12, Islamabad 44000, Pakistan; javed.ashraf@mcs.edu.pk (J.A.); hasnat@mcs.edu.pk (H.K.)
[2] School of Engineering and Information Technology, University of New South Wales at the Australian Defence Force Academy, Canberra 2610, Australia; e.debie@unsw.edu.au (E.D.); w.haider@adfa.edu.au (W.H.)
[3] Department of computer Science, Riphah University, Islamabad 44000, Pakistan; abdulwahab86@gmail.com
* Correspondence: nour.moustafa@unsw.edu.au

Received: 25 May 2020; Accepted: 12 July 2020; Published: 20 July 2020

Abstract: The Internet of Things (IoT) is poised to impact several aspects of our lives with its fast proliferation in many areas such as wearable devices, smart sensors and home appliances. IoT devices are characterized by their connectivity, pervasiveness and limited processing capability. The number of IoT devices in the world is increasing rapidly and it is expected that there will be 50 billion devices connected to the Internet by the end of the year 2020. This explosion of IoT devices, which can be easily increased compared to desktop computers, has led to a spike in IoT-based cyber-attack incidents. To alleviate this challenge, there is a requirement to develop new techniques for detecting attacks initiated from compromised IoT devices. Machine and deep learning techniques are in this context the most appropriate detective control approach against attacks generated from IoT devices. This study aims to present a comprehensive review of IoT systems-related technologies, protocols, architecture and threats emerging from compromised IoT devices along with providing an overview of intrusion detection models. This work also covers the analysis of various machine learning and deep learning-based techniques suitable to detect IoT systems related to cyber-attacks.

Keywords: IoT security; IoT protocols; intrusion detection system; machine learning; deep learning; cyber-attacks

1. Introduction

The recent development in communications and information technologies, such as the Internet of Things (IoT), has extraordinarily surpassed the traditional sensing of nearby environments. IoT technologies have facilitated the development of systems that can improve life quality. IoT is one of the fastest-growing technologies in computing, with an estimated 50 billion devices by the end of 2020 [1]. It has been estimated that, by the year 2025, the IoT and related applications have a potential economic impact of $3.9 trillion to $11.1 trillion per year [2]. The IoT devices can become smart objects by taking advantage of its core technologies like communication technologies, pervasive and ubiquitous computing, embedded devices, Internet protocols, sensor networks, and Artificial Intelligence (AI)-based applications [3].

The ubiquitous interconnection of physically distributed IoT devices extends the computation and communication to other IoT devices with different specifications [4]. Multiple types of sensors, embedded in these devices, enable them to gather real-time data from the physical devices remotely.

The collected data from the devices allows us to make intelligent decision systems as well as effectively managing IoT environments. However, connecting the commonly used real-world devices to the Internet also raises concerns about cybersecurity threats [5,6]. Therefore there is a requirement to design and develop intelligent security solutions for the protection of IoT devices and against attacks generated from compromised IoT devices.

1.1. Motivation

While IoT technologies play a vital part in improving real-life smart systems, like smart cities, smart homes, smart healthcare, the large scale and ubiquitous nature of IoT systems has introduced new security challenges [5–7]. Furthermore, since IoT devices generally work in an unattended environment, an attacker may physically access these devices with malicious intent [8,9]. Also, because IoT devices are connected usually over wireless networks, eavesdropping can be used to access private information from a communication channel [10,11]. On top of these security challenges, IoT devices cannot afford the implementation of advanced security features because of their restricted energy and computation resources. Due to the interconnected and interdependent settings of the IoT, new attack surfaces are emerging very regularly [12,13]. Thus, IoT systems are more vulnerable as compared to traditional computing systems. This necessitates research in specific detective and preventive techniques for IoT systems to protect against IoT devices based threats.

For protecting IoT systems against cyber threats, another line of defense should be developed in IoT networks. Intrusion Detection Systems (IDSs) fulfill this purpose [14,15]. Various surveys have attempted to describe machine learning-based IDSs for protection against IoT networks or compromised IoT devices. The surveys cover research work on IDSs for cloud-based IoT systems [16], Wireless sensor networks [17–19], cyber-physical systems [20], and mobile ad hoc networks (MANETs) [21–23]. However, traditional IDS methods are less effective or insufficient for the security of IoT systems because of their peculiar characteristics mentioned above, in particular, limited energy, ubiquitous, heterogeneity, limited bandwidth capacity and global connectivity. Machine Learning (ML) and Deep Learning (DL) based techniques have recently gained credibility in a successful application for the detection of network attacks including IoT networks. This is because ML/DL based methods can capture benign and anomalous behavior in IoT environments. IoT devices and network traffic can be captured and investigated to learn normal patterns. Any deviation from these normal learned patterns can be used to detect anomalous behavior. Furthermore, ML/DL based methods have been tested to predict new or zero-day attacks. Hence, ML/DL based algorithms provide robust security protocols for designing the security of IoT devices and networks.

Various surveys have discussed different techniques for designing IDS for IoT systems, but most of the aforementioned surveys did not address the implementation of ML or DL techniques as detection mechanisms in IoT networks and their lightweight devices in a comprehensive manner. Some of these studies published in [24–29] revealed that the focus was on studying the issues in IoT security generally and their classification in different layers related to applications, network, encryption and authentication, and access controls. A comprehensive study covering a detailed review of ML and DL based techniques for IDSs in IoT networks still needs further systematic analysis and investigation, which is a major focus of this study.

1.2. Scope of This Survey

This survey includes six important areas related to IDSs for IoT systems and networks: (1) IoT architectures and technologies; (2) IoT threats and attack types; (3) IDS architectures and their design; (4) an explanation of ML and DL techniques applied in the design of IDSs; (5) a description of various datasets available to researchers for evaluation of their proposed IDS; and (6) future research challenges and directions.

1.3. Main Contribution

In this paper, a detailed review of network threats from IoT networks and their devices with corresponding ML and DL based attack detection techniques is presented. Table 1 summarizes a comparison of our survey with the other surveys conducted on IDSs in IoT networks. As described in the table, this survey covers all important aspects on the subject of ML and DL based techniques used for IDS in IoT networks and their systems. The table also shows that other surveys partially cover some of the aspects and there is no single paper that explains all the aspects. The key contributions of this survey are described as follows:

- Discussion of IoT architectures and IoT Protocols, covering their technologies, frequency bands, and data rates.
- Explanation of vulnerabilities, threat dimensions and attack surfaces of IoT systems, including attack types related to IoT protocols, which are discussed in detail.
- Review of ML- and DL-based IDSs, involving their design choices, pros, cons and detection methods, which are covered in detail.
- Discussion of the datasets available for network and IoT security-related research, covering the advantages and limitations of each enumerated with details.
- Explanation of the applications of ML and DL techniques for developing IDSs in IoT networks and their systems.
- Presentation of the current research challenges and their future directions for research in this field.

The organization of the paper is presented as follows. In Section 2, recent studies conducted related to the anomaly and intrusion detection in IoT networks are discussed. In Section 3, an overview of IoT systems is presented covering IoT architecture and reference models and IoT protocols. Section 4 describes various attacks and threats against IoT systems. Following this, Section 5 discusses IDS architecture, its design choices and various detection methods, including their ML and DL techniques described in Sections 6 and 7, respectively. Section 8 describes briefly the datasets that are available and used for testing IDS. Finally, the future challenges and paper's conclusion are provided in Sections 9 and 10, respectively.

2. Current Reviews

Various survey studies have been carried out in the field of IoT security by describing vulnerabilities in IoT systems. However, most of the existing studies on IoT security have not mainly focused on the applications of ML/DL techniques for IoT security. Table 1 summarizes a comparison of our survey with the other surveys conducted on IDSs in IoT networks. The comparison discusses the contributions of each survey related to the design of IoT-based IDSs.

Table 1. A comparison of this survey with others in terms of developing IoT-based Intrusion Detection Systems (IDSs).

Survey Ref	IoT IDS Aspects						
	IoT Architecture	**IoT Protocols**	**IoT Threats**	**IoT IDS Design Choices**	**IoT IDS-ML Techniques**	**IoT IDS-DL Techniques**	**IoT Datasets**
[30]	✓	✓	✓	✓	×	×	×
[16]	×	×	✓	✓	✓	×	×
[31]	×	×	✓	×	×	×	×
[19]	×	×	×	✓	×	×	×
[21]	×	×	✓	✓	×	×	×
[22]	×	×	✓	✓	×	×	×

Table 1. *Cont.*

Survey Ref	IoT IDS Aspects						
	IoT Architecture	**IoT Protocols**	**IoT Threats**	**IoT IDS Design Choices**	**IoT IDS-ML Techniques**	**IoT IDS-DL Techniques**	**IoT Datasets**
[23]	×	×	✓	✓	✓	✓	×
[32]	×	✓	✓	×	×	×	×
[33]	×	✓	✓	✓	×	×	×
[34]	×	×	✓	×	✓	×	×
[35]	×	×	×	×	✓	✓	✓
[36]	×	×	✓	×	✓	✓	✓
[37]	✓	×	×	✓	✓	✓	✓
[38]	×	×	✓	✓	×	×	×
This Survey	✓	✓	✓	✓	✓	✓	✓

In [32], the authors studied the challenges of IoT security at the communication layer. A study in [33] focused on reviewing IDSs for IoT networks. The work in [34] covered a brief discussion of the ML technique's relevance in the context of IoT security and privacy. Moreover, they identified limited bandwidth, computation power and lack of adequate storage as bottlenecks in any implementation of ML-based security solutions for IoT networks. There are other studies [35,36], which discussed the feasibility of both ML and data mining techniques to detect intrusions in IoT networks by implementing these techniques in IDSs either through detecting anomalies or classification of traffic. In [21], the authors highlighted differentials between IDSs running over wired networks and those running over wireless infrastructure, especially IoT networks. Due to fundamental architectural variations, the application of ML techniques in IoT IDSs needs specific treatment related to the type of attacks, underlying protocols (both in communications and networks), and application layer.

Another study published in [22] discussed the implementation of IDS in the context of MANETs. The authors described that there are three different types of IDS architectures feasible in MANETs. First architecture can be a layered architecture organized in multiple hierarchical layers. Second architecture can be a flat one for deploying in a distributed and cooperative environment. While the third one can be a hybrid of both using mobile agents. Another study [23] discussed various Intrusion Detection algorithms related to IDS implementation in MANET. According to the authors, these IDS algorithms can be categorized in various categories based on the underlying principle used for the detection of an attack. These principles can either be a rule, statistics, heuristics, signature, state, reputation score, or route used. These techniques were later classified further as anomaly detection, misuse, signature-based, or hybrid techniques. There were other classification criteria proposed by the authors [23] like real-time/offline, attack types and effectiveness of detection (scalability, reliability, timeliness, etc.).

Another survey presented in [30], the authors explained a classification of IDS for Wireless Sensor Networks (WSN) based on the deployment model of the IDS agent. The deployment model can be either distributed, central, or a hybrid mode, which is suggested as the best-suited model for WSNs. A similar study [31] carried out a classification of WSNs based on IDS using the criteria of detection type used by the IDS. The classes identified included anomaly detection, misuse detection and detection based on specifications. Another aspect of cloud-based IoT environment was discussed in [16], where the authors studied and classified various cloud-based IDSs affecting Confidentiality, Integrity, and Availability (CIA) of cloud computing-based IoT networks. They explained Hypervisor-based IDS, Host-based IDS (HIDS), Network-based IDS (NIDS) and Distributed IDS. In [30], the authors presented a survey on IoT IDS with a focus on an IDS architecture. The survey covered existing IoT protocols, standards and technologies, IoT security threats, detection types and concludes by suggesting proposed IoT IDS architecture.

The authors in [39], proposed a novel multi-stage anomaly detection technique based on Boruta Firefly Aided Partitioning Density-Based Spatial Clustering of Applications with Noise (BFA-PDBSCAN). The authors claimed that their proposed technique produced better results in comparison to the related techniques of Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN). In [40], the authors proposed a hybrid data processing model for network anomaly detection that utilizes Grey Wolf Optimization (GWO) and Convolutional Neural Network (CNN) techniques. The authors stated that their model achieved better accuracy and detection rate in comparison to the other state-of-the-art IDSs. In [41], an anomaly detection method based on a deep autoencoder was used to detect attacks of IoT botnets. The method comprises extracting statistical features from behavioral snapshots of normal IoT device traffic sequences and training of a DL based autoencoder on the extracted features. The reconstruction error for traffic observations is then compared with a threshold to classify them as normal or anomalous. The authors evaluated the proposed detection method on the BASHLITE and Mirai botnets dataset generated using commercial IoT devices. In a recent survey paper published in [37], learning-based NIDSs for IoT systems were discussed in an overview of ML-based NIDSs for IoT systems.

3. IoT System Environment

The adoption of IoT throughout real-world applications, such as home automation, industrial automation and city automation, resulted in a plethora of micro computation devices and energy-efficient communication technologies, specifications and protocols. IoT systems have been widely employed in applications of military, agriculture, power systems, education and commerce. Diverse areas of applications resulted in the realization of various devices, communication standards and protocols. The IoT system paradigms illustrate its various applications, where the access network technology is presented in Figure 1 that shows a loose clustering of various IoT communication technologies and protocols to the corresponding network.

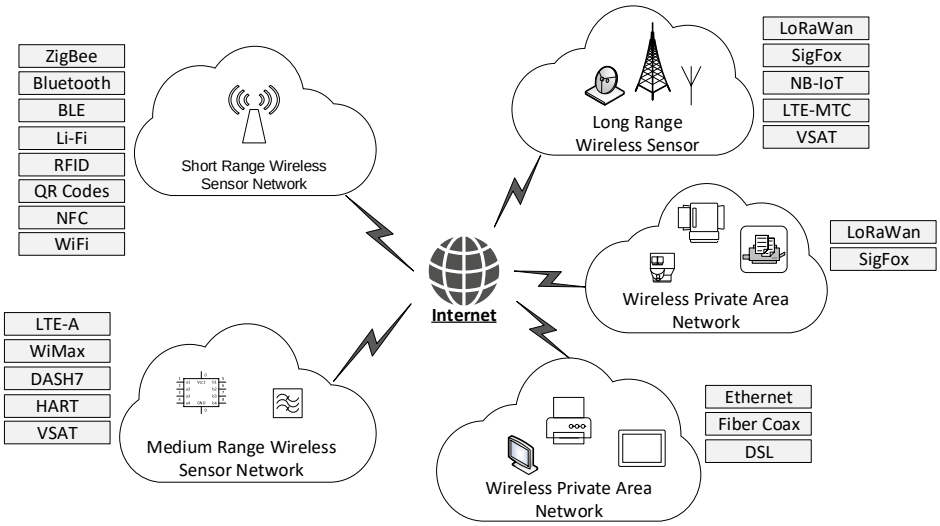

Figure 1. IoT system environment—applications and related access networks and protocols.

3.1. IoT Architecture

The IoT architecture consists of physical objects integrated into a communication network and supported by computational equipment to deliver smart services to users. The IoT system should be capable of connecting billions of heterogeneous devices through the Internet, so there is a need for a layered and flexible architecture. There are numerous architectures and reference models proposed by various authors and organizations but those have not yet converged to a formally recognized reference model [3–7,42–44]. The most common architectures and reference models (the terms "architecture" and "reference model" used interchangeably by the authors) are explained as follows:

- **A 3-layer architecture.** The most common and basic model is a 3-layer architecture comprising of the perception, network and application layers [3,4,43], as depicted in Figure 2, the perception layer is also called 'the device layer' that includes physical devices and sensors. The network layer is also named 'the transmission layer', which should securely transmit the telemetry data of sensors to processing and data analytical systems. The application layer offers global management of applications using the systems at the network layer.
- **International Telecommunication Union (ITU) recommended Reference Model for IoT.** ITU recommends a reference model for IoT that comprises four layers, along with security and management capabilities linked to the layers [45]. The layers are as follows: device layer, network layer, application support layer, service support and application layer, as shown in Figure 3.
- **IoT-A Architectural Reference Model proposed by the European Commission (FP7).** The European Commission within the Seventh Framework Program (FP7) supported the project IoT-A proposed by Martin Bauer et al. [6]. The IoT-A model attempts to design an architecture that could meet the requirements of the industry and researchers. It offers high-level architectural perspectives and views for building IoT systems. The architecture comprehensively describes the structuring and modeling of IoT business process management, IoT services, cross-service organization and virtual entities, information and functional viewpoints, in an abstract way [46]. Amongst these various views, a functional view of IoT architecture is depicted in Figure 4.
- **An IoT Reference Architecture developed by Web Service Oxygen (WSO2).** WSO2, an open-source technology provider, has proposed an Architectural Reference Model based on its skills in the IoT solutions development. Figure 5 depicts the WSO2 recommended architecture. It consists of five layers: (1) Client/external communications—Web/Portal, Dashboard, Application Programming Interface (APIs), (2) Event processing and analytics (including data storage), (3) Aggregation/bus layer—Enterprise Service Bus (ESB) and message broker, (4) Relevant transports—XMPP/CoAP/AMQP/HTTP/MQTT, etc. and (5) Devices [47]. The model includes the cross-cutting layers that have (1) a device manager, and (2) an identity and access management system.
- **An IoT Reference Architecture suggested by Cisco:** Cisco introduced a seven-layered IoT reference model [48]. The model and its levels are illustrated in Figure 6. The authors described that control information flows from level 7 to level 1 in a control pattern. The flow of information is the reverse in a monitoring pattern and it is bidirectional in most systems.

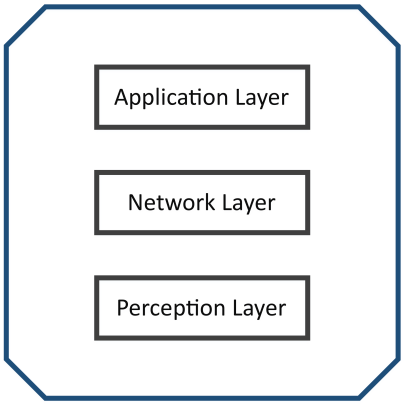

Figure 2. Illustration of basic IoT architecture.

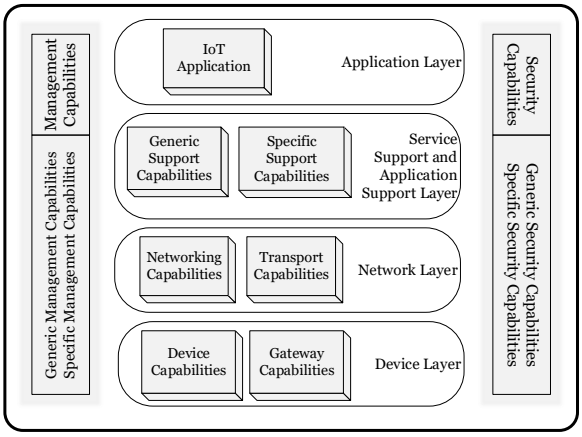

Figure 3. ITU-T reference model [45].

