IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 27, 2020, accepted June 20, 2020, date of publication June 29, 2020, date of current version July 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3005643

Towards a Secure Internet of Things: A Comprehensive Study of Second Line Defense Mechanisms

KAMALDEEP[1], MAITREYEE DUTTA[1], AND JORGE GRANJAL[2], (Member, IEEE)
[1]National Institute of Technical Teachers Training and Research, Chandigarh 160019, India
[2]Centre for Informatics and Systems of the University of Coimbra, 3030-790 Coimbra, Portugal

Corresponding author: Jorge Granjal (jgranjal@dei.uc.pt)

This work was supported in part by the MobiWise Project: From Mobile Sensing to Mobility Advising under Grant P2020 SAICTPAC/0011/2015, in part by the COMPETE 2020, Portugal 2020-POCI, European Union's ERDF, and in part by the Portuguese Foundation for Science and Technology (FCT).

ABSTRACT The Internet of Things (IoT) exemplifies a large network of sensing and actuating devices that have penetrated into the physical world enabling new applications like smart homes, intelligent transportation, smart healthcare and smart cities. Through IoT, these applications have consolidated in the modern world to generate, share, aggregate and analyze large amount of security-critical and privacy sensitive data. As this consolidation gets stronger, the need for security in IoT increases. With first line of defense strategies like cryptography being unsuited due to the resource constrained nature, second line of defense mechanisms are crucial to ensure security in IoT networks. This paper presents a comprehensive study of existing second line of defense mechanisms for standardized protocols in IoT networks. The paper analyzes existing mechanisms in three aspects: Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS) and Intrusion Response Systems (IRS). We begin by providing an overview of standardized protocol stack, its layers and defensive security systems in IoT. From there, we build our narrative by presenting an extended taxonomy of IDS, IPS and IRS classifying them on their techniques, deployment, attacks, datasets, evaluation metrics and data pre-processing methods. We then thoroughly review, compare and analyze the research proposals in this context, considering the unique characteristics involved in these systems. Based on the extensive analysis of the existing defensive security systems, the paper also identifies open research challenges and directions for effective design of such systems for IoT networks, which could guide future research in the area.

INDEX TERMS Internet of Things, intrusion detection, intrusion prevention, intrusion response, Internet of Things Security, standardized protocols.

I. INTRODUCTION

The Internet of things (IoT) is touted to be one of the key enablers of the next revolution in the digital world. The IoT allows to connect everyday objects (also referred to as things) to the Internet, by equipping them with identifying, sensing, actuating, networking and processing capabilities. Such capabilities allows objects with sensing and actuating capabilities to communicate with one another, and with other devices and services over the Internet, in order to accomplish tasks in the context of IoT applications. In recent years, IoT applications have increased tremendously with some of them being already deployed at various levels. These applications include smart homes, wearable technology, smart grid, smart cities, smart healthcare, smart agriculture, among many others [1].

The IoT is poised for explosive growth, with about 125 billion smart devices connected to the Internet by 2030 [2]. IoT was coined by British entrepreneur Kevin Ashton in the year 1999 while commencing his work at MIT AutoID Center. Since then, IoT has seen exponential growth as it is fuelled and overlapped with other traditional technologies like Wireless Sensor Networks (WSN), Cyber-Physical Systems (CPS), Radio Frequency Identification (RFID) technology, Near Field Communication (NFC) and Internet Protocol Version 6 (IPv6) [3], to name a few. Simply put,

The associate editor coordinating the review of this manuscript and approving it for publication was Fan Zhang.

an IoT network is made up of a great number of heterogeneous devices and technologies, produced by different vendors, for different purposes and, in practice, with different capabilities, ranging from elementary RFID tags to advanced powerful servers.

IoT technology has the potential to make human life better informed, healthier, more productive, and more connected. Despite such tremendous potential, IoT introduces new challenges for security and privacy. As per reports by Open Web Application Security Project (OWASP) [4], IoT security is challenged by constrained resources, in particular in what respects memory and energy, limited computational power, the usage of insecure operating system, insufficient authentication and authorization, lack of transport encryption, insecure network access and weak interfaces, among others [5]. Besides, the usage of open architectures motivate intruders to spread malicious content such as botnets. In the year 2016, the famous Mirai Botnet [6] targeted IoT devices such as IP Cameras, baby monitors, printers, home routers and gateways and, in consequence, several Internet services in North America and Europe were brought down. Since then, many variations and advancements of this botnet have evolved, focusing on the exploitation of unpatched IoT devices. Recently, Gartner also predicted that IoT attacks will represent 25% of all cyber attacks by 2020 [7]. These reports show that effect of intrusions in IoT is unavoidable and, thus, that a unified security framework is needed to enforce the security policies and control intrusions.

The most important layer of protection for IoT is undoubtedly securing the data, network and communication among IoT devices and with the Internet. Often referred to as the first line of defense, this protection layer includes firewalls and cryptographic primitives of authentication, encryption, access control and secure key management, among others [8]. However, when such measures are either absent or broken, a number of attacks can be triggered against IoT devices, networks and applications. Also, such measures cannot prevent IoT networks from intrusions. Most of these intrusions aim to disrupt the routing, availability, access control and other services, by targeting specific vulnerabilities in IoT protocols. Hence, unlike in WSN environments, IoT networks are susceptible to attacks both from inside the wireless network as well as from the Internet. Therefore, the usage of a second line of defense, particularly Intrusion Detection Systems (IDSs), is deemed important for detecting malicious intrusions. IDS is any hardware or software that identifies and detects intrusions. An extension of IDS are Intrusion Prevention Systems (IPS), often referred to as inline IDS, and such systems aim to detect and prevent intrusions in real time. However, both IDS and IPS are useless without Intrusion Response Systems (IRS), which implement security countermeasures to monitor system performance and even identify and handle potential intrusions. IDS, IPS and IRS systems can collectively be termed as defensive security systems for intrusions.

Though there has been a tremendous growth in the intrusion defense technologies for traditional Internet and WSN communications environments, research efforts for IoT are, so far, inadequate, due to the unique characteristics of IoT networks and the constraints of sensing and actuating devices. While most of the attacks are similar to that in WSN, defensive systems developed for WSN cannot be used directly in IoT. The reasons are global accessibility, Internet integration and the difference in protocol stacks and technologies of the IoT devices. Consequently, the attack surface in IoT has increased, with attacks from the Internet in addition to that which already exist in closed WSN environments.

In the context of the issues discussed above, our focus in the article is to investigate IDS, IPS and IRS for IoT systems, both by performing a detailed state of the art analysis, and also by identifying open research issues which could provide future avenues for research. Keeping this in mind, we present an exhaustive review on more than 65 IDS, IPS and IRS solutions in IoT, while identifying and discussing the limitations of existing solutions and emphasizing future directions. The contributions of this paper are as follows:

- Compared to other survey papers, this paper considers, in the analysis, the most significant defensive security features for ensuring security in the standardized protocol stack of IoT.
- An extended taxonomy and preliminary analysis of the types of IDS, on the basis of various characteristics of intrusion detection, which enable the researchers to enhance the design of IDSs in IoT.
- A detailed review of the state of the art on research proposals in defensive security systems (IDS, IPS and IRS) in IoT networks, their significance and challenges in implementation.
- Identification of contributions of those research proposals by introducing representative technique in context of each characteristic followed by open issues and challenges to achieve effective intrusion detection.
- Motivation for integrating intrusion prevention and response systems in IoT with an elaborate discussion on the open challenges and possible strategies for future research work in the area.

A. PAPER OUTLINE

Our discussion in the survey is performed as follows. Section II introduces standardized protocols for IoT and presents the concept of defensive security systems, together with its significance in IoT networks. In addition, this section places this survey in existing body of knowledge by examining some relevant reviews on intrusion detection in IoT. Section III presents the extended taxonomy of IDS and introduces representative technique in each category along with challenges and issues. Section IV presents the review of state of the art proposals along the taxonomy of defensive security systems in IoT, which are analyzed and discussed in Section V. Section VI discusses issues, challenges and future directions for research in the area and, finally, we conclude the survey in Section VII.