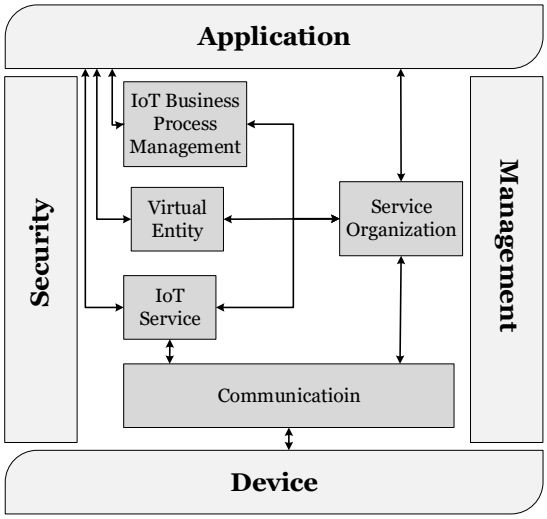

Figure 4. IoT-A functional view [6,46].

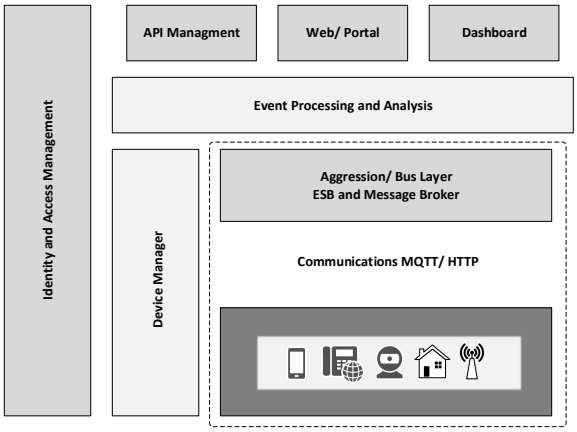

Figure 5. Web Service Oxygen (WSO2) IoT reference architecture [47].

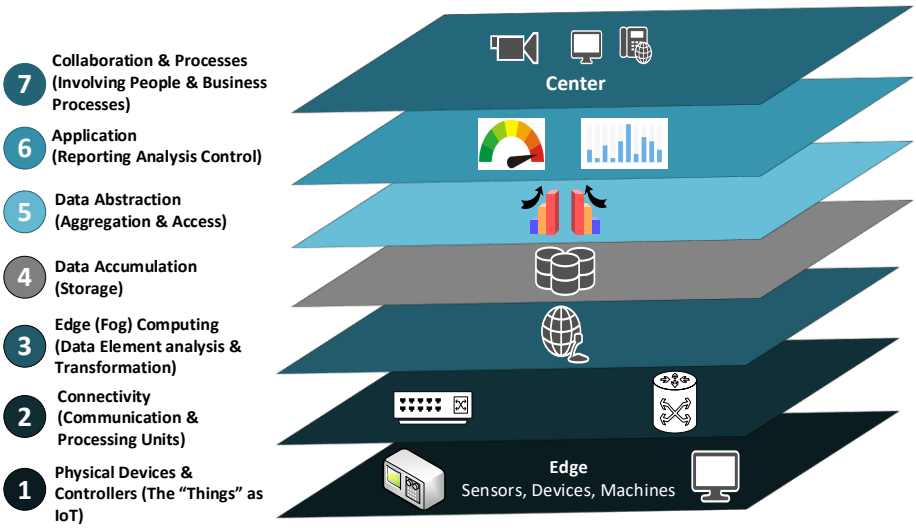

Figure 6. Cisco IoT reference model [48].

3.2. IoT Protocols

Several protocols and specifications inherited from the TCP/IP model, some technologies are specifically developed for IoT systems. IEEE 802.15.4 (transmission and communication specification standards) is not alone in the paradigm of IoT specific technologies and standards. In Table 2, a description of the IoT technologies with respective frequency bands and supported data rates and area coverage.

Table 2. IoT enabling technologies, their frequency bands and data rates.

Technology	Frequency Bands	Data Rate	Physical Coverage
WiFi 802.11	2.5 GHz, 5 GHz	<1 Gbps	up to 50 m
WiFi HaLow	900 MHz	0.3–234 Mbps	up to 1 km
White-Fi	54–790 MHz	26.7–568.9 Mbps	up to 100 m
Bluetooth	2.4 GHz	100 Kbps	up to 100 m
Bluetooth LE	2.4 GHz	<1 Gbps	up to 50 m
Z-Wave	686 MHz, 908 MHz, 2.4 GHz	40 K	up to 100 m
ZigBee	868 MHz, 915 MHz, 2.4 GHz	20 kbps to 250 kbps	10–75 m
ISA100.11a	868 MHz, 915 MHz, 2.4 GHz	20 kbps to 250 kbps	up to 600 m
MiWi	868 MHz, 915 MHz, 2.4 GHz	20 kbps to 250 kbps	20–50 m
Thread	868 MHz, 915 MHz, 2.4 GHz	20 kbps to 250 kbps	up to 100 m
WirelessHART	868 MHz, 915 MHz, 2.4 GHz	20 kbps to 250 kbps	30–100 m
LTE-A	Cellular bands	1 G (up), 500 M (down)	up to 50 km
GSM	Cellular bands	150 Mbps	up to 50 km
LTE-Cat M	Cellular bands	up to 1 Mbps	15 km
NB-IoT	Cellular bands	<180 kbps	15 km
LoRaWAN	169/433/868/780/915 MHz ISM	300 bit to 100 kbit/s	2.5–15 km
NFC	13.56 MHz	up to 424 kbps	<20 cm
DASH7	433/868/915 MHz ISM/SRD	9.6–166.667 kbit/s	up to 5 km
nWave	Sub-1 GHz ISM	100 bit/s	10–30 km
SigFox	868/902 MHz ISM	100 bit/s	12–30 km

4. IoT-Based Threats and Attacks

IoT systems suffer from various security risks as compared to conventional computing systems due to several reasons [15,47]. First, IoT systems are highly diverse with regards to devices, platforms, communication means and protocols. Second, IoT systems comprise "things" not planned to be connected to the Internet, where control devices are used to link physical systems. Third, there are no well-defined boundaries in IoT systems, which regularly change due to the mobility of users and devices. Forth, IoT systems, or part of them, would be physically insecure. Last but not least, due to the limited energy of IoT devices, it is usually very hard to deploy advanced security techniques and tools on IoT devices.

An IoT network often contains hundreds of nodes with assigned functions ranging from sensing of light, temperature and noise to associated control systems to regulate lighting and heating, ventilation, and air conditioning (HVAC) systems, etc. All these sensors and control systems communicate through different network protocols like Bluetooth, WiFi, ZigBee, etc. An IoT gateway is used to connect these devices to the Internet. Being composed of layers of standards, services and technologies, the IoT environment has privacy and security concerns at each of these layers. While it seems that the IoT environment has similar security concerns to the Internet, cloud and mobile communication networks, there are distinct characteristics that set IoT environments, along with the applications of contemporary security controls [10]. These can share data, computing capacity limitation and a large number of networked IoT devices.

One instance of the susceptibility of IoT devices to attacks was demonstrated in September 2016, where an IoT botnet built from the Mirai malware—possibly the largest botnet on record—was responsible for a 620 Gbps attack directed towards Brian Krebs's security blog [11]. Mirai followed a simple strategy, where it tried a list of 62 common user credentials to get access to digital video recorders, home routers and network-enabled cameras, which generally had fewer defenses than other

IoT devices. Later, in the same month, the French webhost OVH (On Vous Héberge) was attacked by the Mirai-based attack, which broke the record for the largest recorded distributed denial of service (DDoS) attack peaking at 1.1 Tbps [12]. The attack was made possible due to default and weak security configurations. Similarly, in [49], the authors described the relative ease of compromising various IoT devices, due to flaws in protocol implementations.

The rapid proliferation of IoT based devices is likely to make such networks susceptible to attacks against privacy and security aspects. In [13], the authors identified various security issues in IoT networks built with commercially available IoT devices like sensors. One example cites a smart watering system that is capable of measuring environmental variables like temperature and humidity, etc. An actuator module was employed for functionality implementation with a web-based user interface. The system was built on an Arduino Uno. The authors described the exposure of such network to spoofing attacks through a software-enabled access point (SoftAP), where an attacker managed all IoT devices in a network to shut down for a while as the SoftAP broadcasts de-authentication packets.

Due to the limited processing capabilities of IoT devices, the hacker made all IoT devices vulnerable in the network to connect to the SoftAP as it appeared to have a stronger signal than the actual access point (AP) with the same service set identifier (SSID). This allowed the compromise of all network communications to eavesdropping and man in the middle (MiTM) attacks. Such attack scenarios built a case for the deployment of IDSs in IoT networks to discover vulnerabilities of IoT devices. The idea of IoT revolves around the intelligent integration of a real physical environment with the Internet to enable interactivity. For this reason, IoT environments have interconnections and dependencies with multiple heterogeneous environments. This exposes each IoT system to cyber threats from each connected environment [50,51]. IoT environments face threats from multiple dimensions both from physical and virtual domains. Figure 7 illustrates multiple threat dimensions of an IoT environment that would be exploited.

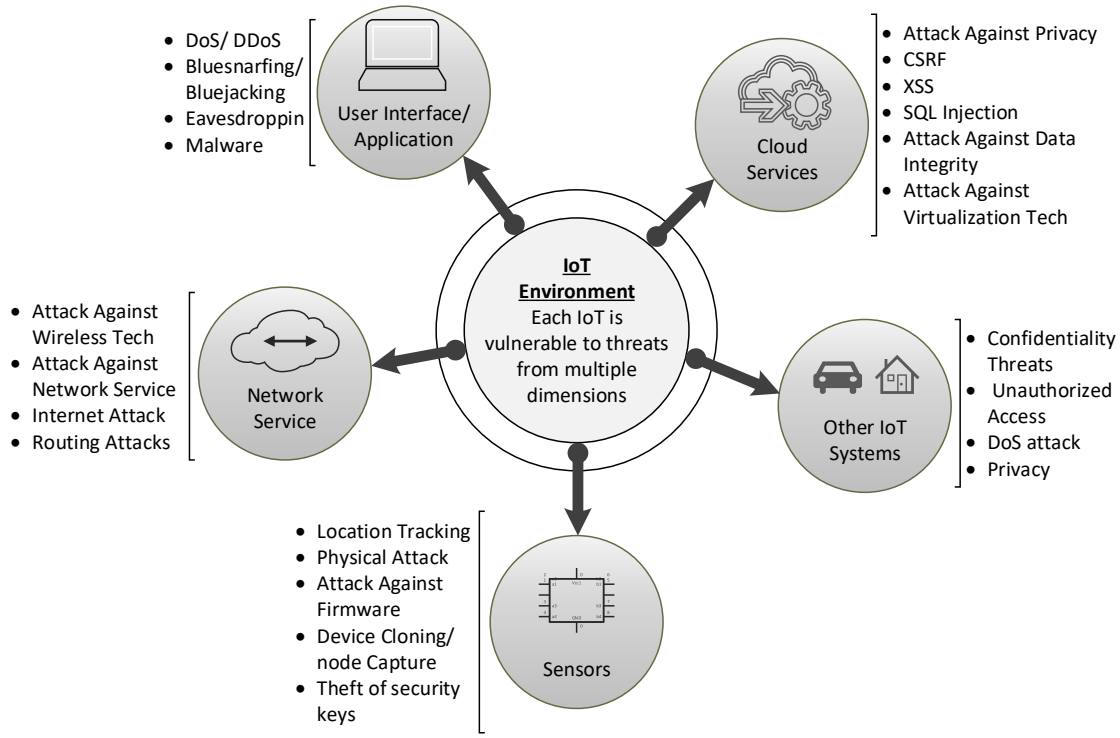

Figure 7. IoT environment threat dimensions.

Though IoT Security threats can be broadly divided into cyber and physical domains, our survey is mainly concerned with cyber threats, which can take the form of either active or passive attacks. Passive Attacks are characterized by a lack of any alteration to information or its flow, thereby only compromising the confidentiality and privacy of communications. In some cases, a passive attack can enable location tracking of IoT devices [52–54]. Active Attacks involve active alteration and modification of information and its flow, but are not limited to device settings, control messages and software components.

One active attack is when the IoT system is used as a vector to launch massive DDoS against Internet systems. IoT systems are a suitable vector for these attacks because of their large numbers and comparative ease of their compromise, due to poor security practices and weak defense mechanisms. Mirai can be used as an example of a botnet attack through for compromising IoT systems [11,55,56]. IoT systems face many threat dimensions from multiple directions, including user interface, cloud services, other interconnected IoT systems associated to sensors and network services [12], as shown in Figure 7. A discussion of these dimensions is presented in the following subsections.

4.1. User Interface

Most use cases of IoT systems involve the provision of services to users by IoT systems through some sort of a user interface (mobile, desktop or web application). The case of smart home appliances can be controlled by users through mobile applications. The rapid proliferation of smartphones has provided malicious actors to disguise malicious applications and malware as benign utility mobile applications and publish them through applications to store without being detected [57,58]. Also, smartphones can sometimes be hacked through platform vulnerabilities of these devices like Android vulnerabilities. This leads to exposing all information stored on the phone with the possibility of malware compromise. Eavesdropping, location tracking, Denial of Service (DoS)/DDoS, bluejacking and bluesnarfing are attacks enabled through user interface platforms [59–61].

4.2. Cloud Services

Though Cloud services and IoT systems lie at two ends of the resource availability spectrum, the two can complement each other to produce an excellent blend of technologies. Cloud services are characterized by ubiquitous access to computing power and storage, etc., which can offset the resource limitations of IoT systems [62]. The potential of IoT systems can be maximized through integrated use with cloud services to conserve energy and provide all types of services without being constrained by storage and processing power limitations [63]. Likewise, cloud services can benefit from large deployments of IoT systems through integrated applications [64]. Such a distributed architecture opens up vulnerable points for many attacks at multiple layers, as explained below.

- **Authorization Attacks.** Through the exploitation of vulnerabilities in data security mechanisms, an attacker may be able to gain unauthorized access to information on both cloud and IoT systems.
- **Integrity Attacks.** Such attacks enable an attacker to compromise the integrity of data through spoofing and bypass the authorization controls to gain direct access to databases.
- **Compromise of Visualization platform.** A vulnerability in the virtualization platform can be exploited by an attacker to bypass security and isolation controls between the host and the guest operating system (OS), resulting in privilege escalation and pivoting attacks [65].
- **Confidentiality Attacks.** IoT systems, like wearable devices, are used to monitor health-related data of highly confidential nature. Similarly, smart home devices capture sensitive private data of the users. Privacy and confidentiality concerns overshadow the advantages of cloud services. Moreover, multi-tenancy and geographical location of cloud services pose a serious threat to the confidentiality of data through privilege escalation and hacking [66].

4.3. Connections of Multiple IoT Systems

Various IoT systems are designed to work autonomously and interact with other IoT systems, such as sensors and actuators of smart cars and smart homes, without requiring human involvement. Such an interaction is aimed at achieving an autonomous and collaborative functionality. Smart cars and smart homes can communicate with each other and provide interdependent services and functions. For instance, [67] described such a scenario where sensing increased temperature by a temperature sensor, coupled with sensing of unplugging of a smart plug, the windows of the room are automatically opened. The window opening actuator would be reachable for an attack as it may manipulate the temperature sensing device through its interface and in turn that compromises the actuator [67]. This example highlights the fact that the weakest part of interdependent IoT systems can compromise other parts as well.

A large number of interconnected devices in IoT systems increases the vulnerability and also the impact of any attack, where one compromised device can lead to the compromise of billions of devices. Such a scenario can impact any externally connected networks and systems also. One study [68] demonstrated that an experimental malware attack against Philips Hue smart lamp was so successful that it compromised all such lamps in the network, despite the presence of reliable cryptographic authentication mechanisms against malicious firmware updates. Similar attacks could provide the control of lights of an entire city or their use in DDoS against outside targets [68].

Various types of sensors are an essential part of IoT systems like GPS, Radio-Frequency Identification (RFID), temperature gauge and IP cameras. This also includes sensors and actuators embedded in autonomous vehicles and the internet of vehicles (IOVs). These physical devices are vulnerable to physical attacks and manipulation by malicious actors. Another component of IoT systems susceptible to such physical attacks is the actuator part, which performs some function based on readings of sensor devices. Both actuators and sensors would be subjected to DoS attacks through flooding, eavesdropping, location tracking, cloning and spoofing attacks [69–71].

An IoT system consists of several interconnected devices using either wireless or wired networks. A large network linked to devices would have weak security profiles, where sensors and actuators are vulnerable to a multitude of attacks. WSNs provide information to external entities without any restriction. When they are integrated with conventional networks services, they cause regression in the security of conventional networks [72,73].

4.4. Protocols Level Attacks

IoT systems are different from traditional Internet protocols, which require lightweight protocols to address issues of limited energy, data rate and computing power. A detailed description of IoT protocols based attacks can be found in [74]. Attacks of IoT technologies are presented with threat types in Table 3.

Table 3. Summary of attacks against main IoT technologies (C: Confidentiality, I: Integrity, A: Availability).

Technology/Protocol	Attack	Threat	Threat Category
RFID	Tag Disable	Jamming	A
	Tag Modification	Unauthorized Access and modification of critical information	C, I
	Cloning Tags	Counterfeiting and spoofing	C, A
	Reverse Engineering	Counterfeiting and spoofing	C, A
	Eavesdropping	Unauthorized Access of critical information	C
	Snooping	Unauthorized Access of identity and data	C
	Skimming	Imitates the original RFID tag	C, A

Table 3. *Cont.*

Technology/Protocol	Attack	Threat	Threat Category
	Replay Attack	Deceiving readers	C, A
	Relay Attacks	man-in-the-middle	I, A
	EM Interference	Jamming	A
	Fake RFID tag queries	illicit Tracing and Tracking	C, A
	Cryptograph Decipher attack	Password Decoding	C, I
	blocker tag Attack	DoS Attack	A
ZigBee	Sniffing	sniffing the keys	C
	Replay attack	MiTM	C, I
	Killerbee Packer	Manipulation Attack Device Spoofing	C
	Killerbee - zbassocflood	Crash the device	A
	Eavesdropping	MiTM	C, I
	ZED Sabotage Attack	DoS	A
WiFi	FMS/KoreK/PTW/ARP Injection/Dictionary Attack	Key Retrieving Attacks	C
	ChopChop/ Fragmentation/Caffe Latte/Hirte	Keystream Retrieving Attacks	C, I
	Authentication related Attacks	DoS	A
	Association related Attack	DoS	A
	Flooding related attacks	DoS	A
	Honeypot	MiTM	C, I
	Evil Twin/Rogue AP	MiTM	C, I
Bluetooth	Bluebugging	Espionage	C, I
	Bluesnarfing	Espionage and DoS	C, A
	Sniffing Attacks	Interception	C
	Hijacking	DoS and spoofing	A, C, Identity
	Fuzzing	DoS	A
	Spoofing	Spoofing	Identity
NFC	Interception	Eavesdropping	C
	Data corruption through Interception	DoS	I, A
	Data Modification through interception	MiTM	I
	Data Insertion	MiTM	I
	NFC Data Exchange Format (NDEF) attacks	Identity Theft and Non repudiation	C, Identity

4.5. Radio-Frequency Identification (RFID)

Because the communication between the reader and RFID tags is made through an unprotected wireless channel, the transmitted data is exposed by unauthorized readers. RFID systems face different security threats as compared to the security threats encountered by traditional wireless systems [75]. Various hacking techniques against RFID are discussed as follows:

- **Tag Disable.** An attacker may remove the tag, delete the tag memory by sending a kill command, remove the antenna, give a high energy wave to a tag, and use a Faraday cage to block electromagnetic waves.
- **Tag Modification.** An attacker modifies or deletes valuable data from the memory of the tag.
- **Cloning Tags.** An attacker imitates or clones the tags after skimming the tag's information.
- **Reverse Engineering.** Using reverse engineering, an attacker can make a copy of a tag, and using tag examination, the attacker may get confidential data stored within a tag.
- **Eavesdropping.** RFID systems working in ultra high frequency (UHF) are more vulnerable to this threat. An attacker gathers the information shared between a valid tag and valid reader.
- **Snooping.** An attacker introduces an unauthorized reader to interact with the tag.
- **Skimming.** An attacker snoops data shared between a legitimate reader and legitimate tag.
- **Replay Attack.** An attacker spies to collect information about the IoT device or node replays eavesdropped information to achieve deception.
- **Relay Attacks.** An attacker places an illegitimate device between the tag and the reader to intercept, modify and forward information directly to other systems.