II. IoT: STANDARDIZED PROTOCOL STACK AND LAYERS

In this section, we analyze the protocols and communication layers for IoT. Such protocols are fundamental to extend the Internet communications infrastructure to encompass constrained sensing and actuating devices, thus enabling what we refer to as IoT. Also, to reduce the subjectiveness surrounding IoT, we constrain ourselves to standardized protocols of IoT as presented in [9], [10]. Consequently, the communication stack consists of six layers namely, *physical, data link, adaptation, network and routing, transport and application layers*. As shown in Table 1, the protocols at these layers include IEEE 802.15.4 [11] at the physical and MAC layers, IPv6 over Low power Wireless Personal Area Network (6LoWPAN) [12] at the adaptation layer, IPv6 Routing Protocol for low power and lossy networks (RPL) [13] and IPv6 at the network layer, and Constrained Application Protocol (CoAP) [14] at the application layer, as we proceed to discuss each in detail.

TABLE 1. IoT standardized protocol stack.

IoT Layer	IoT Protocol
Application Layer (Layer 6)	CoAP [14]
Transport Layer (Layer 5)	TCP [17], UDP [18]
Network and Routing Layer (Layer 4)	IPv6 [3], RPL [13]
Adaptation Layer (Layer 3)	6LoWPAN [12]
Data Link Layer (Layer 2)	IEEE 802.15.4 [11]
Physical Layer (Layer 1)	

IEEE 802.15.4 is a wireless communication standard that defines physical and data link layer for low rate wireless personal area networks. It has become the de facto standard of link layer communication in the IoT protocol stack. This standard can be used with many other upper layer protocols such as Zigbee, WirelessHART and 6LoWPAN. Each of these protocols extend IEEE 802.15.4 for defining the upper layers. IEEE 802.15.4 suffers from MAC layer attacks like collision attacks and back-off manipulation attack and also from physical layer attacks like power analysis attacks, jamming and exhaustion attacks. Jamming and exhaustion attacks are types of Denial-of-Service (DoS) attack which easily target IoT devices since they are resource constrained [15].

At the network and routing layer, in 2012, the Routing over Low power and Lossy networks (ROLL) group developed the RPL which has been standardized by the IETF in RFC 6550 [13]. A major objective of RPL is to minimize energy consumption and allow for varied traffic flows like one-to-one, one-to-many and many-to-one communications. Although some cryptography-based security mechanisms have been proposed for RPL, they only protect it from external threats, particularly preventing from analysis of the communication. IoT attacks can be triggered from insider or outside of network. The internal routing attacks may be topology attacks, like modifying the route, or performance attacks, like flooding and dropping. Besides that, the inherent operation of RPL has rules for network performance optimization like rank calculation, local repair etc. Among the attacks on such operations we find the rank attack, local repair attack and version number attack [16].

To realize the adaptation of IP packets over Low Power and Lossy Networks (LLNs), the IETF 6LoWPAN working group was formed in 2007, to draft specifications for transmitting IPv6 packets over IEEE 802.15.4 networks [12]. The packet size in IEEE 802.15.4 networks is restricted to 127 bytes, whereas IPv6 packets are at least 1280 bytes. Thus to counter this situation, 6LoWPAN defines mechanisms of header compression, fragmentation and reassembly to adapt IPv6 packets over IEEE 802.15.4 networks. Though it allows IPv6 packets to route over IEEE 802.15.4 networks, 6LoWPAN mechanisms often suffer a bottleneck in processing and forwarding fragmented packets that further lead to problems of buffer overflow in constrained devices. UDP, being an unreliable, connectionless and lighter protocol, has been accepted as de facto standard in IoT. But, UDP has inherent weakness whereby attackers can launch DDoS attacks, one them being the UDP flooding attack. The most promising standard application layer protocol for small IoT devices is CoAP [14]. CoAP is an open standard, and is better suited for IoT than HTTP, running on top of IP and aiming to be inter-operable with the traditional web. CoAP is a specialized web transfer protocol deployed as an alternative to HTTP for use with constrained nodes and LLNs, and works on the top of UDP and IP. To address the security concern, CoAP uses Datagram Transport Layer Security (DTLS) [8] which is an adaptation of Transport Layer Security (TLS) protocol. DTLS provides the same security as TLS, while over UDP. However, DTLS being an end-to-end security solution, is implemented at the transport layer and thus, its security is not consolidated with CoAP protocol itself, which makes CoAP susceptible to a number of internal and external attacks. Among external attacks, we find Man in The Middle (MiTM) proxying attacks, amplification attacks (which turns a small packet into a large packet), spoofing attacks and cross-layer attacks, which are able to bypass firewalls [14]. In internal attacks, an attacker may try to subvert normal functioning of CoAP protocol and even initiate timing attacks on constrained nodes of IoT. Hence, end to end security provided by DTLS must be complemented with appropriate intrusion detection and prevention approaches.

Within this standardized protocol stack, the lack of appropriate first line of defense security measures (like cryptography) creates concerns. Thus, there is a need to develop second line of defense solutions and determining exactly how IoT network must be secured. Therefore, this section described an overview of IoT standard protocols and their security issues. As IoT networks are very diverse, there is a non-exhaustive list of protocols and technologies which may be seen in current scenarios. As we proceed to discuss in next section, we focus only on defense security solutions for those developed specifically for the standardized protocols mentioned in Table 1.

A. IoT DEFENSIVE SECURITY SYSTEMS (IDS, IPS AND IRS)

Defensive security systems for IoT include IDS, IPS and IRS. In order to prevent (IPS) or react (IRS) to an intrusion, the intrusion must first be detected. The technology that allows intrusions to be detected is the IDS. When we look at standard intrusion detection, it is essentially defined as the process of monitoring, detecting and identifying malicious activities of a computer system or a network. These activities are called intrusions that aim to gain unauthorized access to a system or a network. The history of IDS dates back to 1980 when James Anderson, a civilian contractor published a paper, ''Computer Security Threat Monitoring and Surveillance'', which laid the foundation for detecting intrusions in a network [19]. Since then, system and networks that have been protected using IDS range from local area networks (LANs) and wide area networks (WANs) to ad hoc networks, wireless local area (WLAN) and wireless personal area networks (WPANs). A typical IDS includes three generic components, namely monitoring, detection and reaction. Monitoring component analyze the behaviour of traffic flows. The detection component detects any suspicious behavior and informs the reaction component of any detected occurrence. The reaction component raises an alarm or reports to the network administrator. It is also useful to consider how an IDS can be classified, and in this respect we encounter centralized or distributed systems, network or host-based, anomaly or signature based, flat and clustered, among others. Section III classifies IDS and proposes an extended taxonomy of IDS characteristics. Normally, an IDS is a passive actor that may generate alerts but does not prevent or respond to a detected attack.

IPSs, on the other hand, are considered extensions of IDS because they monitor system or network activity and attempt to stop/block intrusions. Unlike IDS, IPS are placed in-line and are able to proactively prevent intrusions that are detected. More precisely, IPS can take actions such as dropping malicious packets, sending alarms, resetting the connection, correcting transmission errors, cleaning unwanted network and transport layer options etc.

IRS are responsible to generate reactive responses for intrusions. IDS detects intrusions whereas IRS blocks intrusions and mitigate the effect of attacks. Similar to IPS, IRS work with IDS and works closely with the detection engine of IDS. An effective response must consist of pre-planned defensive measures that may include an incident response team and measures to collect logs for future use. Another responsibility of the IRS is to keep track of latest threats, assess the damage and make decision to recover from the intrusion [20].

Although, IDS, IPS and IRS have somewhat different functions, they share and even update a common base of knowledge. As shown in Figure 1, the different phases of defensive security include prevention, detection and response to intrusions. Many times, they work together or can be integrated into one defensive system to maximize the protection of the network.

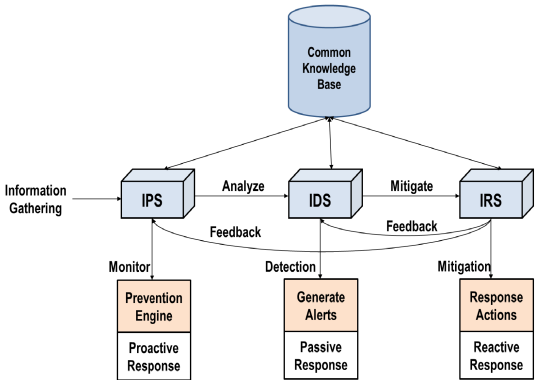


FIGURE 1. From IPS to IRS.