- **Electromagnetic (EM) Interference.** An attacker creates a signal in the same range as the reader to preclude tags from communicating with readers.
- **Fake RFID Tag Query.** An attacker sends queries and gets the same response from a tag at various locations to determine the location of a specific tag.
- **Cryptograph Decipher Attack.** An attacker decodes encryption algorithms by launching violent attacks and gets the plain text by deciphering the intercepted cryptography.
- **Blocker tag Attack.** Using a blocker tag, an attacker attempts to restrict the reader from reading tags.

4.6. Zigbee Protocol

The Zigbee protocol is one of the most popular IoT protocols used for communication in IoT devices because of its low cost, low power consumption and scalability. While the importance of security was considered during the design of Zigbee, some trade-offs have been kept to bring the cost of devices down and make them scalable at a low cost. Some of the standard security measures could not be implemented which ultimately resulted in security vulnerabilities. The major security threats against Zigbee networks are enumerated below.

- **Sniffing.** Zigbee networks are exposed to sniffing attacks since they do not implement encryption techniques. The attacker can capture some packets to execute malicious activities using some software tools like KillerBess's zbdump tool [76].
- **Replay Attack.** If an attacker is able to intercept the packets, the attacker can sniff raw packets of a network and could re-send the captured data as normal traffic [76].
- **Attaining the Link or Network key.** Since keys need to be reinstalled on the air when its objects require reflashing, an attacker can obtain the ZigBee network or link keys. Also, physical attacks can be used to obtain the key, where the keys can be extracted from ZigBee devices' flash memory when the device is physically accessed [77,78].
- **Eavesdropping.** An attacker can eavesdrop a ZigBee network and redirect its packets using an MiTM attack.
- **ZED Sabotage Attack.** Authors in [79] proposed an attack against the ZigBee protocol called the ZigBee End-Device (ZED). The purpose of the attack is to make the ZED unavailable by transmitting a particular signal periodically to wake up the device to drain its battery.

4.7. Wireless Fidelity (WiFi)

A detailed review of attacks against various versions of the 802.11 security mechanism (i.e., WPA, WPA2, WEP) is explained in [80]. The most common WiFi attacks are described below.

- **Attacks Related to Retrieving Key.** An attacker would monitor specific packets and then crack the key process offline. The common attacks in this category are Pyshkin, Tews, and Weinmann (PTW) attacks, Fluhrer, Mantin, and Shamir (FMS) attack, KoreK Family Attacks, Dictionary Attack and address resolution protocol (ARP) Injection [80].
- **Attacks Related to Retrieving Keystream.** An attacker only required to monitor for specific packets and then go on to perform the key cracking process offline. The common attacks in this category are PTW attacks, FMS attack, KoreK Family Attacks, Dictionary Attack and ARP Injection [80].
- **DoS or Availability Attacks.** This category of attacks includes those attacks that result in the unavailability of some service or network that is commonly called a DoS attack. These attacks usually target either a specific user or device, or try to exhaust network resources (e.g., the network router or Access Point), resulting in corrupting services for all users in that network. These attacks mostly depend on the broadcast of forged 802.11 management messages, which are easy to launch in versions of the WiFi standards up to 802.11n, as the management messages are transmitted

unguarded [81]. Attacks in this category include: Disassociation Attack, Block ACK flood, Authentication Request Flooding Attack, Deauthentication Broadcast Attack, Fake Power Saving Attack, Beacon Flooding Attack, Probe Request and Response Flooding Attacks. A survey of DoS attacks in 802.11 is covered in [82].

4.8. Bluetooth

Most of the issues found in Bluetooth are related to the pairing process. Attacks can be launched during the pairing process stages, like before the completion of the pairing process and after the pairing of devices is completed [83]. For instance, based on information collected after pairing, attackers can launch man-in-the-middle attacks. A review of Bluetooth security issues is explained in [83–85]. The common attacks against Bluetooth are discussed below.

- **PIN Cracking Attack.** This type of attack is performed during the pairing of the device and the process of authentication. An attacker collects the random number (RAND) and the Bluetooth Device Address (BD_ADDR) of the targeted device using some frequency sniffer tool. Then, a brute-force algorithm (for example, E22 algorithm) is applied to check all possible combinations of the PIN with the data collected earlier until the correct PIN is determined [84].
- **MAC Spoofing Attack.** An attack is launched during the process of link keys generation and before encryption is established. Devices manage to authenticate each other using generated link-keys. In this, attackers can imitate another user. Attackers can also dismiss connections or even alter data [84].
- **Man-in-the-Middle (MIM) Attack.** MIM attacks are launched when devices are trying to pair [86]. After the attack is launched, devices share messages unknowingly [58]. During this time authentication is performed without the shared secret keys [58]. When the attack is successful, the two devices are paired to the attacker [57,58], while they believe the pairing was successful.
- **Bluebugging.** An attacker exploits vulnerabilities of old devices firmware to spy on phone calls, send and receive messages, and connect to the Internet without legal users' knowledge.
- **Bluesnarfing.** An attacker gets unauthorized access to devices to retrieve information and redirect the incoming calls.
- **BluePrinting Attack.** This attack is launched to capture the device model, manufacturer, and firmware version of the device. This attack will work only if the target device's BD_ADDR is known.
- **Fuzzing Attack.** In a fuzzing attack, a device is forced to behave abnormally by an attacker through sending malformed data packets to Bluetooth radio of the device.
- **Brute-Force BD_ADDR Attack.** Since the first three bytes of BD_ADDR are fixed and known publicly, the brute-force attack is launched to scan on the last three bytes [84].
- **Worm Attacks.** In this attack, an attacker sends a malicious software or Trojan file to available vulnerable Bluetooth devices. Examples of these attacks are Sculls' worm, Cabir worm and Lasco worm.
- **DoS attacks.** These attacks target the physical layer or above layers in the protocol stack. Some typical DoS attacks are battery exhaustion, BlueChop, BD_ADDR duplication, BlueSmack, Big NAK (Negative Acknowledgement) and L2CAP guaranteed service.

4.9. Near Field Communication (NFC)

Although the communication range of NFC is restricted to a few centimeters, the International Organization for Standardization (ISO) standard does not guarantee secure communication. The common attacks against NFC technologies are briefly mentioned below [87].

- **Eavesdropping.** By using powerful and bigger antennas than those of mobile devices, NFC communications can be received or intercepted by an attacker in the vicinity of the devices. This allows an attacker to eavesdrop an NFC communication across larger distances.

- **Data Corruption.** An attacker can modify data transmitted over an NFC interface. If the attacker alters the data into an unrecognized format, this may result in DoS attacks.
- **Data Modification.** An attacker alters the actual data using amplitude modulations of data transmissions.
- **Data Insertion.** Malicious and undesirable data can be inserted in the form of messages into the data during the data exchange between two devices.
- **NFC Data Exchange Format (NDEF) attacks.** An attacker would exploit partial signatures, record composition attacks and establish trust [88].

4.10. IEEE 802.15.4

IEEE 802.15.4 is a technical standard, used by several IoT protocols, which describes the operation of low-rate wireless personal area networks (LR-WPANs). It stipulates the PHY layer and MAC for LR-WPANs. The IoT protocols based on IEEE 802.15.4 include 6LowPAN, ZigBee, Wireless HART, ISA 100.11a, MiWi, Thread and SubNetwork Access Protocol (SNAP). These protocols extended the standard by developing the upper layers, which are not covered in IEEE 802.15.4. The common attack types related to the IEEE 802.15.4 standard are explained in [89–91].

- **Radio interference Attack.** An attacker transmits high transmission powered radio interference signals over all channels of the related frequency band.
- **Symbol Flipping/ Signal Overshadowing Attack.** An attacker injects wrong data into a network by converting a legitimate data frame into an altered frame comprising information of the attacker's choice.
- **Steganography Attack.** Adversaries would use a hidden channel to exchange information about the launching of new attacks in the network.
- **Node-Specific Flooding.** In this, the emission of packets is used to cause degradation throughput IoT networks by flooding massive fake data.
- **Back-Off Manipulation.** An attacker transmits unnecessary packets to the victim and due to excessive packet reception, the targeted nodes' power sources are ultimately exhausted.
- **Battery Life Extension (BLE) Pretense.** An attacker transmits unnecessary packets to the victim and due to excessive packet reception, the targeted nodes' power sources are ultimately exhausted.
- **Random Number Generator (RNG) Tampering.** An attacker uses RNG in a way that guarantees that the back-off periods chosen by the adversary are much smaller than those selected by legitimate nodes.
- **Back-Off Countdown Omission.** This type of attack implicates the complete exclusion of the random back-off countdown by a malicious attacker.
- **Clear Channel Assessment (CCA) Manipulation/ Reduction/Omission.** An attacker gains channel access more frequently and quickly than it is done by legitimate network nodes.
- **Same-Nonce Attack.** An attacker obtains ciphertext keys to gather valuable information about transmitted data.
- **Replay-Protection Attack.** In this type of attack, frames with large sequence numbers are sent by attackers to targeted legitimate nodes. This results in dropping data frames with smaller sequence numbers from other legitimate nodes.
- **Acknowledgment (ACK) Attack.** An attacker sends back a false ACK on behalf of the receiver with the correct expected sequence number to the sender. This prohibits data retransmission by misleading the sender into believing that the frame has been delivered to the receiver successfully [89].
- **Guaranteed Time Slot (GTS) Attack.** GTS attacks are initiated against the network by exploiting the GTS management scheme.
- **Personal Area Networks Identifier (PANId) Conflict Attack.** An attacker can abuse the conflict resolution procedure by sending fake PANId conflict notifications to the targeted PAN coordinator

to start conflict resolution, thus temporarily preventing or delaying communications between the PAN coordinator and member nodes.

- **Ping-Pong Effect Attack.** This attack causes packet loss and service interruption, dropping node performance, and increasing consumption of energy and network load.
- **Bootstrapping Attack.** An attacker forces a targeted network node to become unrelated with its PAN at a time of the attacker's choosing by initiating any of the MAC or PHY layer attacks with the ultimate aim of causing DoS.
- **Steganography Attack.** An attacker hides information within the MAC and PHY frame fields of the IEEE 802.15.4 protocol [89]. Data can be hidden in IEEE 802.15.4 networks by using the PHY header field of PHY frames. Similarly, Steganography attacks would also be launched by hiding information within the MAC fields. Steganography attacks form a hidden channel between cooperating attackers in the network, which opens up a large number of prospects for adversaries.

4.11. Routing Protocol for Low Power and Lossy Network (RPL) Attack

The RPL protocol has been designed to allow point to point, multiple-point to point, and point to multiple-point communication. It is a distance-vector routing protocol based on IPv6. The RPL devices work on a specific topology that joins tree and mesh topologies called Destination Oriented Directed Acyclic Graphs (DODAG) [74,92]. Attacks against routing protocol can cause communication failures within IoT systems [93]. The interconnection of IoT systems to the Internet multiplies the vulnerabilities exponentially through exposure to innumerable attack vectors. The main attacks against RPL are discussed as follows:

- **Sinkhole Attack.** An attacker may announce a favorable route or falsified path to entice many nodes to redirect their packets through it.
- **Sybil Attack.** An attacker may use different identities in the same network to overcome the redundancy techniques in scattered data storage. Also, this can be used to attack routing algorithms.
- **Wormhole Attack.** An attacker disturbs both traffic and network topology. This attack can be launched by generating a private channel between two attackers in the network and transmitting the selected packets through it.
- **Blackhole Attack.** An attacker maliciously advertises itself as the shortest path to the destination during the path-discovering mechanism and drops the data packets silently.
- **Selective Forward Attack.** It is a variant of the Blackhole attack, where an attacker only rejects a specific subpart of the network traffic and forwards all RPL control packets. This attack is mainly targeted to disturb routing paths; however, it can also be used to filter any protocol [74].
- **Hello flooding attack.** An attacker can announce itself as a neighbor to many nodes, even the complete network by broadcasting a "HELLO" message with a strong powered antenna and a favorable routing metric. This is done by an attacker in order to deceive other objects to send their packet through it [94].

4.12. Internet Protocol (IPv6) and Low-Power Wireless Personal Area Networks (6LoWPAN) Based Attacks

6LoWPAN was designed to meet the communication requirements of connecting resource-constrained, low-powered objects and IPv6 networks. To achieve this, 6LoWPAN uses fragmentation at the adaptation layer. The main attacks against 6LoWPAN are explained as follows:

- **Fragmentation Attack.** IoT object communicating in IEEE 802.15.4 has a Maximum Transmission Unit (MTU) of 127 bytes, as opposed to in IPv6, which has a minimum MTU of 1280 bytes. This is done using a fragmentation mechanism. Since fragmentation is performed without using any type of authentication, an attacker can inject fragments among a fragmentation chain [95].
- **Authentication Attack.** In the absence of an authentication mechanism in 6LowPAN, any malicious object can join the network and get legitimate access [92].

- **Confidentiality Attack.** In the absence of an encryption technique in 6loWPAN, attacks affecting confidentiality, like eavesdropping, spoofing and Man in the Middle can be launched.

5. Intrusion Detection System (IDS)

Most IDSs have a common structure that includes: (1) a data gathering module collects data, which possibly contains evidence of an attack, (2) an analysis module detects attacks after processing that data, and (3) a mechanism for reporting an attack. In the data gathering module, the input data of each part of IoT systems can be gathered and examined to find normal behavior of interaction, thereby detecting malicious behavior at the early stages. The Analysis module can be implemented using various techniques and methods, however, ML and DL based methods are more suitable and dominant for data examination to learn benign and anomalous behavior based on how IoT devices and systems interact with one another in IoT environments. Furthermore, ML/DL methods can predict new attacks, which are often different from previous attacks, because ML/DL methods can intelligently predict future unknown attacks through learning from existing legitimate samples [12]. Figure 8 shows the components of typical IDS based on ML/DL methods.

5.1. Design Choices of ML/DL Based IDS

As depicted in Figure 9, the main differences in the design choices for IDSs depends on the following factors:

- **Detection methods.** It could be signature-based, anomaly-based or hybrid-based detection.
- **Architecture.** It can be classified as centralized and distributed architecture.
- **Data source.** It would be host-based, network-based, or hybrid-based data inputs.
- **Time of detection.** It can be online or offline detection.
- **Environment.** It would be wired, wireless, ad-hoc networks.

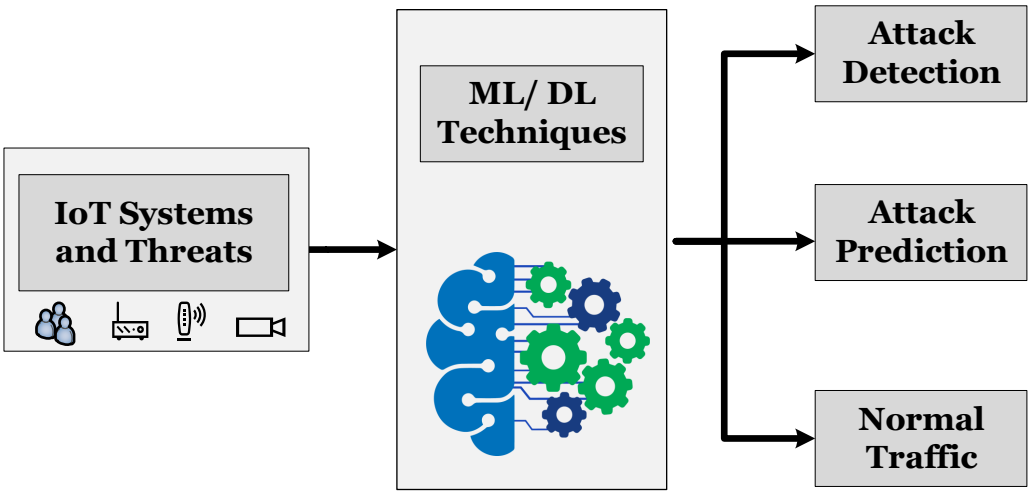

Figure 8. Role of Machine Learning/Deep Learning (ML/DL) Based IDS for IoT system.

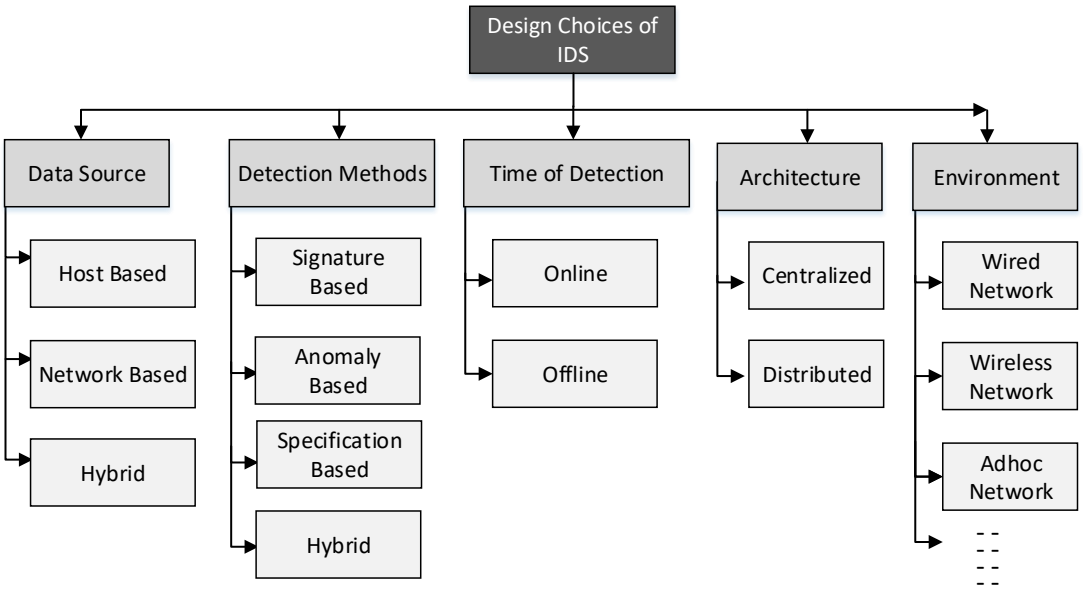

Figure 9. Taxonomy of design choices of IDS for IoT system.

5.2. Detection Methods of IDSs

The detection methods used for IDSs can be divided into four methodological types [33], as shown in Figure 9 and explained below.

5.2.1. Signature-Based Detection Techniques

Signature-based detection techniques contain a repository of attack signatures and compares the network traffic or system actions against this repository of signatures. As soon as any match is found, a detection alert is raised. Though sufficiently accurate against known attacks for which signatures exist in the repository, this technique cannot detect zero day (new) attacks. Even if it is not effective against mutations of an existing attack [54,96,97].

Some research, like [98], proposed means to overcome this deficiency of signature-based techniques through the use of an Artificial Immune System (AIS). This technique designed detectors relying on signatures/patterns of attacks using the model of immune cells, which can detect if a packet is normal or malicious through its classification as self or non-self element. The system has the capacity for the adoption of new patterns from continuous monitoring of the system. However, the feasibility of such a detection technique in a resource-constrained IoT environment is questionable.

The authors in [99] resolved this predicament of resource constraints in signature-based IDS through utilizing a separate Linux machine with an adapted version of the Suricata-based signature IDS. However, the authors did not provide any clues of updating attack signatures. The authors in [100] extended the work published in [99] by proposing modifications in signature matching techniques. Another research by [101] tackled processing power constraints of IoT systems through the use of auxiliary shift values with a multiple pattern detection algorithm, which enables a reduction in the number of matching operations required between attack signatures and network traffic packets. The system used signature repositories of the open-source IDS (Snort) and the open-source antivirus (ClamAV).

5.2.2. Anomaly-Based Detection Techniques

Anomaly-based detection techniques rely on a baseline normal behavior profile for the monitored environment [97,102]. This normal baseline is then used for comparison of system actions at any given moment. Any deviations out of bounds of the allowed threshold are reported by raising an alert without providing any classification for the type of attack detected. There are also attempts of using machine learning models that learn normal and attack events as behavioral detection models, but building normal profiles are better than learning normal and attack events that can not include new attack events in real-world networks. In comparison with signature-based detection techniques, anomaly-based detection techniques are more effective in discovering new attacks. One drawback of this technique is the difficulty in building the normal behavior baseline profile, which gives rise to increased false positive rates [20,103,104]. Anomaly-based detection techniques rely on ML algorithms to build a baseline normal profile of monitored systems. The use of such ML techniques in resource and energy-constrained IoT environments is still a challenge, due to high computational resources needed to train and validate ML techniques.

5.2.3. Specification-Based Detection Techniques

The basic principle of both anomaly-based detection and specification-based detection techniques is the same, where the normal behavior of a system is profiled through some means and is compared against current system actions to detect out of range deviations. However, in anomaly-based techniques, normal behavior is learned through ML, whereas for specification-based techniques it needs to be manually specified through a repository of rules and associated ranges of deviations by a human expert [105]. This allows for lowering the false-positive rates as compared to the anomaly-based detection techniques [20]. Having the advantage of not requiring any learning phase after specifying a rule set [105], these techniques suffer from lack of adaptability to varied environments and are liable to errors in specifications [19].

5.2.4. Hybrid-Based Detection Techniques

Hybrid-based detection techniques employ a mix of the earlier mentioned techniques to offset the shortcomings and optimize the advantages of detecting existing and new attacks. The authors in [106] proposed SVELTE, which is an IDS for IP-connected IoT systems that use RPL as a routing protocol in 6LoWPAN networks. This IDS was designed using a hybrid of anomaly and signature based detection techniques to obtain a balance between storage and processing requirements of each of these two techniques. They tried to balance the storage cost of the signature-based detection and computing cost of the anomaly-based techniques.

6. Machine Learning (ML) Techniques for IDS

As discussed in the previous section, apart from specification-based detection, all types of detection techniques rely on some sort of ML algorithm for the training phase of the IDS. In this section, an overview of different ML techniques used in IoT environment based IDSs is presented. Table 4 gives a brief overview of ML methods, their advantages and limitations along with reference to related research work conducted. In the end, Table 5 summarizes research works conducted to propose IDSs using various ML methods, as detailed below. Figure 10 illustrates the most common ML techniques used for designing IDSs in IoT networks.

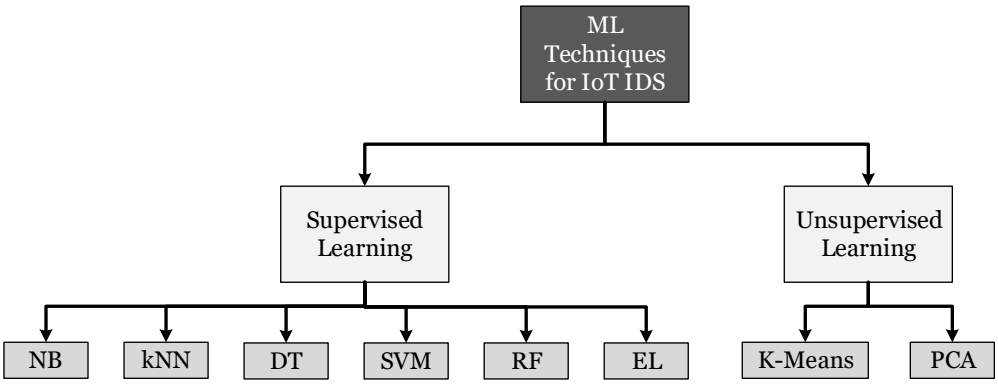

Figure 10. A taxonomy of ML Techniques for IoT-based IDSs.

6.1. Naive Bayes (NB) Classifier

This algorithm employs Bayes' theorem to predict the probability of occurrence of an event based on previous observations of similar events [107]. In ML scenarios, this can be used for classification of normal and abnormal behaviors based on previous observations in supervised learning mode. The NB classifier is a commonly used supervised classifier known for its simplicity. NB calculates posterior probability and based on that a labeling decision is made to classify unlabeled traffic as normal or anomalous. An independent set of features of the observed traffic like, status flags, protocol, latency, are used to forecast the probability of traffic being normal or otherwise. Being simple and easy to implement an algorithm, various IDSs have employed an NB classifier to identify anomalous traffic [108–111]. It requires very few samples for training [112] and can classify in both binary and multi-label classification. However, it fails to take into account interdependencies between features for classification purposes, which affect its accuracy [113].

6.2. K-Nearest Neighbor (KNN)

KNN does not require any parameters for its working. Euclidean distance is used to measure the distance between neighbors [114]. Figure 11 shows the basic principle behind the KNN classification algorithm, used to classify a new data instance into already observed classes based on its relative distance to either of the classes. The green squares depict the normal behavior class and red triangles show the abnormal behavior class, any newly observed unknown instance (blue hexagon) can now be classified based on the number of maximum nearest neighbors from either of the classes. Accordingly, this new instance is classified as a known class. k is the number of nearest neighbors used for classification.

The classification will change with the value of k. For $k = 1$, the red hexagon will be classified as an abnormal class, but for $k = 2$ and $k = 3$, it will be classified as a normal class. Hence, obtaining the optimal value of k through testing is vital for the accuracy of this algorithm [115]. Some researches [116–118] have used KNN based classification for anomaly and intrusion detection in general and IoT based network intrusion detection in particular [119,120] with reasonable accuracy in detecting User to Root (U2R) and Remote to Local (R2L) attacks. While KNN is simple to use, determining the optimal value of k and identifying missing nodes are time-consuming and costly in terms of accuracy.

6.3. Decision Trees (DTs)

Decision Trees (DTs) work by extracting features of the samples in a dataset and then organizing an ordered tree based on the value of a feature. Every feature is represented by a node of the tree and its corresponding values are represented by the branches originating from that node. Any feature node that optimally divides the tree in two is considered the origin node for the tree [121]. Various metrics

are utilized for identification of the origin node, which optimally divides the training datasets like the Gini index [122] and Information Gain [123].

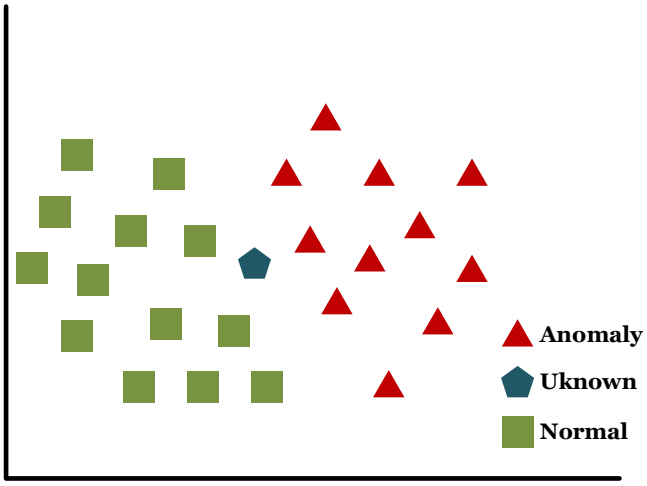

Figure 11. K-Nearest Neighbor (KNN) classification principle.

Figure 12 illustrates decision tree nodes. DT algorithms involve two processes, namely induction and inference, aimed at building the model and then carrying out the classification [124]. During the induction process, construction of a DT starts with adding nodes and branches. Initially, these nodes are unoccupied, and then through a process of feature selection through information gain and other measures, a feature is selected that is deemed to split the training dataset samples. This feature is then assigned as the origin vertex of the DT.

The process continues to select feature root nodes, to minimize the overlapping between different classes found in the training dataset. Resultantly, the accuracy of classifier increases in identifying distinct instances of a class. In the end, the leaves of each sub-DT are identified and classified according to their corresponding classes. After the construction of DT, the inference process can start, where any unknown instances of classes with features can be classified through iterative comparison with constructed DT. After the acquisition of a matching leaf node, the classification process for the new sample is completed [124]. In context of intrusion detection DTs have potential for use as classifier [125,126]. However, aspects of bigger storage requirements and computational complexity must be considered [124]. In the IoT environment, research published in [127] used DT to detect DDoS attacks through analysis of network traffic for identifying malicious sources.

6.4. Support Vector Machines (SVMs)

SVM is another type of classifier that works through the creation of a hyperplane in the feature set of two or more classes. The splitting hyperplane is found through a maximum distance of the nearest data point of each compared class [128], as shown in Figure 13. SVMs are most appropriate for the use case where classes containing large feature sets are required to be classified based on a fewer number of data samples [35,129,130]. Based on statistical learning [128], SVMs are ideal for anomaly detection where classification between normal and abnormal classes is required. SVMs are highly scalable due to simplicity and are capable of performing tasks like anomaly-based intrusion detection in real-time including online learning [131–133]. In [134], authors use an optimized version of SVM to propose "Sec-IoV", a multi-stage model for anomaly detection, for detection of anomalous traffic in vehicle-to-vehicle (V2V) communications in Internet of Vehicles (IoV) networks.

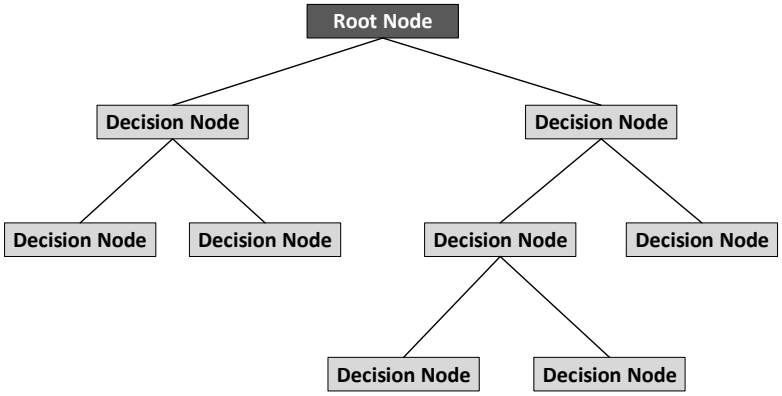

Figure 12. Depiction of Decision Tree Structure.

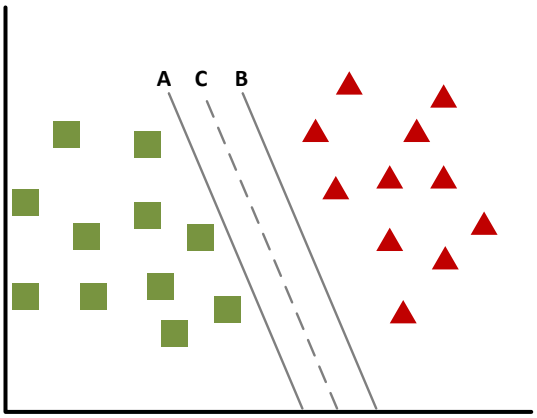

Figure 13. Depiction of Support Vector Machines (SVM) hyperplane splitting.

Another advantage of using SVM is its use of lesser storage/memory. The use of SVM-based IDSs in an IoT system has been evaluated in various research studies [135–137], where SVM showed more accurate results than other ML algorithms including DTs, NB and Random Forest. However, the use of optimal kernel function in SVM, which is used to separate the data when it is not linearly separable, remains a challenge to achieve the desired classification speed.

6.5. Ensemble Learning (EL)

EL works by building on strengths of various classifiers, through a combination of their results and then generating a majority vote out for classification, as shown in Figure 14. This improves classification accuracy through a combination of various homogeneous/heterogeneous classifiers' outputs [138,139]. EL is based on the study [140], where it was found that every ML classification algorithm depends on the application and associated data for its accuracy. Hence, no ML algorithm can be described as "one size fits all solution" and for generalized applications, EL like combinations may be best suited for maximizing accuracy through a reduction in variance and avoiding overfitting [141].

The accuracy of EL leads to the cost of increased time complexity, due to the use of multiple classifiers in parallel [142,143]. The efficacy of EL for intrusion detection has been examined in various studies [144–146]. The feasibility of EL under limited resource environments like IoT has been studied [147] with a generalized application lightweight EL framework being proposed for online anomaly detection in IoT networks. This study showed that such an EL algorithm produced better and accurate results than each member classifier individually [147].

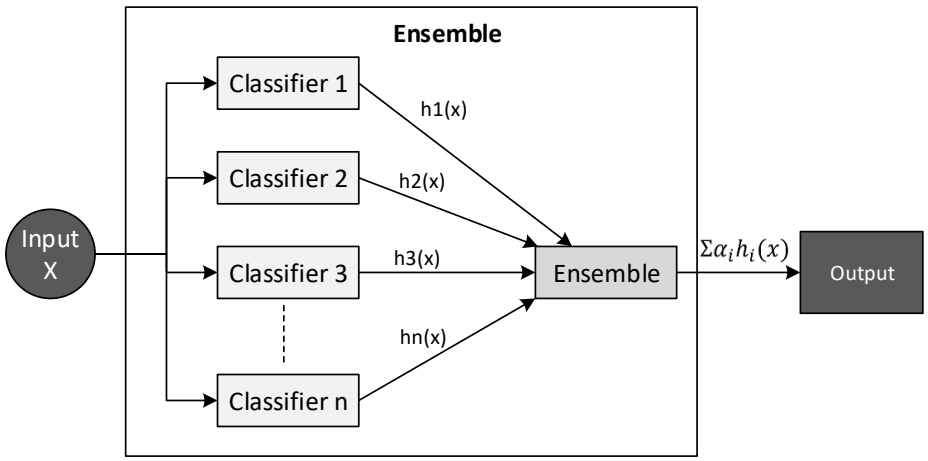

Figure 14. Working of an ensemble classifier.

6.6. Random Forest (RF)

RFs can be categorized as a supervised ML algorithm. An RF is built using multiple DTs to predict more accurate and error resistant classification results [148,149]. Randomly constructed DTs are trained to output classification results based on majority voting [148]. Though DTs can be considered as components of RF, there are two distinct classification algorithms due to the reason that contrary to DTs, which build a rule-set during training for subsequent classification of new samples, RF builds a rule-subset using all member DTs. This results in a more robust and accurate output, which is resistant to overfitting and requires substantially fewer inputs and does not require the process of feature selection [35]. As proposed by some studies [150,151], RF is suitable for anomaly and intrusion detection in IoT networks. Moreover, another study [152] has shown RF to be better than KNN, artificial neural network (ANN) and SVM at DDoS detection in IoT networks because it requires fewer input features and can bypass heavy computations required for feature selection in real-time IDS [153].

6.7. k-Means Clustering

It is an unsupervised algorithm, which is based on the discovery of *k* clusters in the data samples. Each instance of sample data is assigned to a particular cluster based on its features. The samples are distributed over *k* clusters according to their features using the estimation of centroids as per squared Euclidean distance. Recalculation of centroids of each cluster is then performed through taking the mean of data points allocated to that cluster, as shown in Figure 15. The process continues iteratively until no modifications to the clusters can be made [154,155]. Selection of an appropriate value of k and the assumption that the sample dataset will be equally distributed over the k clusters act as limitations for the k-means clustering algorithm. Previous studies presented in [156,157] suggest the suitability of k-means clustering for anomaly detection through calculating feature similarity. The authors [158] suggested combining DT with k-means clustering for anomaly detection in IoT networks to improve the performance.

6.8. Principle Component Analysis (PCA)

PCA is not an anomaly detection technique, but it is commonly used as a feature selection or a feature reduction technique from a large dataset. The selected feature sets can then be used along with some other ML classifiers to detect anomalies in an IoT network. The PCA technique transforms a large set of variables into a reduced set of features without losing much of the information. Various research works [159–162] used a combination of PCA with various classifiers to detect anomalies in IoT networks.

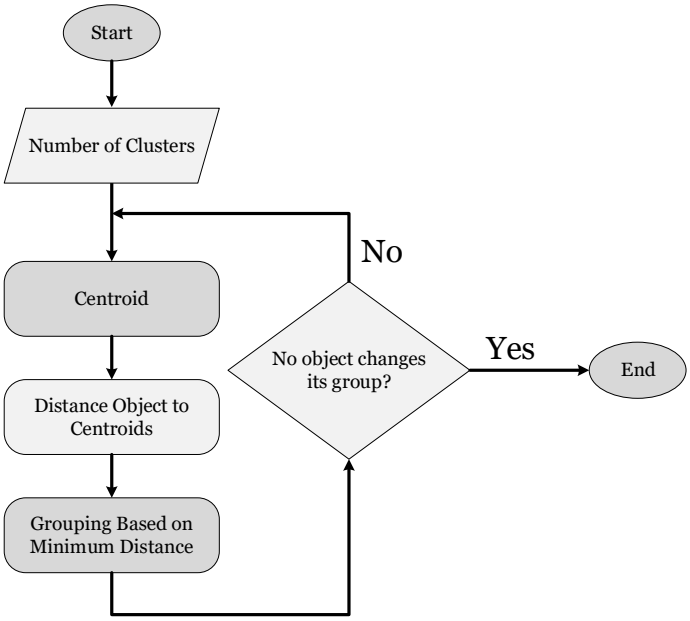

Figure 15. Illustration of k-means clustering.

Table 4. Taxonomy of ML based methods for IoT systems security.