Now, let us discuss the need of defensive security systems in IoT. As an extremely heterogeneous technology, security is a critical aspect of IoT on multiple levels of securing data, communication and networks. Attacks on IoT have become extremely targeted and sophisticated. As already discussed, first line of defense mechanisms like cryptography are insufficient due to the resource constrained nature of IoT devices which limits their ability to host and implement sophisticated cryptographic algorithms in real time. Also, the ad hoc nature of IoT networks lets devices connect to each other at runtime, typically for shorter durations, consequently creating a collaborative network. The ideal security protection will involve defensive security systems in place to cover these IoT devices and networks. To strengthen security capabilities, these defensive security systems are typically implemented at the edge of the network or on IoT devices with their complexity trimmed down without compromising the robustness. Also, the easy integration of such systems with low communication overhead within the network allow these second line of defensive mechanisms meet the requirements of constrained devices in IoT networks. The goals to be achieved by such IoT defensive security systems include prevention, response and detection of intrusions in realtime. Currently, the IoT is plagued because of constrained memory, power and computation availability in IoT devices. Consequently, the defensive security systems that are deployed in powerful nodes in traditional networks, are harder to implement on IoT devices. Secondly, the architecture and network topology in IoT is extremely different where the border router is presumed to be always accessible and devices in the network are globally identifiable and accessible by an IPv6 address. Finally, the protocols in IoT are entirely different from traditional networks, which poses numerous challenges in the design of defensive mechanisms for IoT.

The need for defensive security systems in IoT is also evident from relevant works on IoT security [9], [16], [21]–[25], which discuss IoT attacks and emphasize on their detection in IoT networks. For example, [9], [24] lists attacks in 6LoWPAN-based IoT communication environments. Similarly, Mayzaud *et al.* [16] establish a taxonomy of the attacks on RPL, and categorize them according to attacks targeting

network resources, attacks modifying the network topology and attacks related to network traffic. The authors in [26] survey Sybil attacks and their respective defense schemes in IoT. In this work, the authors classify Sybil attacks in three classes based on the attacker's capabilities and also compared respective detection schemes. The authors in [25] specify IDS for RPL intrusions as the second line of defense, the first being traditional cryptography. They analyze the different detection methodologies, detection data, system architectures and also the intrusion response of IDS, the third line of defense. In the discussion that follow, we explicitly state the defensive security system (IDS, IPS and IRS) being considered.

B. COMPARISON WITH EXISTING SURVEYS

During the last decade, numerous attempts have been made to propose IDS, IPS and IRS for the IoT. Likewise, there are a variety of surveys that review these proposed solutions considering a taxonomy in varied degrees of depth and scope. In a brief work, the authors in [27] gave a holistic view of IDS proposed for WLAN, LAN, WSN, RFID and mobile-based networks and differentiated them from solutions proposed for IoT. The authors identified that a hybrid, interoperable and cross-layer IDS with anomaly-based intelligence integrated into the 6LoWPAN protocol stack would be a promising approach for IoT. The work in [28] reviews the IDS based countermeasures for insider attacks and proposed a baseline for IDS design in IoT. They propose a framework for IDS which is lightweight and integrated with a firewall in border router just above the adaptation layer in IoT. Likewise the authors in [29] provide comprehensive classification of IDS based on placement technique, security threat, detection method and validation strategy, and analyzed 18 papers devoted to IDS in IoT. As an extension, the work in [30] focuses on machine and deep learning methods applied for intrusion detection in IoT. However, some of the characteristics like performance metrics, location and usage frequency of IDS are further considered in the taxonomy proposed in [31]. In 2018, a critical review in [32] concentrates on recent advances in intrusion detection approaches in IoT emphasizing on IoT architecture and protocols. Both the surveys in [33], [34] refer and summarize 22 papers which point out, among other aspects, the need to design lightweight and robust IDS for IoT. Whilst this is correct, the analysis performed in such surveys suffers from the limitation that they do not consider the placement strategy, validation and usage frequency in the considered taxonomy. Two more recent taxonomies of IDSs based on machine learning detection techniques are discussed in [35], [36]. The survey by the authors in [35] discusses datasets related to IoT and Chaabouni *et al.* [36] presents more detailed and critical review of network intrusion detection solutions for IoT security. The authors concluded their survey with future research directions in NIDS for IoT. However, these taxonomies were limited to certain type of IDSs.

Therefore, these reviews focus on following points:

- **Attack Detection Technique:** A variety of intrusion detection methods with varied effectiveness exists in literature often targeting specific attacks and specific protocols.
- **Validation:** Whilst solutions have been proposed, their appropriate validation with realistic IoT datasets is necessary.
- **Lightweight and Robust IDS:** A point focused in all these surveys is that conventional IDS is not suitable for IoT networks due to the resource constraints of IoT devices as well as diverse traffic which increases attack surface beyond limits.
- **Insider Attacks:** The vast majority of work emphasize on attacks within the IoT network, there is a particular lack of works that provide defenses for outsider attacks.