ML Method	Attack Types Handled	Pros	Cons
KB [108,109,111,113]	HTTP attacks (Buffer overflow, Shell attacks) [111], DoS, Probe, R2L [109]	-It requires very few samples for training [112]. -It can classify in both binary and multi-label classification. -It shows robustness to irrelevant features.	It fails to take into account interdependencies between features for classification purposes, which affect its accuracy [113].
KNN [116–120]	U2R, R2L,Flooding attacks, DoS, DDoS	-Simple to use.	Determining optimal value of K and identifying missing nodes are challenging.
DT [125–127]	DDoS [127], U2R, R2L [125]	-Easy and simple to use method.	-It requires bigger storage -It is computationally complex -It is easy to use only if few DTs are used.
SVM [131–133]	Scan, DDoS (TCP, UDP flood), smurf, portsweep	-SVMs are highly scalable due to simplicity and are capable of performing tasks like anomaly-based intrusion detection in real-time including online learning. -SVMs are considered suitable for data containing a large number of feature attributes. -SVMs use lesser storage and memory.	-The use of optimal kernel function in SVM, which is used to separate the data when it is not linearly separable, remains a challenge to achieve desired classification speed. -It is difficult to understand and interpreting SVM-based models.
EL [144–146,163]	DoS, Probe, R2L, U2R attacks	-It is robust to overfitting. -Performs better than a single classifier. -It reduces variance.	-Increased time complexity, due to the use of multiple classifiers in parallel
RF [150,151]	DoS, Probe, R2L, U2R	-It produces a more robust and accurate output which is resistant to overfitting. -It requires substantially fewer inputs and does not require the process of feature selection.	-Since RF constructs several DTs, its use may be impractical in real-time applications requiring large dataset.
***K*-Means** [157,158,164]	DoS, Probe, R2L, U2R	-k-Means clustering does not require labeled data.	-It is less effective as compared to supervised learning technique, in particular detecting known attacks.
PCA [159–162]	Used in combination with other ML methods	-PCA is suitable where the dataset involves large set of variables as PCA transforms it to reduced set of features without losing much of information. -Can reduce the complexity in the data.	-It is not an anomaly detection method, it must be used with some other ML methods to design a security model.

Table 5. Comparison of studies that used ML and DL techniques in IoT Security.

Study	Machine Learning							Deep Learning							Dataset	Threat Detected
	NBC	KNN	DT	SVM	EL	RF	*K*-Means	RNN	CNN	AE	RBM	DBN	EDLN	GAN		
[110,111]	✓	-	-	✓	-	-	✓	✓	-	-	-	-	-	-	KDD99	Anomaly Detection
[116]	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	KDD99	apache2 udpstorm processtable mailbomb
[125]	-	-	-	-	-	-	-	✓	✓	-	-	-	-	-	ADFA-LD and ADFA-WD	Adduser Meterpreter Webshell
[129]	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	DARPA dataset	Probe attack, U2R attack
[143]	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	KDD99	Network Traffic anomaly detection
[150]	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	Boot-strapped	Worms, Buffer overflows
[157]	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	KDD99	-
[165]	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	ISCX2012	PROBE attacks or non-PROBE attacks
[166]	-	-	-	-	-	-	-	✓	✓	-	-	-	-	-	Android Malware Genome project	Malware
[167]	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	Outlier Detection DataSets	Anomaly detection
[168]	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	KDD	-
[169]	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	NSL-KDD	-
[170]	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	500 samples for dataset	Anomaly detection

7. Deep Learning (DL) Techniques for IDSs

DL algorithms outperform ML algorithms in applications involving large datasets. DL becomes most relevant in IoT security applications as IoT environments are characterized by the production of vast amounts and a variety of data [171]. Furthermore, DL is capable of the automatic modeling of complex feature sets from the sample data [171]. Another advantage of DL algorithms is their ability to allow deep linking in IoT networks [172]. This enables automatic interactions between IoT-based systems in the absence of human intervention [171] to perform assigned collaborative functions.

Because of their ability to extract hierarchical feature representations in complex deep architecture, DL can be classified as a branch of ML algorithms that uses multiple non-linear layers of processing to extract feature sets. These feature sets are then used for abstraction and pattern detection after necessary transformations [173]. As shown in Figure 16, DL can be used in a generative mode with unsupervised learning, discriminative mode using supervised learning, or a hybrid approach by combining both modes.

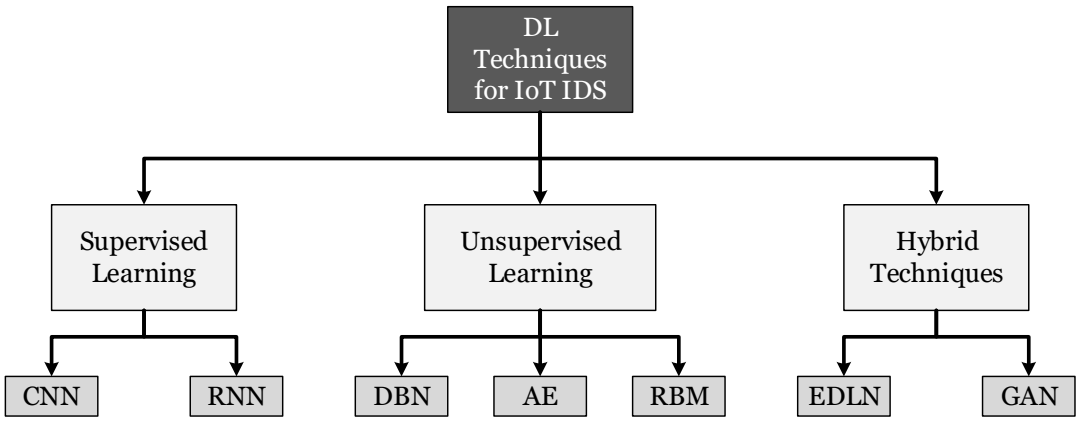

Figure 16. Taxonomy of potential DL techniques for IoT IDS.

In this section, various major DL based techniques used for designing an IDS are discussed. Table 5 below summarizes research studies conducted to propose IDS using various DL-based methods. Details about each research work along with the DL technique is explained in respective sub-sections below.

7.1. Recurrent Neural Networks (RNNs)

RNN is a discriminative DL algorithm, which is best suited in environments where data is to be processed sequentially. Unlike other neural networks, its output is dependent on back-propagation instead of forward propagation [173–175]. A temporal layer is incorporated in an RNN for analyzing data sequentially followed by learning about multi-dimensional differences in unrevealed units of recurrent components [165]. Modifications to these unrevealed units are then made corresponding to data encountered by the neural network, causing continuous updates and the manifestation of the current state of the neural network.

The current unrevealed state of the neural network is processed by an RNN algorithm through the estimation of succeeding hidden states as triggering of a previously unrevealed state. A simple explanation of RNN functioning is described in Figure 17. Here, outputs from neurons are sent back as feedback to the neurons of the previous layer. Because IoT environments are characterized by the generation of large amounts of sequential data like network traffic flows, RNNs become relevant in IoT security applications, especially network intrusion detection. Previous research [176] has proposed the use of an RNN for network intrusion detection through analysis of network traffic behavior and reported obtaining useful results, particularly time series-based threats. Another recent research [177]

proposes an IDS that uses cascaded filtering stages in which deep multi-layered RNN are applied for each filter. RNNs are then trained to detect common attacks launched in IoT environments, like R2L, Dos, U2R and Probe.

Long short-term memory (LSTM) network architectures, which are a specialized form of RNN, have also been used in the designing of IDS. The main attribute of LSTM based RNNs is to persist information or cell state for later use in the network. This feature makes them appropriate for performing analysis of temporal data that changes over time. Thus, LSTM networks are preferred to solve problems related to anomaly detection in time-series sequence data. Various forms of RNN, including LSTM based RNNs, have been used for anomaly and intrusion detection in IoT networks by researchers in [178–183]. While RNNs have demonstrated promising results in predicting time series data, the detection of anomalous traffic using these predictions is still challenging.

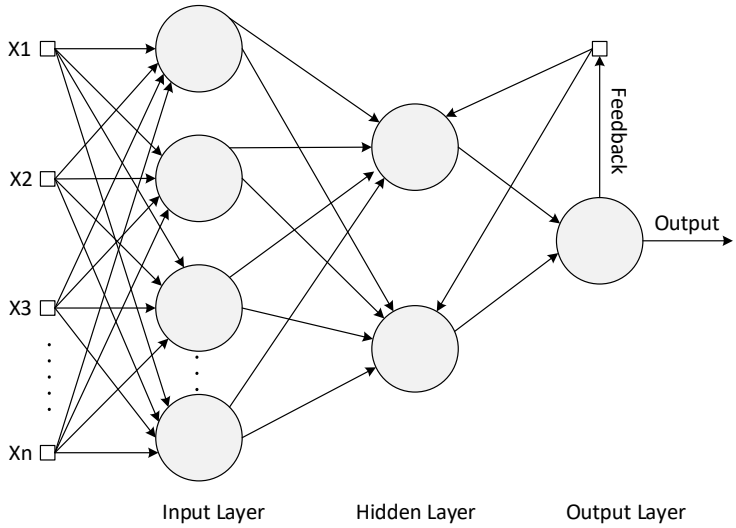

Figure 17. Illustration of recurrent neural network algorithm.

7.2. Convolutional Neural Network (CNN)

CNN is also a discriminative DL algorithm, which was designed to minimize the number of data inputs required for a conventional artificial neural network (ANN) through the use of equivariant representation, sparse interaction and sharing of parameters [184]. Thus CNN becomes more scalable and requires less time for training. There are three-layer types in a CNN, namely convolutional layer, pooling layer and activation unit, as shown in Figure 18. The convolutional layers use various kernels for convoluting data inputs [185]. The pooling layers downsize samples, thus minimizing the sizes of succeeding layers. It involves two techniques: Max pooling and average pooling, where the former chooses a maximum value for every cluster of past layers after distributing the input among distinctive clusters [186,187].

The average pooling, on the other hand, calculates the average values of every cluster in the previous layer. The activation unit is able to trigger an activation function on every feature in the feature set in a non-linear fashion [187]. CNN is best suited for highly efficient and fast feature extraction from raw data but at the same time CNN requires high computational power [188]. Hence using CNN on resource-constrained IoT devices for their security is highly challenging. This challenge is somewhat addressed through distributed architecture where a lighter version of Deep NN is trained and implemented on-board with only a subset of vital output classes, whereas, the high computational power of the cloud is used to perform the complete the training of the algorithm [166]. Their use in IoT environment security was discussed in previous research published in [189,190] for malware detection. In [40], authors propose a hybrid data processing model for network anomaly detection that utilizes

Grey Wolf Optimization (GWO) and CNN techniques. Authors claim to have achieved better accuracy and detection rate in comparison to other state-of-the-art IDS.

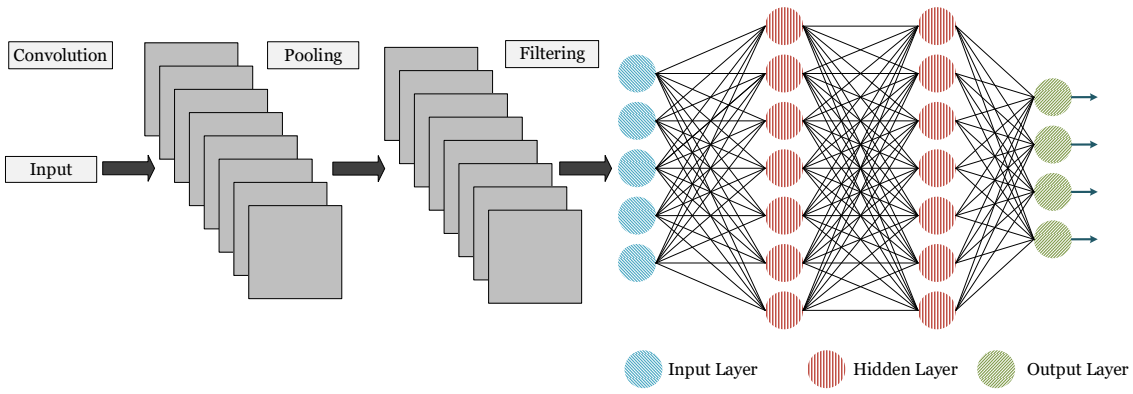

Figure 18. Illustration of convolution neural network working.

7.3. Deep Autoencoders (AEs)

It is an unsupervised algorithm designed for the reproduction of its input at its output through the use of a decoder function and a hidden layer containing the definition of a code utilized for the representation of input [184]. The other function in an AE neural network is called the encoder function and is responsible for the conversion of the acquired input into code. During training, reconstruction errors must be minimized [191]. One use case for AE is feature extraction from the datasets. However, these suffer from the requirement of high computational power. Deep AEs have been used for the detection of network-based malware in previous research with better accuracy than SVM and KNN [167]. Kitsune [41] is one such study where an ensemble of deep auto-encoders was used to implement an online lightweight IDS for IoT environments based on unsupervised learning and anomaly detection where authors demonstrate better accuracy as compared to other ML and DL techniques.

7.4. Restricted Boltzmann Machine (RBM)

It is an unsupervised learning-based algorithm and builds a deep generative and undirected model [168]. There are no two nodes in any layer of an RBM that have any connection with each other. Visible and hidden layers are the two types of layers making up an RBM. Known input parameters are contained in the visible layer, while the unknown potential variables are included with several layers forming the hidden layer. Working hierarchically, features extracted from a dataset are then passed on to the next layer as latent variables. RBMs were used in various research work [192,193] for network/IoT intrusion detection systems. The challenge of implementing RBMs is that it needs high computational resources while implementing it on low-powered IoT devices. Furthermore, Single RBM lacks the capability of feature representation. However, this limitation can be overcome by applying two or more RBM stacked to form a Deep Belief Network (DBN).

7.5. Deep Belief Network (DBN)

Being formed by stacking two or more RBMs, DBN can be considered as unsupervised learning based generative algorithms [194]. They perform robustly through unsupervised training for each layer separately [165]. Initial features are extracted in the pre-training phase for each layer, followed by a fine-tuning phase where the application of a softmax layer is executed on the top layer [170]. It is mainly composed of two layers, i.e., visible layer and hidden layer, as shown in Figure 19. Though the study in [188,195] discussed malicious attack detection using DBNs with comparatively better results than ML algorithms, no evidence of applicability in the IoT environment was reported in the literature.

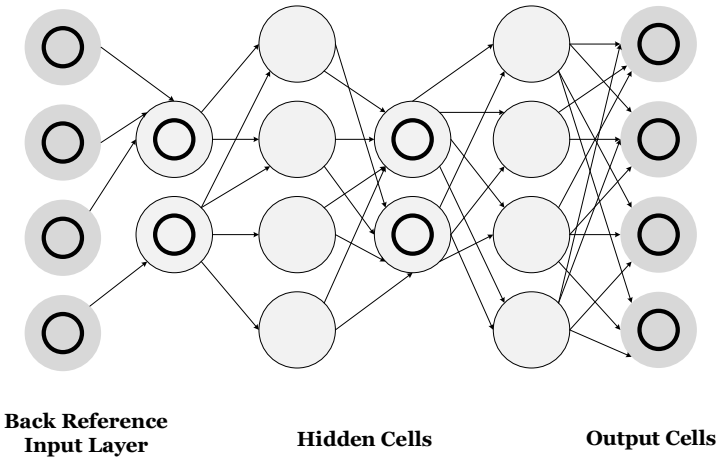

Figure 19. Illustration of deep belief network working.

7.6. Generative Adversarial Network (GAN)

It is a hybrid DL method that uses both generative and discriminative models at the same time for training [196]. Distributions of the dataset and samples is obtained by the generative model predictions about the authentic origination of a given sample from a training dataset and are made by the discriminative model [196]. As shown in Figure 20, both generative and discriminative models work as adversaries where the generative model attempts deception through the generation of a sample using random noise. On the other hand, the discriminative model attempts to authenticate real training data samples from deceptive samples generated by the generative model. Here, *D(x)* represents a binary classification giving output as real or fake (generated). The measure of correct/incorrect classification determines the accuracy and performance of both the models in an inversely proportional fashion. This results in models updating in each iteration [191]. The study published in [169] discussed the utility of the GAN algorithm for detecting anomalous behavior in IoT environments with promising results due to their ability to counter zero-day attacks through the generation of samples mimicking zero-day attacks, thereby causing the discriminator to learn different attack scenarios. However, the challenge with using GAN is that its training is difficult and it produces unstable results [196,197].

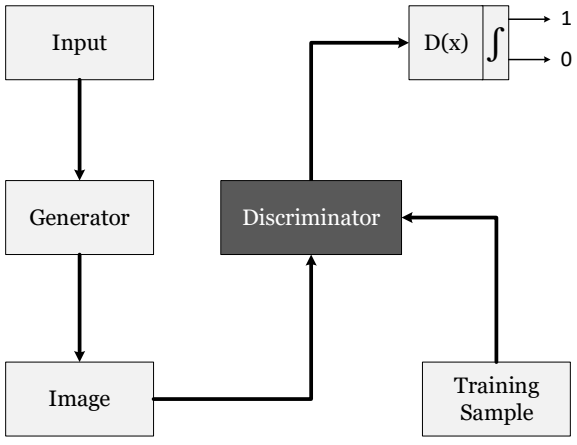

Figure 20. Illustration of generative adversarial network (GAN) working.

7.7. Ensemble of DL Networks (EDLNs)

As discussed earlier, the ensemble of various ML classifiers proves more effective than individual ML classifier results. Similarly multiple DL algorithms can be used in parallel through organizing in an ensemble to produce better results than each component DL algorithm. EDLNs can have any

combination of a discriminative, generative, or hybrid type of DL algorithms. Best suited for solving complex issues, EDLNs perform better in uncertain environments with a high number of features. A heterogeneous EDLN has classifiers from the different genres, whereas a homogeneous EDLN has classifiers from the same genre. Both compositions are aimed at increasing efficiency and producing accurate results [198]. Application of EDLN for IoT security requires further study and research, to evaluate the possibility of improving the performance and accuracy of the IoT security system [12]. Table 6 illustrates common attack types handled by corresponding DL methods along with reference to related research. Table 6 also describes advantages and limitations of each suggested DL method. Later, Table 5 below covers the comparison of work conducted on ML and DL techniques on IoT Security.

Table 6. Taxonomy of DL based methods for IoT systems security.

DL Technique	Attack Types Handled	Pros	Cons
RNNs [176–183]	-R2L, DoS, U2R and Probe [177] -Botnet [176] -In particular suitable to detect anomalies in time series data [179–183]	-Best suited in environments where data is to be processed sequentially. -In some cases, IoT system environment produces sequential data, hence RNNs are suitable in IoT security.	The major challenge in the use of RNNs is handling the issue of vanishing or exploding gradients, which hinders learning of long data sequences.
CNNs [189,190]	Malware attacks	-CNN is best suited for highly efficient and fast feature extraction from raw data. -Since CNN can automatically learn behavior from raw network security data, they have potential application in IoT security.	-CNN requires high computational power; thus using CNN on resource-constrained IoT devices for their security is highly challenging.
Deep Autoencoders [41,167]	-Malware attacks -Botnet attacks [41]	-AEs have been successfully used for feature extraction and dimensionality reduction.	-AEs are computationally heavy. -May not produce desired results if the training dataset is not representative of the testing dataset.
RBM [192,193]	-R2L, DoS, U2R and Probe	- Feedback function of RMBMs facilitates extraction of important attributes which are then used to capture the behavior of IoT traffic.	-RBMs needs high computational resources while implementing it on low-powered IoT devices. -Single RBM lacks the capability of feature representation.
DBNs [188,195]	-R2L, DoS, U2R and Probe	-Suitable for vital feature extraction with training on unlabeled data.	-DBNs require high computational costs.
GAN [191]	-Botnet (Mirai, Bashlite), Scanning, MiTM	-Ability to detect zero-day attacks -Generating a sample needs only one pass through the model.	-Its training is difficult and it produces unstable results.
EDLNs [41,198]	-Malware, DoS, Botnet, MiTM	-Ensemble of DL classifiers can achieve better model performance -EDLNs perform better in uncertain environments with a high number of features.	- EDLNs are computationally heavy and complex.

8. Datasets Available for IoT Security

Evaluating the effectiveness of any IDS entails a reliable and current dataset that contains present benign and anomalous activities. Most of the earlier research in IDS relied on the KDD99 [199] dataset due to the absence of other datasets for about two decades. However, analysis suggests that the KDD99 dataset negatively affects the IDS results in [199,200] and [201]. Numerous research efforts have been undertaken to address the weaknesses of KDD99 and other datasets that appeared after that. A brief description of the most common datasets for evaluating IDS is presented below.

- **KDD99.** This is a modification of the DARPA funded DARPA98 dataset that initiated from an IDS program conducted at MIT's (Massachusetts Institute of Technology) Lincoln Laboratory for evaluating IDSs that differentiate between inbound normal and attack connections. Later on, this dataset was used in the International Knowledge Discovery and Data Mining Tools Competition [202] after some filtering, resulting in what is known as the KDD CUP 99 dataset [199]. This dataset has been used by most of the researchers for the last two decades now. The absence of alternatives has resulted in several works directed on the KDD CUP 99 dataset [199] as a widespread benchmark for the accuracy of the classifier. However, KDD-99 possesses numerous weaknesses, which discourage its use in the current context, including its age,

highly skewed targets, non-stationarity between training and test datasets, pattern redundancy, and irrelevant features.

- **NSL-KDD.** NSL-KDD is an effort by the researchers who published their work in [199] to overcome the weaknesses of KDD-99. It is a more balanced resampling of KDD-99 where the emphasis is laid on examples that are expected to be missed by classifiers trained on the basic KDD-99. However, as their authors acknowledge themselves, there are still weaknesses in the dataset, like its non-representation of low footprint attacks [200].
- **The DEFCON dataset.** DEFCON-8 dataset, generated in 2000, comprises of port scanning and buffer overflow attacks. Another version, the DEFCON-10 dataset, generated in 2002 uses bad packets, FTP by telnet protocol, administrative privilege, port scan and sweeps attacks [203]. The traffic produced during the Capture the Flag (CTF) competition is dissimilar from network traffic of the real world because it primarily consists of attack traffic as opposed to usual background traffic, therefore its applicability for evaluating IDS is limited. The dataset is mostly used to assess alert correlation techniques [204,205].
- **The Center of Applied Internet Data Analysis (CAIDA)—2002–2016 datasets [203,206].** This organization has three different datasets: (1) the CAIDA OC48, covering various types of data observed on an OC48 link, (2) the CAIDA DDOS, which comprises of one-hour DDoS attack traffic that happened in August 2007, and (3) the CAIDA Internet traces 2016, which is passive traffic traces from CAIDA's Equinix-Chicago monitor on the high-speed Internet backbone [207]. These datasets are specific to certain events or attacks and are anonymized with their protocol information, payload, and destination. These are not effective benchmarking datasets because of several shortcomings, as discussed in [207], like the unavailability of ground truth about the attack instances.
- **The LBNL dataset** contains anonymized traffic, which is comprised of only header data. The dataset was generated at the Lawrence Berkley National Laboratory, by gathering real outbound, inbound and routing traffic from two edge routers [206]. It lacked the labeling process and also no extra features were created [206].
- **The UNSW-NB15** is a dataset developed at UNSW Canberra by the researchers of [208] for the evaluation of IDS. The researchers used the IXIA PerfectStorm tool to generate a mixture of attack and benign traffic, at the Australian Center of Cyber Security (ACCS) over two days, in sessions of 16 and 15 h. They generated a dataset of size 100 GB in the form of pcap files with a substantial number of novel features. NB15 was planned as a step-up from the KDD99 dataset discussed above. It covers 10 targets: one benign, and nine anomalous, namely: DoS, Exploits, Analysis, Fuzzers, Worms, Reconnaissance, Generic, Shell Code and Backdoors [208]. However, the dataset was designed based on a synthetic environment for producing attack activities.
- **The ISCX datasets [207].** The Canadian Institute for Cybersecurity has been working on the generation of numerous datasets that are used by independent researchers, universities and private industry around the world. A few datasets relevant to our work are IPS/IDS dataset on AWS (CSE-CIC-IDS2018), IPS/IDS dataset (CICIDS2017), CIC DoS dataset (application-layer), ISCX Botnet dataset, ISCX IDS 2012 dataset, ISCX Android Botnet dataset, and ISCX NSL-KDD dataset. Their latest dataset related to our work is CICIDS2017. This dataset covers benign and the most up-to-date common attacks, which is comparable to the real-world data [209]. The CICIDS2017 consists of multiple attack scenarios, with realistic user-related background traffic produced by using the B-Profile system. For this dataset they built the abstract behavior of 25 users based on the FTP, SSH, HTTP, HTTPS and email protocols. However, the ground truth of the datasets, which would improve the reliability of the labeling process, was not shared. Moreover, applying the idea of profiling, which was used to produce these datasets, in real networks could be problematic due to their intrinsic complexity [209].
- **The Tezpur University IDS (TUIDS) dataset [206].** This dataset was generated by the professors from Tezpur University, India. This dataset features DoS, Probing, Scan, U2R and DDoS attack

scenarios, performed in a testbed. However, the flow level data does not contain any new features other than those produced by the flow-capturing process [209].

- **BoT-IoT [209]** The BoT-IoT dataset was created by designing a realistic network environment in the Cyber Range Lab of The center of UNSW Canberra Cyber. The environment incorporates a combination of normal and botnet traffic. Researchers also present a testbed setting for handling the existing dataset shortcomings of capturing complete network information, correct labeling, and the latest and complex attack diversity. In their work, the authors also evaluate the reliability of the BoT-IoT dataset using different ML and statistical techniques for forensics purposes in comparison to the other datasets discussed above. The dataset's source files are provided in different formats, including the original pcap files, the generated argus files and CSV files. The files were separated, based on attack category and subcategory, to better assist in the labeling process. The dataset contains OS and Service Scan, DoS, DDoS, Data exfiltration and Keylogging attacks. Based on the protocol used, the DDoS and DoS attacks are further organized [209].
- **IoTPoT Dataset [210].** This dataset was generated through honeypots, so there was no process for manual labeling and anonymization; however, it has restricted view of the network traffic since only attacks launched at the honeypots could be observed. Authors claim that IoTPoT examines Telnet-based attacks against different IoT devices running on different CPU architectures such as MIPS, ARM and PPC. During 39 days of operation, authors recorded 76,605 download attempts of malware binaries from 16,934 visiting IP. Authors further claim that none of these binaries could have been detected by existing honeypots that handle the Telnet protocol, such as telnet password honeypot and honeyd, because these honeypots are not capable of handling different incoming commands initiated by the attackers [210].
- **N-BaIoT Dataset.** The most recent dataset, specifically related to the evaluation of IDS for IoT networks is generated by the authors [41] as part of their research work on online network IDS. They generated and collected traffic from two networks: one, an IP camera video surveillance network where they launched eight different types of attacks that affect the availability and integrity of the video uplinks; two, an IoT network comprising of three PCs and nine IoT devices, out of which one was infected with the Mirai botnet malware. A detailed explanation of the attacks and network topologies is available in their published papers [41]. The authors compiled a dataset of extracted feature vectors for each of these nine attacks. The attack types include: OS Scan, Fuzzing, Video injection, ARP MiTM, Active Wiretap, SSDP flood, SYN DoS, SSL Renegotiation and Mirai.

9. Challenges and Future Research Directions

A large number of studies and research works have been published related to IDSs for IoT. However, there are still a large number of open research challenges and issues, particularly in the use of ML and DL techniques for anomaly and intrusion detection in IoT. The challenge is that there exists no standard mechanism that guarantees validation of the proposed systems or method. The research works mostly demonstrate evaluation of their proposed systems based on synthesized datasets and address one specific problem which may not work in the real world on real data and in the presence of other problems. As evident from this and other similar studies conducted on state of the art in IDS for IoT, it is very difficult to design an IDS which covers, at least, the most important aspects of an effective IDS, that is it is deployable, online, scalable, works effectively on real data and satisfies all stakeholders requirements. Instead, most of the published work share evaluation results tested on contrived datasets, cover a single or some part of the system, and show results using biased parameters.

Furthermore, a proof of completeness and accuracy of any proposed IDS is very hard to define or accomplish. Thus, one of the conclusions from this study is that it is very hard to design a comprehensive IDS, which can offer good accuracy, scalability, robustness and protection against all types of threats. Below, some of the major issues and challenges that researchers face today and in the future are described. Since the IoT security measures are still not matured, there is enormous

scope for future research in this area, particularly in anomaly and intrusion detection using ML and DL techniques.

The most recent challenges related to anomaly and intrusion detection in IoT networks are discussed in the following:

- To test and validate proposed NIDS, a good quality dataset related to IoT IDS is very essential. Such a dataset should possess a reasonable size of network flow data covering both attack and normal behavior with the corresponding label. Furthermore, in order to capture normal behavior, normal traffic data from each type of IoT device is required, other than the attack data for testing the NIDS. However, as discussed in the previous section, most of the publicly available datasets lack in providing the required features, like missing labels, incomplete network features, missing raw pcap files and are difficult to comprehend and/or have incomplete CSV files. Moreover, datasets available only capture normal behavior of a specific type of IoT devices, which restricts training of IDS on those devices only. Creating a dataset that can address these issues in a real environment will be a challenge and a potential area of research.
- Developing an online and real-time, anomaly-based IDS for IoT networks is very challenging. This is because such an IDS would require to learn a normal behavior first to detect abnormal or malicious behavior. The learning phase assumes that there is no noise or attack traffic during this period which cannot be guaranteed. Such an IDS may generate false alarms if these issues are not addressed.
- As also described in this paper, most of the anomaly-based NIDS tries to construct a model that captures the profile of all possible behavior or patterns of normal traffic. This, however, is extremely challenging because it has been proven that such models tend to bias towards the dominated class, that is, normal class, resulting in high false-positive rates. Furthermore, it is also not possible to capture all possible normal observations that may be generated in a network, particularly in a heterogeneous environment of IoT networks, which increases false-negative rates. Completely avoiding or minimizing false-positive rates and false-negative rates in NIDS is another research challenge.
- It would be interesting to develop models trained on specific types of devices. These models can be applied to IDSs in other organizations using a similar type of device. This will assist other organizations, which can deploy these models and thus save time that would have been required to collect the data and train the IDSs. It will also help in detecting malicious IoT devices, which are already compromised because their behavior would be different from normal behavior captured by trained models. Developing such models is a challenging task and a potential area for future research.
- Different stages involved in the design and implementation of NIDS, like data-preprocessing and feature reduction, model training and deployment, in particular, ML and DL based NIDS, increase computational complexity. Thus designing an efficient NIDS that is light on computational requirements is another challenge and area for future research.
- Feature selection and dimensionality reduction methods used for proposed IDSs are suitable to work on a specific type of normal traffic and to detect a particular type of attacks which may not work once the environment of normal or attack sequences change a bit, especially under a fast-changing environment of IoT devices and networks. Thus, dynamic and computationally efficient mechanism for feature selection which can work under all types of normal and attack traffic is a potential research challenge.
- DL and ML-based techniques and algorithms are being widely used for training a model on a large dataset. This has facilitated in effective handling of cyber-attacks. However, with regards to the use of DL and ML algorithms for attack detection in IoT networks, some challenges need the attention of researchers; for example, resource constraints issue with IoT devices limits the use of DL/ML algorithms [163] for protection of IoT networks. Another challenge with the use

of ML/DL techniques in large and distributed networks, like that of IoT networks, is that they face scalability issues, for example in terms of various scenarios and choices of IDS deployment. One possible solution to limitations of individual DL or ML algorithms suggested by some of the authors [211] is the use of an ensemble of ML/DL algorithms that performed better in comparison to an individual ML algorithm; however, such algorithms were computationally expensive and thus resulted in network latency issues, which cannot be afforded in critical systems involving risks to human lives, like health and autonomous or internet of vehicles (IoVs) systems.

- The techniques of semi-supervised learning, transfer learning and reinforcement learning (RL) are still not well explored and experimented for designing an IDS for IoT security in order to achieve important objectives like real-time, fast training and unified models for anomaly detection in IoT and thus are potential areas of future research. Moreover, it would be an interesting research area to use RL in combination with DL because their combined use can be beneficial in IoT network scenarios involving large data dimensionality and non-stationary environments.

10. Conclusions

During the last decade, the use of IoT devices has increased exponentially in all walks of life due to its capacity of converting objects from different application areas into Internet hosts. At the same time, users' privacy and security are threatened due to IoT security vulnerabilities. Therefore, there is a requirement to develop more robust security solutions for IoT. Machine and deep learning-based IDS is one of the key techniques for IoT security. In this work, a survey of ML and DL based Intrusion Detection techniques used in IDS for IoT networks and systems is presented. The IoT architecture, protocols, IoT systems vulnerabilities, and IoT protocol-level attacks have been discussed in detail. Then, this paper surveyed various research work available in the literature, which suggested IDS methodology for IoT or proposed attack detection techniques for IoT that could be part of an IDS, specifically about various ML and DL techniques available for IDS in IoT and their use by the researchers. Also, a review of various datasets available for IoT security-related research is elaborated. This work attempts to provide the researchers with the summarized but comprehensive and useful insight into the various security challenges currently being faced by IoT systems and networks and possible solutions, with a focus on intrusion detection, based on ML and DL based methods.

Author Contributions: Conceptualization, J.A., N.M. and H.K.; methodology, J.A.; software, Not applicable; validation, N.M., E.D. and J.A.; formal analysis, J.A., N.M. and A.W.; investigation, J.A. and W.H.; resources, W.H. and A.W.; data curation, J.A. and W.H.; writing—original draft preparation, J.A. and N.M.; writing—review and editing, J.A. and E.D.; visualization, J.A. and H.K.; supervision, N.M. and H.K.; project administration, A.W.; funding acquisition, Not Applicable. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

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

References

1. Ray, S.; Jin, Y.; Raychowdhury, A. The changing computing paradigm with internet of things: A tutorial introduction. *IEEE Des. Test* **2016**, *33*, 76–96. [CrossRef]
2. Diechmann, J.; Heineke, K.; Reinbacher, T.; Wee, D. The Internet of Things: How to Capture the Value of IoT. Technical Report, Technical Report May. 2018, pp. 1–124. Available online: https://www.mckinsey.com/featured-insights/internet-of-things/our-insights/the-internet-of-things-how-to-capture-the-value-of-iot# (accessed on 13 July 2020).
3. Khan, R.; Khan, S.U.; Zaheer, R.; Khan, S. Future internet: The internet of things architecture, possible applications and key challenges. In Proceedings of the 2012 10th International Conference on Frontiers of Information Technology, Islamabad, India, 17–19 December 2012; pp. 257–260.
4. Yang, Z.; Yue, Y.; Yang, Y.; Peng, Y.; Wang, X.; Liu, W. Study and application on the architecture and key technologies for IoT. In Proceedings of the 2011 International Conference on Multimedia Technology, Hangzhou, China, 26–28 July 2011; pp. 747–751.

5. Atzori, L.; Iera, A.; Morabito, G. The internet of things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805. [CrossRef]
6. Torkaman, A.; Seyyedi, M. Analyzing IoT reference architecture models. *Int. J. Comput. Sci. Softw. Eng.* **2016**, *5*, 154.
7. Chaqfeh, M.A.; Mohamed, N. Challenges in middleware solutions for the internet of things. In Proceedings of the 2012 International Conference on Collaboration Technologies And Systems (CTS), Denver, CO, USA, 21–25 May 2012; pp. 21–26.
8. Moustafa, N.; Creech, G.; Sitnikova, E.; Keshk, M. Collaborative anomaly detection framework for handling big data of cloud computing. In Proceedings of the 2017 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 14–16 November 2017; pp. 1–6.
9. Moustafa, N.; Choo, K.K.R.; Radwan, I.; Camtepe, S. Outlier Dirichlet mixture mechanism: Adversarial statistical learning for anomaly detection in the fog. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 1975–1987. [CrossRef]
10. Sicari, S.; Rizzardi, A.; Grieco, L.A.; Coen-Porisini, A. Security, privacy and trust in Internet of Things: The road ahead. *Comput. Netw.* **2015**, *76*, 146–164. [CrossRef]
11. Kolias, C.; Kambourakis, G.; Stavrou, A.; Voas, J. DDoS in the IoT: Mirai and other botnets. *Computer* **2017**, *50*, 80–84. [CrossRef]
12. Al-Garadi, M.A.; Mohamed, A.; Al-Ali, A.; Du, X.; Guizani, M. A survey of machine and deep learning methods for internet of things (IoT) security. *arXiv* **2018**, arXiv:1807.11023.
13. Kolias, C.; Stavrou, A.; Voas, J.; Bojanova, I.; Kuhn, R. Learning internet-of-things security "hands-on". *IEEE Secur. Priv.* **2016**, *14*, 37–46. [CrossRef]
14. Marsden, T.; Moustafa, N.; Sitnikova, E.; Creech, G. Probability risk identification based intrusion detection system for SCADA systems. In *International Conference on Mobile Networks and Management*; Springer: Berlin, Germany, 2017; pp. 353–363.
15. Moustafa, N.; Misra, G.; Slay, J. Generalized outlier gaussian mixture technique based on automated association features for simulating and detecting web application attacks. *IEEE Trans. Sustain. Comput.* **2018**. [CrossRef]
16. Modi, C.; Patel, D.; Borisaniya, B.; Patel, H.; Patel, A.; Rajarajan, M. A survey of intrusion detection techniques in cloud. *J. Netw. Comput. Appl.* **2013**, *36*, 42–57. [CrossRef]
17. Rizwan, R.; Khan, F.A.; Abbas, H.; Chauhdary, S.H. Anomaly detection in wireless sensor networks using immune-based bioinspired mechanism. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 684952. [CrossRef]
18. Moustafa, N.; Creech, G.; Slay, J. Anomaly detection system using beta mixture models and outlier detection. In *Progress in Computing, Analytics and Networking*; Springer: Berlin, Germany, 2018; pp. 125–135.
19. 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]
20. Mitchell, R.; Chen, I.R. A survey of intrusion detection techniques for cyber-physical systems. *ACM Comput. Surv. (CSUR)* **2014**, *46*, 55. [CrossRef]
21. Mishra, A.; Nadkarni, K.; Patcha, A. Intrusion detection in wireless ad hoc networks. *IEEE Wirel. Commun.* **2004**, *11*, 48–60. [CrossRef]
22. Anantvalee, T.; Wu, J. A survey on intrusion detection in mobile ad hoc networks. In *Wireless Network Security*; Springer: Berlin, Germany, 2007; pp. 159–180.
23. Kumar, S.; Dutta, K. Intrusion detection in mobile ad hoc networks: Techniques, systems, and future challenges. *Secur. Commun. Netw.* **2016**, *9*, 2484–2556. [CrossRef]
24. Sfar, A.R.; Natalizio, E.; Challal, Y.; Chtourou, Z. A roadmap for security challenges in the Internet of Things. *Digit. Commun. Netw.* **2018**, *4*, 118–137. [CrossRef]
25. Keshk, M.; Moustafa, N.; Sitnikova, E.; Creech, G. Privacy preservation intrusion detection technique for SCADA systems. In Proceedings of the 2017 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 14–16 November 2017; pp. 1–6.
26. Zhao, K.; Ge, L. A survey on the internet of things security. In Proceedings of the 2013 Ninth International Conference on Computational Intelligence and Security, Leshan, China, 14–15 December 2013; pp. 663–667.
27. Kumar, J.S.; Patel, D.R. A survey on internet of things: Security and privacy issues. *Int. J. Comput. Appl.* **2014**, *90*. [CrossRef]