Although there are a number of surveys on IDS for IoT, their diversity reflects that reviews must be extended effectively and extensively with respect to IDS characteristics and IoT. Thus, we aim to include only those security solutions that are based on standardized protocol stack [10] of IoT, as we proceed to discuss in depth later in the paper. In addition to that, the existing taxonomies lack some important IDS characteristics like data pre-processing techniques and their classifications. To the best of our knowledge, this is the first survey that considers nine different dimensions of an IDS in IoT. Specifically, none of the mentioned works discuss features and feature selection algorithms which have direct impact on accuracy of learning based IDS. Also, there is no discussion of IPS and IRS in IoT which too are crucial to contemplate defensive security in the heterogeneous IoT environments. Table 2 highlights the IoT defensive security considerations covered in comparison to previous review articles.

TABLE 2. Comparison of recent survey articles.

IoT Defensive Security Contributions		This Review	[27]	[28]	[29]	[30]	[31]	[32]	[33]	[34]	[35]	[36]
IDS Taxonomy	Detection Technique	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Placement Strategy/Architecture	✓		✓	✓		✓	✓				✓
	Location/Visibility	✓					✓	✓	✓			✓
	Usage Frequency	✓					✓					
	Validation	✓			✓							✓
	IoT Attacks	✓			✓	✓	✓			✓		✓
	Performance Evaluation Metrics	✓					✓		✓		✓	✓
	Dataset/ Database	✓									✓	✓
	Data Preprocessing and Feature Selection Technique	✓										
IPS		✓										
IRS		✓										

III. A TAXONOMY OF IDS, IPS AND IRS IN IoT

This section presents an extended and global taxonomy of IDS, IPS and IRS in IoT. We begin by classifying IDS along various approaches and detection techniques, placement methodology, information source, usage frequency, validation strategy, security threats, evaluation metrics, data sets and feature selection techniques. Later, we dwell into the various types of IPS and IRS. Figure 2 illustrates the approaches considered in our following discussion.