28. Suo, H.; Wan, J.; Zou, C.; Liu, J. Security in the internet of things: A review. In Proceedings of the 2012 International Conference on Computer Science and Electronics Engineering, Hangzhou, China, 23–25 March 2012; Volume 3, pp. 648–651.
29. Kouicem, D.E.; Bouabdallah, A.; Lakhlef, H. Internet of things security: A top-down survey. *Comput. Netw.* **2018**, *141*, 199–221. [CrossRef]
30. Benkhelifa, E.; Welsh, T.; Hamouda, W. A critical review of practices and challenges in intrusion detection systems for IoT: Toward universal and resilient systems. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 3496–3509. [CrossRef]
31. 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]
32. Granjal, J.; Monteiro, E.; Silva, J.S. Security for the internet of things: A survey of existing protocols and open research issues. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 1294–1312. [CrossRef]
33. Zarpelao, B.B.; Miani, R.S.; Kawakani, C.T.; de Alvarenga, S.C. A survey of intrusion detection in Internet of Things. *J. Netw. Comput. Appl.* **2017**, *84*, 25–37. [CrossRef]
34. Xiao, L.; Wan, X.; Lu, X.; Zhang, Y.; Wu, D. IoT security techniques based on machine learning. *arXiv* **2018**, arXiv:1801.06275.
35. 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]
36. Mishra, P.; Varadharajan, V.; Tupakula, U.; Pilli, E.S. A detailed investigation and analysis of using machine learning techniques for intrusion detection. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 686–728. [CrossRef]
37. 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]
38. Lawal, M.A.; Shaikh, R.A.; Hassan, S.R. Security analysis of network anomalies mitigation schemes in IoT networks. *IEEE Access* **2020**, *8*, 43355–43374. [CrossRef]
39. Garg, S.; Kaur, K.; Batra, S.; Kaddoum, G.; Kumar, N.; Boukerche, A. A multi-stage anomaly detection scheme for augmenting the security in IoT-enabled applications. *Future Gener. Comput. Syst.* **2020**, *104*, 105–118. [CrossRef]
40. Garg, S.; Kaur, K.; Kumar, N.; Kaddoum, G.; Zomaya, A.Y.; Ranjan, R. A hybrid deep learning-based model for anomaly detection in cloud datacenter networks. *IEEE Trans. Netw. Serv. Manag.* **2019**, *16*, 924–935. [CrossRef]
41. Mirsky, Y.; Doitshman, T.; Elovici, Y.; Shabtai, A. Kitsune: An ensemble of autoencoders for online network intrusion detection. *arXiv* **2018**, arXiv:1802.09089.
42. Sethi, P.; Sarangi, S.R. Internet of things: Architectures, protocols, and applications. *J. Electr. Comput. Eng.* **2017**, *2017*. [CrossRef]
43. Wu, M.; Lu, T.J.; Ling, F.Y.; Sun, J.; Du, H.Y. Research on the architecture of Internet of Things. In Proceedings of the 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), Chengdu, China, 20–22 August 2010; Volume 5.
44. Tan, L.; Wang, N. Future internet: The internet of things. In Proceedings of the 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), Chengdu, China, 20–22 August 2010; Volume 5.
45. ITU, T. Telecommunication Standardization Sector of ITU. *Annex C RTP Payload Format H* **1993**, *261*, 108–113.
46. Weyrich, M.; Ebert, C. Reference architectures for the internet of things. *IEEE Softw.* **2015**, *33*, 112–116. [CrossRef]
47. Fremantle, P. A Reference Architecture for the Internet of Things. WSO2 White Paper. 2015. Available online: https://docs.huihoo.com/wso2/wso2-whitepaper-a-reference-architecture-for-the-internet-of-things.pdf (accessed on 13 July 2020).
48. Green, J. *The Internet of Things Reference Model*; Internet of Things World Forum: Geneva, Switzerland, 2014; pp. 1–12.
49. Notra, S.; Siddiqi, M.; Gharakheili, H.H.; Sivaraman, V.; Boreli, R. An experimental study of security and privacy risks with emerging household appliances. In Proceedings of the 2014 IEEE Conference on Communications and Network Security, San Francisco, CA, USA, 29–31 October 2014; pp. 79–84.
50. Banerjee, A.; Venkatasubramanian, K.K.; Mukherjee, T.; Gupta, S.K.S. Ensuring safety, security, and sustainability of mission-critical cyber–physical systems. *Proc. IEEE* **2011**, *100*, 283–299. [CrossRef]

51. AlTawy, R.; Youssef, A.M. Security tradeoffs in cyber physical systems: A case study survey on implantable medical devices. *IEEE Access* **2016**, *4*, 959–979. [CrossRef]
52. Wamba, S.F.; Anand, A.; Carter, L. A literature review of RFID-enabled healthcare applications and issues. *Int. J. Inf. Manag.* **2013**, *33*, 875–891. [CrossRef]
53. Malasri, K.; Wang, L. Securing wireless implantable devices for healthcare: Ideas and challenges. *IEEE Commun. Mag.* **2009**, *47*, 74–80. [CrossRef]
54. Keshk, M.; Sitnikova, E.; Moustafa, N.; Hu, J.; Khalil, I. An Integrated Framework for Privacy-Preserving based Anomaly Detection for Cyber-Physical Systems. *IEEE Trans. Sustain. Comput.* **2019**. [CrossRef]
55. Bertino, E.; Islam, N. Botnets and internet of things security. *Computer* **2017**, *50*, 76–79. [CrossRef]
56. Neshenko, N.; Bou-Harb, E.; Crichigno, J.; Kaddoum, G.; Ghani, N. Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2702–2733. [CrossRef]
57. Faruki, P.; Bharmal, A.; Laxmi, V.; Ganmoor, V.; Gaur, M.S.; Conti, M.; Rajarajan, M. Android security: A survey of issues, malware penetration, and defenses. *IEEE Commun. Surv. Tutor.* **2014**, *17*, 998–1022. [CrossRef]
58. Huang, J.; Zhang, X.; Tan, L.; Wang, P.; Liang, B. Asdroid: Detecting stealthy behaviors in android applications by user interface and program behavior contradiction. In Proceedings of the 36th International Conference on Software Engineering, Hyderabad, India, 31 May–7 June 2014; pp. 1036–1046.
59. Alaba, F.A.; Othman, M.; Hashem, I.A.T.; Alotaibi, F. Internet of Things security: A survey. *J. Netw. Comput. Appl.* **2017**, *88*, 10–28. [CrossRef]
60. Bekara, C. Security issues and challenges for the IoT-based smart grid. *Procedia Comput. Sci.* **2014**, *34*, 532–537. [CrossRef]
61. Steinhubl, S.R.; Muse, E.D.; Topol, E.J. The emerging field of mobile health. *Sci. Transl. Med.* **2015**, *7*, 283rv3. [CrossRef]
62. Stergiou, C.; Psannis, K.E.; Kim, B.G.; Gupta, B. Secure integration of IoT and cloud computing. *Future Gener. Comput. Syst.* **2018**, *78*, 964–975. [CrossRef]
63. Lee, K.; Murray, D.; Hughes, D.; Joosen, W. Extending sensor networks into the cloud using amazon web services. In Proceedings of the 2010 IEEE International Conference on Networked Embedded Systems for Enterprise Applications, Suzhou, China, 25–26 November 2010; pp. 1–7.
64. Botta, A.; De Donato, W.; Persico, V.; Pescapé, A. Integration of cloud computing and internet of things: A survey. *Future Gener. Comput. Syst.* **2016**, *56*, 684–700. [CrossRef]
65. Bhattasali, T.; Chaki, R.; Chaki, N. Secure and trusted cloud of things. In Proceedings of the 2013 Annual IEEE India Conference (INDICON), Mumbai, India, 13–15 December 2013; pp. 1–6.
66. Subashini, S.; Kavitha, V. A survey on security issues in service delivery models of cloud computing. *J. Netw. Comput. Appl.* **2011**, *34*, 1–11. [CrossRef]
67. Zhou, W.; Jia, Y.; Peng, A.; Zhang, Y.; Liu, P. The effect of IoT new features on security and privacy: New threats, existing solutions, and challenges yet to be solved. *IEEE Internet Things J.* **2018**, *6*, 1606–1616. [CrossRef]
68. Ronen, E.; Shamir, A.; Weingarten, A.O.; O'Flynn, C. IoT goes nuclear: Creating a ZigBee chain reaction. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; pp. 195–212.
69. Jing, Q.; Vasilakos, A.V.; Wan, J.; Lu, J.; Qiu, D. Security of the Internet of Things: Perspectives and challenges. *Wirel. Netw.* **2014**, *20*, 2481–2501. [CrossRef]
70. Karlof, C.; Sastry, N.; Wagner, D. TinySec: A link layer security architecture for wireless sensor networks. In Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, Baltimore, MD, USA, 3–5 November 2004; pp. 162–175.
71. Garg, S.; Kaur, K.; Kaddoum, G.; Ahmed, S.H.; Jayakody, D.N.K. SDN-based secure and privacy-preserving scheme for vehicular networks: A 5G perspective. *IEEE Trans. Veh. Technol.* **2019**, *68*, 8421–8434. [CrossRef]
72. 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]
73. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. A survey on sensor networks. *IEEE Commun. Mag.* **2002**, *40*, 102–114. [CrossRef]

74. Abdul-Ghani, H.A.; Konstantas, D.; Mahyoub, M. A comprehensive IoT attacks survey based on a building-blocked reference model. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 355–373.
75. Khattab, A.; Jeddi, Z.; Amini, E.; Bayoumi, M. *RFID Security: A Lightweight Paradigm*; Springer: Berlin, Germany, 2016.
76. Fan, X.; Susan, F.; Long, W.; Li, S. Security Analysis of Zigbee. 2017. Available online: https://courses.csail.mit.edu/6.857/2017/project/17.pdf (accessed on 13 July 2020).
77. Lee, K.; Lee, J.; Zhang, B.; Kim, J.; Shin, Y. An enhanced Trust Center based authentication in ZigBee networks. In *International Conference on Information Security and Assurance*; Springer: Berlin, Germany, 2009; pp. 471–484.
78. Dini, G.; Tiloca, M. Considerations on security in zigbee networks. In Proceedings of the 2010 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, Newport Beach, CA, USA, 7–9 June 2010; pp. 58–65.
79. Vidgren, N.; Haataja, K.; Patino-Andres, J.L.; Ramirez-Sanchis, J.J.; Toivanen, P. Security threats in ZigBee-enabled systems: Vulnerability evaluation, practical experiments, countermeasures, and lessons learned. In Proceedings of the 2013 46th Hawaii International Conference on System Sciences, Maui, HI, USA, 7–10 January 2013; pp. 5132–5138.
80. Kolias, C.; Kambourakis, G.; Stavrou, A.; Gritzalis, S. Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 184–208. [CrossRef]
81. IEEE Computer Society LAN/MAN Standards Committee. IEEE Standard for Information Technology-Telecommunication and Information Exchange between Systems-Local and Metropolitan Area Networks-Specific Requirements Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment1: Radio Resource Measurement of Wireless LANs. 2009. Available online: http://standards.ieee.org/getieee802/download/802.11n-2009.pdf (accessed on 13 July 2020).
82. Bicakci, K.; Tavli, B. Denial-of-Service attacks and countermeasures in IEEE 802.11 wireless networks. *Comput. Stand. Interfaces* **2009**, *31*, 931–941. [CrossRef]
83. Cope, P.; Campbell, J.; Hayajneh, T. An investigation of Bluetooth security vulnerabilities. In Proceedings of the 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 9–11 January 2017; pp. 1–7.
84. Lonzetta, A.M.; Cope, P.; Campbell, J.; Mohd, B.J.; Hayajneh, T. Security vulnerabilities in Bluetooth technology as used in IoT. *J. Sens. Actuator Netw.* **2018**, *7*, 28. [CrossRef]
85. Hassan, S.S.; Bibon, S.D.; Hossain, M.S.; Atiquzzaman, M. Security threats in Bluetooth technology. *Comput. Secur.* **2018**, *74*, 308–322. [CrossRef]
86. Liu, Y.; Cheng, C.; Gu, T.; Jiang, T.; Li, X. A lightweight authenticated communication scheme for smart grid. *IEEE Sens. J.* **2015**, *16*, 836–842. [CrossRef]
87. Singh, M.M.; Adzman, K.A.A.K.; Hassan, R. Near Field Communication (NFC) Technology Security Vulnerabilities and Countermeasures. *Int. J. Eng. Technol.* **2018**, *7*, 298–305.
88. Roland, M.; Langer, J.; Scharinger, J. Security vulnerabilities of the NDEF signature record type. In Proceedings of the 2011 Third International Workshop on Near Field Communication, Hagenberg, Austria, 22 February 2011; pp. 65–70.
89. Amin, Y.M.; Abdel-Hamid, A.T. A comprehensive taxonomy and analysis of IEEE 802.15. 4 attacks. *J. Electr. Comput. Eng.* **2016**, *2016*, 4.
90. Amin, Y.M.; Abdel-Hamid, A.T. Classification and analysis of IEEE 802.15. 4 PHY layer attacks. In Proceedings of the 2016 International Conference on Selected Topics in Mobile & Wireless Networking (MoWNeT), Cairo, Egypt, 11–13 April 2016; pp. 1–8.
91. Amin, Y.M.; Abdel-Hamid, A.T. Classification and analysis of IEEE 802.15. 4 MAC layer attacks. In Proceedings of the 2015 11th International Conference on Innovations in Information Technology (IIT), Dubai, UAE, 1–3 November 2015; pp. 74–79.
92. Mayzaud, A.; Badonnel, R.; Chrisment, I. A Taxonomy of Attacks in RPL-based Internet of Things. *Int. J. Netw. Secur.* **2016**, *18*, 459–473.
93. Cao, Z.; Hu, J.; Chen, Z.; Xu, M.; Zhou, X. Feedback: Towards dynamic behavior and secure routing forwireless sensor networks. In Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA'06), Vienna, Austria, 18–20 April 2006; Volume 2, pp. 160–164.
94. Sen, J. Security in wireless sensor networks. *Wirel. Sens. Netw. Curr. Status Future Trends* **2012**, *407*, 53–57.

95. Hummen, R.; Hiller, J.; Wirtz, H.; Henze, M.; Shafagh, H.; Wehrle, K. 6LoWPAN fragmentation attacks and mitigation mechanisms. In Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks, Budapest, Hungary, 17–19 April 2013; pp. 55–66.
96. Vacca, J.R. *Computer and Information Security Handbook*; Steve Elliot: Sydney, Australia, 2012.
97. Keshk, M.; Turnbull, B.; Moustafa, N.; Vatsalan, D.; Choo, K.K.R. A Privacy-Preserving Framework based Blockchain and Deep Learning for Protecting Smart Power Networks. *IEEE Trans. Ind. Inf.* **2020**, *16*, 5110–5118. [CrossRef]
98. Liu, C.; Yang, J.; Chen, R.; Zhang, Y.; Zeng, J. Research on immunity-based intrusion detection technology for the internet of things. In Proceedings of the 2011 Seventh International Conference on Natural Computation, Shanghai, China, 26–28 July 2011; Volume 1, pp. 212–216.
99. 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.
100. Kasinathan, P.; Costamagna, G.; Khaleel, H.; Pastrone, C.; Spirito, M.A. An IDS framework for internet of things empowered by 6LoWPAN. In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, Berlin, Germany, 4–8 November 2013; pp. 1337–1340.
101. Oh, D.; Kim, D.; Ro, W. A malicious pattern detection engine for embedded security systems in the Internet of Things. *Sensors* **2014**, *14*, 24188–24211. [CrossRef]
102. Keshk, M.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Privacy-preserving big data analytics for cyber-physical systems. *Wireless Netw.* **2018**, *2018*, 1–9. [CrossRef]
103. Debar, H. An introduction to intrusion-detection systems. *Proc. Connect* **2000**, *2000*.
104. Scarfone, K.; Mell, P. *Guide to Intrusion Detection and Prevention Systems (Idps)*; Technical report; National Institute of Standards and Technology: Gaithersburg, MA, USA, 2012.
105. Amaral, J.P.; Oliveira, L.M.; Rodrigues, J.J.; Han, G.; Shu, L. Policy and network-based intrusion detection system for IPv6-enabled wireless sensor networks. In Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, Australia, 10–14 June 2014; pp. 1796–1801.
106. Raza, S.; Wallgren, L.; Voigt, T. SVELTE: Real-time intrusion detection in the Internet of Things. *Ad Hoc Netw.* **2013**, *11*, 2661–2674. [CrossRef]
107. D'Agostini, G. A multidimensional unfolding method based on Bayes' theorem. *Nucl. Instrum. Methods Phys. Res. Sect. A Accel. Spectrometers Detect. Assoc. Equip.* **1995**, *362*, 487–498. [CrossRef]
108. Panda, M.; Patra, M.R. Network intrusion detection using naive bayes. *Int. J. Comput. Sci. Netw. Secur.* **2007**, *7*, 258–263.
109. Mukherjee, S.; Sharma, N. Intrusion detection using naive Bayes classifier with feature reduction. *Procedia Technol.* **2012**, *4*, 119–128. [CrossRef]
110. Agrawal, S.; Agrawal, J. Survey on anomaly detection using data mining techniques. *Procedia Comput. Sci.* **2015**, *60*, 708–713. [CrossRef]
111. Swarnkar, M.; Hubballi, N. OCPAD: One class Naive Bayes classifier for payload based anomaly detection. *Expert Syst. Appl.* **2016**, *64*, 330–339. [CrossRef]
112. Box, G.E.; Tiao, G.C. *Bayesian Inference in Statistical Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2011; Volume 40.
113. Ng, A.Y.; Jordan, M.I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems*; 2002; pp. 841–848. Available online: https://ai.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf (accessed on 13 July 2020).
114. Soucy, P.; Mineau, G.W. A simple KNN algorithm for text categorization. In Proceedings of the 2001 IEEE International Conference on Data Mining, San Jose, CA, USA, 29 November–2 December 2001; pp. 647–648.
115. Deng, Z.; Zhu, X.; Cheng, D.; Zong, M.; Zhang, S. Efficient kNN classification algorithm for big data. *Neurocomputing* **2016**, *195*, 143–148. [CrossRef]
116. Adetunmbi, A.O.; Falaki, S.O.; Adewale, O.S.; Alese, B.K. Network intrusion detection based on rough set and k-nearest neighbour. *Int. J. Comput. ICT Res.* **2008**, *2*, 60–66.
117. Li, L.; Zhang, H.; Peng, H.; Yang, Y. Nearest neighbors based density peaks approach to intrusion detection. *Chaos Solitons Fractals* **2018**, *110*, 33–40. [CrossRef]
118. Su, M.Y. Real-time anomaly detection systems for Denial-of-Service attacks by weighted k-nearest-neighbor classifiers. *Expert Syst. Appl.* **2011**, *38*, 3492–3498. [CrossRef]