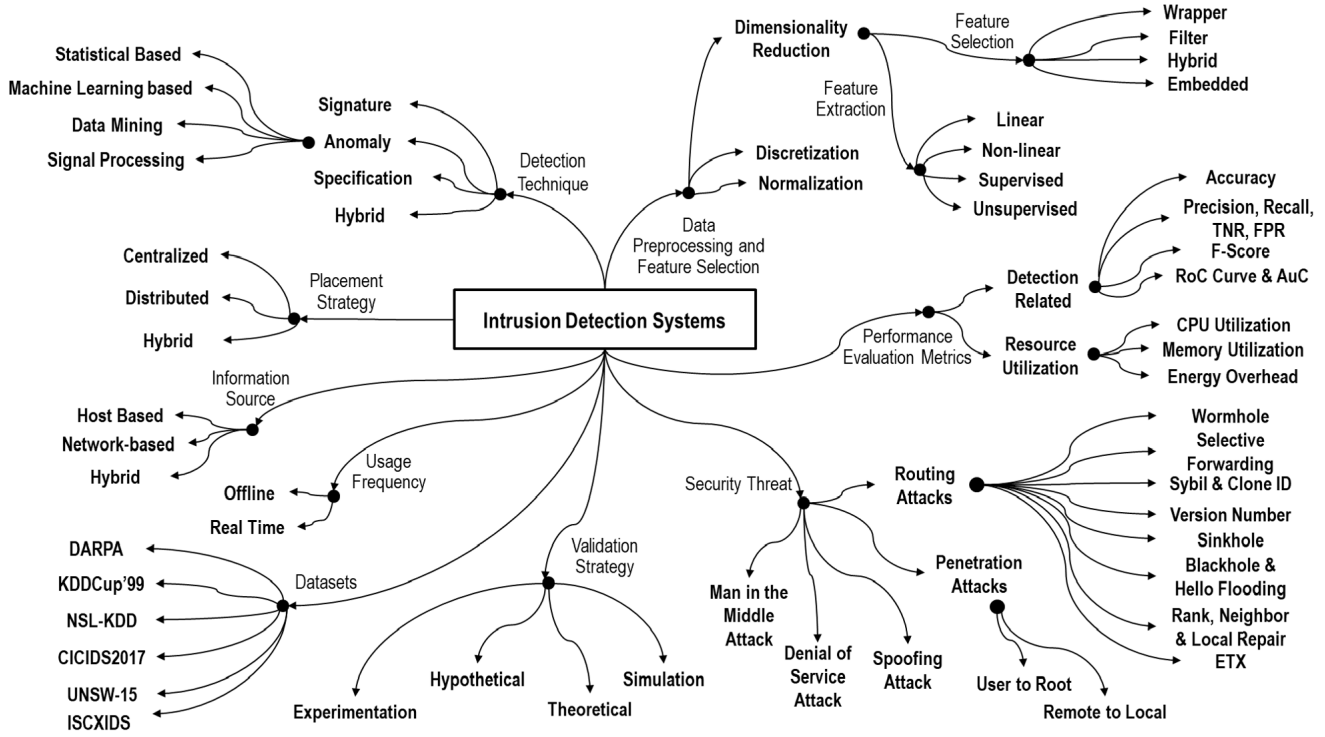


FIGURE 2. A Taxonomy of intrusion detection systems.

A. CLASSIFICATION APPROACHES OF IDS

1) ON THE BASIS OF DETECTION TECHNIQUE

IDS have different methods to detect intrusions, and certain techniques are better suited for detecting different types of intrusions. Also, we need to consider that an IDS consists of more than one of such techniques.

- **Signature or Misuse-based Detection:** In signature-based detection, an IDS has a database of known intrusions in the form of signatures, patterns or rules [37]. These specific signatures/patterns are usually predefined byte sequences in the network packet or malicious intrusion instructions. The IDS looks for these signatures/patterns in the network traffic and raises an alarm when an intrusion is detected. The term signature originated from anti-virus software, and a signature must be crafted in such a way that it is the smallest byte sequence to detect the related attack. Besides, it must be accurate and should not lead to false positives. However, signature based IDS are suited only for detecting known attacks but cannot recognize unknown or zero-day attacks [29].
- **Anomaly-based detection:** Anomaly-based IDS are capable of detecting abnormal behaviour or anomalies, by comparing communications or actions with the expected or normal behavior which is derived by regular monitoring of networks or hosts, over a period of time [38]. Such IDS typically involve three stages: the pre-processing of input data, the training stage and finally, the detection stage. This method of detection has the ability to detect a large range of malicious intrusions, including those not covered by detection signatures. Similarly to signature-based detection, anomaly detection too has its own limitations. While it detects an extensive range of malicious intrusions, anomaly-based techniques suffer from the problem of false positives. Besides, the data collected during the training period for the purpose of base lining must be non-malicious and free of errors. A number of anomaly-based techniques and models have been proposed for intrusion detection including statistical based techniques, machine learning based techniques, data mining techniques and game theory models. We proceed by discussing such approaches next:
  - *Statistical Methods:* Statistical methods use statistical properties and tests to deduce whether the current behavior deviates from the expected behavior. Such methods generate a profile representing the normal behavior (without any attack) of network traffic. Afterwards, the network is monitored and profiles are created periodically and anomalies are detected by comparison with reference profiles. Such methods can be uni-variate, multi-variate or time series/event based [37].
  - *Machine Learning (ML) Methods:* Machine Learning techniques are the techniques that involve learning from data and making predictions based on such data [39]. The first is called training stage and the latter is called testing stage. Machine learning

can be either supervised or unsupervised. In supervised learning, training is performed with labeled data, whereas in unsupervised learning there are no predefined labels. Classification and regression are supervised learning techniques, whereas clustering and association are unsupervised learning techniques. Supervised learning algorithms include Decision Trees, Logistic Regression, k-Nearest Neighbors, Naive Bayes, Linear Regression, Neural Networks, Linear Support Vector Classifier and Random Forests (RF), among others. Unsupervised learning algorithms include k-Means Clustering, Dimensionality Reduction, Principal Component Analysis and Apriori algorithm, among others. Machine learning approaches based on computational intelligence are nature and biologically inspired algorithms. To name a few, these typically include Artificial Neural Networks (ANN), Fuzzy Logic (FL), Artificial Immune System (AIS), Evolutionary Computation (EC) and Swarm Intelligence (SI).