119. 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**. [CrossRef]
120. 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*. [CrossRef]
121. Kotsiantis, S.B.; Zaharakis, I.; Pintelas, P. Supervised machine learning: A review of classification techniques. *Emerg. Artif. Intell. Appl. Comput. Eng.* **2007**, *160*, 3–24.
122. Du, W.; Zhan, Z. Building decision tree classifier on private data. In *Proceedings of the IEEE International Conference on Privacy, Security and Data Mining*; Australian Computer Society, Inc.: Sydney, Australia, 2002; Volume 14, pp. 1–8.
123. Quinlan, J.R. Induction of decision trees. *Mach. Learn.* **1986**, *1*, 81–106. [CrossRef]
124. Kotsiantis, S.B. Decision trees: A recent overview. *Artif. Intell. Rev.* **2013**, *39*, 261–283. [CrossRef]
125. Goeschel, K. Reducing false positives in intrusion detection systems using data-mining techniques utilizing support vector machines, decision trees, and naive Bayes for off-line analysis. In Proceedings of the SoutheastCon 2016, Norfolk, VA, USA, 30 March–3 April 2016; pp. 1–6.
126. Kim, G.; Lee, S.; Kim, S. A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Syst. Appl.* **2014**, *41*, 1690–1700. [CrossRef]
127. Alharbi, S.; Rodriguez, P.; Maharaja, R.; Iyer, P.; Subaschandrabose, N.; Ye, Z. Secure the internet of things with challenge response authentication in fog computing. In Proceedings of the 2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC), San Diego, CA, USA, 10–12 December 2017; pp. 1–2.
128. Tong, S.; Koller, D. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.* **2001**, *2*, 45–66.
129. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer Science & Business Media: Berlin, Germany, 2013.
130. Miranda, C.; Kaddoum, G.; Bou-Harb, E.; Garg, S.; Kaur, K. A collaborative security framework for software-defined wireless sensor networks. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 2602–2615. [CrossRef]
131. Liu, Y.; Pi, D. A novel kernel svm algorithm with game theory for network intrusion detection. *KSII Trans. Internet Inf. Syst.* **2017**, *11*. [CrossRef]
132. Hu, W.; Liao, Y.; Vemuri, V.R. Robust Support Vector Machines for Anomaly Detection in Computer Security. ICMLA. 2003; pp. 168–174. Available online: https://web.cs.ucdavis.edu/~vemuri/papers/rvsm.pdf (accessed on 13 July 2020).
133. Wagner, C.; François, J.; Engel, T. Machine learning approach for ip-flow record anomaly detection. In *International Conference on Research in Networking*; Springer: Berlin, Germany, 2011; pp. 28–39.
134. Garg, S.; Kaur, K.; Kaddoum, G.; Gagnon, F.; Kumar, N.; Han, Z. Sec-IoV: A multi-stage anomaly detection scheme for internet of vehicles. In Proceedings of the ACM MobiHoc Workshop on Pervasive Systems in the IoT Era, Catania, Italy, 2 July 2019; pp. 37–42.
135. Torres, J.M.; Comesaña, C.I.; García-Nieto, P.J. Machine learning techniques applied to cybersecurity. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 2823–2836. [CrossRef]
136. Ioannou, C.; Vassiliou, V. Classifying Security Attacks in IoT Networks Using Supervised Learning. In Proceedings of the 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), Santorini, Greece, 29–31 May 2019; pp. 652–658.
137. Lin, K.C.; Chen, S.Y.; Hung, J.C. Botnet detection using support vector machines with artificial fish swarm algorithm. *J. Appl. Math.* **2014**, *2014*. [CrossRef]
138. Woźniak, M.; Graña, M.; Corchado, E. A survey of multiple classifier systems as hybrid systems. *Inf. Fusion* **2014**, *16*, 3–17. [CrossRef]
139. Illy, P.; Kaddoum, G.; Moreira, C.M.; Kaur, K.; Garg, S. Securing fog-to-things environment using intrusion detection system based on ensemble learning. In Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 15–18 April 2019; pp. 1–7.
140. Domingos, P.M. A few useful things to know about machine learning. *Commun. ACM* **2012**, *55*, 78–87. [CrossRef]
141. Zhang, H.; Liu, D.; Luo, Y.; Wang, D. *Adaptive Dynamic Programming for Control: Algorithms And Stability*; Springer Science & Business Media: Berlin, Germany, 2012.

142. Baba, N.M.; Makhtar, M.; Fadzli, S.A.; Awang, M.K. Current Issues in Ensemble Methods and Its Applications. *J. Theor. Appl. Inf. Technol.* **2015**, *81*, 266.
143. Santana, L.E.; Silva, L.; Canuto, A.M.; Pintro, F.; Vale, K.O. A comparative analysis of genetic algorithm and ant colony optimization to select attributes for an heterogeneous ensemble of classifiers. In Proceedings of the IEEE Congress on Evolutionary Computation, Barcelona, Spain, 18–23 July 2010; pp. 1–8.
144. Aburomman, A.A.; Reaz, M.B.I. A novel SVM-kNN-PSO ensemble method for intrusion detection system. *Appl. Soft Comput.* **2016**, *38*, 360–372. [CrossRef]
145. Gaikwad, D.; Thool, R.C. Intrusion detection system using bagging ensemble method of machine learning. In Proceedings of the 2015 International Conference on Computing Communication Control and Automation, Pune, India, 26–27 February 2015; pp. 291–295.
146. Reddy, R.R.; Ramadevi, Y.; Sunitha, K. Enhanced anomaly detection using ensemble support vector machine. In Proceedings of the 2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC), Chirala, India, 23–25 March 2017; pp. 107–111.
147. Bosman, H.H.; Iacca, G.; Tejada, A.; Wörtche, H.J.; Liotta, A. Ensembles of incremental learners to detect anomalies in ad hoc sensor networks. *Ad Hoc Netw.* **2015**, *35*, 14–36. [CrossRef]
148. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
149. Cutler, D.R.; Edwards, T.C., Jr.; Beard, K.H.; Cutler, A.; Hess, K.T.; Gibson, J.; Lawler, J.J. Random forests for classification in ecology. *Ecology* **2007**, *88*, 2783–2792. [CrossRef] [PubMed]
150. Chang, Y.; Li, W.; Yang, Z. Network intrusion detection based on random forest and support vector machine. In Proceedings of the 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, China, 21–24 July 2017; Volume 1, pp. 635–638.
151. Zhang, J.; Zulkernine, M. A hybrid network intrusion detection technique using random forests. In Proceedings of the First International Conference on Availability, Reliability and Security (ARES'06), Vienna, Austria, 20–22 April 2006; p. 8.
152. Doshi, R.; Apthorpe, N.; Feamster, N. Machine learning ddos detection for consumer internet of things devices. In Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 24 May 2018; pp. 29–35.
153. Meidan, Y.; Bohadana, M.; Shabtai, A.; Ochoa, M.; Tippenhauer, N.O.; Guarnizo, J.D.; Elovici, Y. Detection of unauthorized IoT devices using machine learning techniques. *arXiv* **2017**, arXiv:1709.04647.
154. Jain, A.K. Data clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* **2010**, *31*, 651–666. [CrossRef]
155. Hartigan, J.A.; Wong, M.A. Algorithm AS 136: A k-means clustering algorithm. *J. R. Stat. Society. Ser. C Appl. Stat.* **1979**, *28*, 100–108. [CrossRef]
156. Bhuyan, M.H.; Bhattacharyya, D.K.; Kalita, J.K. Network anomaly detection: Methods, systems and tools. *IEEE Commun. Surv. Tutor.* **2013**, *16*, 303–336. [CrossRef]
157. Kanjanawattana, S. A Novel Outlier Detection Applied to an Adaptive K-Means. *Int. J. Mach. Learn. Comput.* **2019**, *9*. [CrossRef]
158. Muniyandi, A.P.; Rajeswari, R.; Rajaram, R. Network anomaly detection by cascading k-Means clustering and C4. 5 decision tree algorithm. *Procedia Eng.* **2012**, *30*, 174–182. [CrossRef]
159. 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 International Conference on Dependable, Autonomic and Secure Computing, 15th International Conference on Pervasive Intelligence and Computing, 3rd International Conference 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.
160. Hoang, D.H.; Nguyen, H.D. Detecting Anomalous Network Traffic in IoT Networks. In Proceedings of the 2019 21st International Conference on Advanced Communication Technology (ICACT), Pyeong Chang, Korea, 17–20 February 2019; pp. 1143–1152.
161. Hoang, D.H.; Nguyen, H.D. A PCA-based method for IoT network traffic anomaly detection. In Proceedings of the 2018 20th International Conference on Advanced Communication Technology (ICACT), Chuncheon-si, Korea, 11–14 February 2018; pp. 381–386.
162. Zhang, B.; Liu, Z.; Jia, Y.; Ren, J.; Zhao, X. Network intrusion detection method based on PCA and Bayes algorithm. *Secur. Commun. Netw.* **2018**, *2018*. [CrossRef]

163. Moustafa, N.; Turnbull, B.; Choo, K.K.R. An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. *IEEE Internet Things J.* **2018**, *6*, 4815–4830. [CrossRef]
164. Ahmad, A.; Dey, L. A k-mean clustering algorithm for mixed numeric and categorical data. *Data Knowl. Eng.* **2007**, *63*, 503–527. [CrossRef]
165. Nweke, H.F.; Teh, Y.W.; Al-Garadi, M.A.; Alo, U.R. Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. *Expert Syst. Appl.* **2018**, *105*, 233–261. [CrossRef]
166. De Coninck, E.; Verbelen, T.; Vankeirsbilck, B.; Bohez, S.; Simoens, P.; Demeester, P.; Dhoedt, B. Distributed neural networks for Internet of Things: The Big-Little approach. In *International Internet of Things Summit*; Springer: Berlin, Germany, 2015; pp. 484–492.
167. Yousefi-Azar, M.; Varadharajan, V.; Hamey, L.; Tupakula, U. Autoencoder-based feature learning for cyber security applications. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 3854–3861.
168. Hinton, G.E. A practical guide to training restricted Boltzmann machines. In *Neural Networks: Tricks of the Trade*; Springer: Berlin, Germany, 2012; pp. 599–619.
169. Hiromoto, R.E.; Haney, M.; Vakanski, A. A secure architecture for IoT with supply chain risk management. In Proceedings of the 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Bucharest, Romania, 21–23 September 2017; Volume 1, pp. 431–435.
170. Zhang, Q.; Yang, L.T.; Chen, Z.; Li, P. A survey on deep learning for big data. *Inf. Fusion* **2018**, *42*, 146–157. [CrossRef]
171. 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]
172. Fadlullah, Z.M.; Tang, F.; Mao, B.; Kato, N.; Akashi, O.; Inoue, T.; Mizutani, K. State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2432–2455. [CrossRef]
173. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
174. Hermans, M.; Schrauwen, B. Training and analysing deep recurrent neural networks. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 190–198.
175. Pascanu, R.; Gulcehre, C.; Cho, K.; Bengio, Y. How to construct deep recurrent neural networks. *arXiv* **2013**, arXiv:1312.6026.
176. Torres, P.; Catania, C.; Garcia, S.; Garino, C.G. An analysis of recurrent neural networks for botnet detection behavior. In Proceedings of the 2016 IEEE biennial congress of Argentina (ARGENCON), Buenos Aires, Argentina, 15–17 June 2016; pp. 1–6.
177. Almiani, M.; AbuGhazleh, A.; Al-Rahayfeh, A.; Atiewi, S.; Razaque, A. Deep recurrent neural network for IoT intrusion detection system. *Simul. Model. Pract. Theory* **2019**, *101*, 102031. [CrossRef]
178. Guo, T.; Xu, Z.; Yao, X.; Chen, H.; Aberer, K.; Funaya, K. Robust Online Time Series Prediction with Recurrent Neural Networks. In Proceedings of the 3Rd IEEE/Acm International Conference on Data Science and Advanced Analytics, (Dsaa 2016), Montreal, QC, Canada, 17–19 October 2016; pp. 816–825.
179. Qin, Y.; Song, D.; Chen, H.; Cheng, W.; Jiang, G.; Cottrell, G. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv* **2017**, arXiv:1704.02971.
180. Malhotra, P.; Vig, L.; Shroff, G.; Agarwal, P. *Long Short Term Memory Networks for Anomaly Detection in Time Series*; Presses Universitaires de Louvain: Louvain-la-Neuve, Belgium, 2015; Volume 89, pp. 89–94.
181. Shipmon, D.T.; Gurevitch, J.M.; Piselli, P.M.; Edwards, S.T. Time series anomaly detection; detection of anomalous drops with limited features and sparse examples in noisy highly periodic data. *arXiv* **2017**, arXiv:1708.03665.
182. Bontemps, L.; McDermott, J.; Le-Khac, N.A. Collective anomaly detection based on long short-term memory recurrent neural networks. In *International Conference on Future Data and Security Engineering*; Springer: Berlin, Germany, 2016; pp. 141–152.
183. Zhu, L.; Laptev, N. Deep and confident prediction for time series at uber. In Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, USA, 18–21 November 2017; pp. 103–110.

184. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; The MIT Press: Cambridge, MA, USA, 2016.
185. Chen, X.W.; Lin, X. Big data deep learning: Challenges and perspectives. *IEEE Access* **2014**, *2*, 514–525. [CrossRef]
186. Ciresan, D.C.; Meier, U.; Masci, J.; Gambardella, L.M.; Schmidhuber, J. Flexible, high performance convolutional neural networks for image classification. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Barcelona, Spain, 16–22 July 2011.
187. Scherer, D.; Müller, A.; Behnke, S. Evaluation of pooling operations in convolutional architectures for object recognition. In *International Conference on Artificial Neural Networks*; Springer: Berlin, Germany, 2010; pp. 92–101.
188. Chen, Y.; Zhang, Y.; Maharjan, S. Deep learning for secure mobile edge computing. *arXiv* **2017**, arXiv:1709.08025.
189. McLaughlin, N.; Martinez del Rincon, J.; Kang, B.; Yerima, S.; Miller, P.; Sezer, S.; Safaei, Y.; Trickel, E.; Zhao, Z.; Doupé, A.; et al. Deep android malware detection. In Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy, Scottsdale, AZ, USA, 22–24 March 2017; pp. 301–308.
190. Wang, W.; Zhu, M.; Zeng, X.; Ye, X.; Sheng, Y. Malware traffic classification using convolutional neural network for representation learning. In Proceedings of the 2017 International Conference on Information Networking (ICOIN), Da Nang, Vietnam, 11–13 January 2017; pp. 712–717.
191. Mohammadi, M.; Al-Fuqaha, A.; Sorour, S.; Guizani, M. Deep learning for IoT big data and streaming analytics: A survey. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2923–2960. [CrossRef]
192. Mayuranathan, M.; Murugan, M.; Dhanakoti, V. Best features based intrusion detection system by RBM model for detecting DDoS in cloud environment. *J. Ambient Intell. Hum. Comput.* **2019**, 1–11. [CrossRef]
193. Fiore, U.; Palmieri, F.; Castiglione, A.; De Santis, A. Network anomaly detection with the restricted Boltzmann machine. *Neurocomputing* **2013**, *122*, 13–23. [CrossRef]
194. Hinton, G.E.; Osindero, S.; Teh, Y.W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef]
195. Li, Y.; Ma, R.; Jiao, R. A hybrid malicious code detection method based on deep learning. *Int. J. Secur. Its Appl.* **2015**, *9*, 205–216. [CrossRef]
196. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *2*, 2672–2680.
197. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved techniques for training gans. *Adv. Neural Inf. Process. Syst.* **2016**, 2234–2242.
198. Kuncheva, L.I. *Combining Pattern Classifiers: Methods and Algorithms*; John Wiley & Sons: Hoboken, NJ, USA, 2014.
199. 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, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6.
200. McHugh, J. Testing intrusion detection systems: A critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM Trans. Inf. Syst. Secur. (TISSEC)* **2000**, *3*, 262–294. [CrossRef]
201. 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. A Glob. Perspect.* **2016**, *25*, 18–31. [CrossRef]
202. Stolfo, S.J.; Fan, W.; Lee, W.; Prodromidis, A.; Chan, P.K. Cost-based modeling for fraud and intrusion detection: Results from the JAM project. In Proceedings of the DARPA Information Survivability Conference and Exposition, DISCEX'00, Hilton Head, SC, USA, 25–27 January 2000; Volume 2, pp. 130–144.
203. Sharafaldin, I.; Gharib, A.; Lashkari, A.H.; Ghorbani, A.A. Towards a reliable intrusion detection benchmark dataset. *Softw. Netw.* **2018**, *2018*, 177–200. [CrossRef]
204. Shiravi, A.; Shiravi, H.; Tavallaee, M.; Ghorbani, A.A. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Comput. Secur.* **2012**, *31*, 357–374. [CrossRef]
205. Nehinbe, J.O. A simple method for improving intrusion detections in corporate networks. In *International Conference on Information Security and Digital Forensics*; Springer: Berlin, Germany, 2009; pp. 111–122.

206. Bhuyan, M.H.; Bhattacharyya, D.K.; Kalita, J.K. Towards Generating Real-life Datasets for Network Intrusion Detection. *IJ Netw. Secur.* **2015**, *17*, 683–701.
207. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy, ICISSP, Funchal, Portugal, 22–24 January 2018; pp. 108–116. Available online: https://www.scitepress.org/Papers/2018/66398/66398.pdf (accessed on 13 July 2020).
208. 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.
209. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset. *Future Gener. Comput. Syst.* **2019**, *100*, 779–796. [CrossRef]
210. Pa, Y.M.P.; Suzuki, S.; Yoshioka, K.; Matsumoto, T.; Kasama, T.; Rossow, C. IoTPOT: Analysing the rise of IoT compromises. In Proceedings of the 9th USENIX Workshop on Offensive Technologies (WOOT 15), Washington, DC, USA, 10–11 August 2015.
211. Xiao, L.; Wan, X.; Lu, X.; Zhang, Y.; Wu, D. IoT security techniques based on machine learning: How do IoT devices use AI to enhance security? *IEEE Signal Process. Mag.* **2018**, *35*, 41–49. [CrossRef]