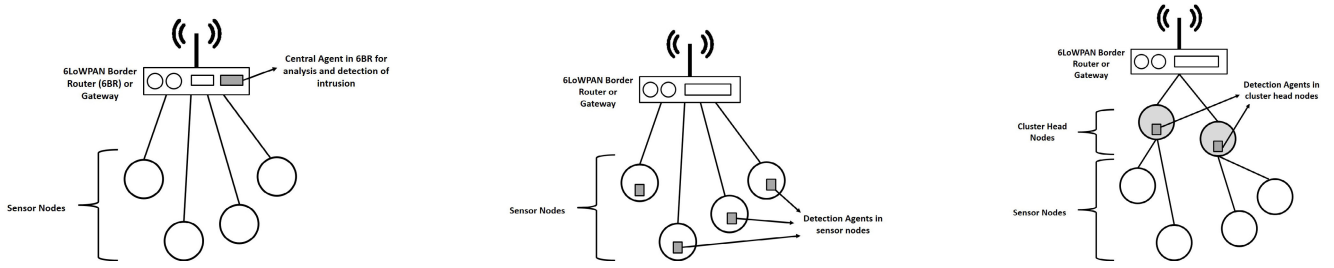
  - *Data Mining Methods:* Data mining refers to the extraction of knowledge from large amounts of data, retrieved from sources such as data warehouses or existing databases (e.g. relational, transactional or spatial) [39]. Thus, such methods rely on existing historical data. The knowledge extracted is used to define patterns in the data, and eventually to characterize data or make predictions from those patterns. For intrusion detection, the data mining engine follows a procedure of rule learning, frequent pattern mining, clustering, classification and regression. Data mining methods generate models automatically and can be used for developing generalized IDS in any computing environment. As compared to machine learning, data mining is manual and involves more human interference.
  - *Signal Processing Model:* This anomaly detection technique analyzes the traffic by using signal processing methods such as wavelets and entropy analysis. The data obtained from sensors is decomposed using wavelet transforms and the coefficients are extracted to build normal traffic model [40].

- **Specification-based detection:** Specifications are, in practice, sets of rules that formally define the legitimate behavior model of network components such as protocols and routing tables. This method detects intrusions when the network behavior deviates from this model. These rules can be a handmade model of states and transitions or statistical rules with certain conditions on normal behavior [37]. Introduced in [41], this method has the same principle as that of anomaly-based detection i.e. identification of deviations from normal behavior. However, in specification-based detection, specifications are defined manually and explicitly by a human expert or network administrator, whereas in case of anomaly-based detection method these specifications are defined by the detection system during the training phase, using various algorithms [29]. This enables such specification-based approaches to detect unknown attacks, while at the same time exhibiting low false positives.
- **Hybrid:** Hybrid detection techniques combine multiple techniques such as anomaly, specification or signature-based intrusion detection, in order to achieve more extensive and accurate detection, while with minimum manual efforts.

2) ON THE BASIS OF PLACEMENT METHODOLOGY

Placement methodology refers to how the detection, monitoring and reporting/management agents of an IDS architecture are placed and located in a network. As shown in Fig.3, in this category an IDS can be further classified as centralized, distributed and hybrid, as we proceed to discuss.

- **Centralized:** A centralized IDS employs a single module or agent to analyze events, detect intrusions and perform appropriate actions, upon the successful detection of such intrusions. In this approach, one or more monitoring modules (also called agents) may be present to collect and transmit data to the central agent. A central agent is often known as the central manager and does not have limited resources. We must also note that this entity represents a single point of failure and, as such, the security of the central agent is of cornerstone importance, and thus must be guaranteed. Also, we must note that centralized IDS are not inherently scalable, as performance may be degraded if more nodes are added to the network.
- **Distributed:** With distributed IDS solutions, the IDS intelligence required for monitoring and detecting attacks is distributed among management modules and other agents of the network. In other words, there is an IDS agent in each node of the network. These agents are partially or fully involved in analyzing the events, detecting intrusions and taking appropriate actions. This placement methodology supports better scalability and avoids single points of failure. The distributed methodology can be individualized or cooperative. In the individualized methodology, each node can detect malicious behavior of other nodes, and send notification to the reporting module in order to take appropriate action. In the cooperative method, nodes cooperatively decide whether the monitored node is malicious or not, as illustrated in Fig.3b.
- **Hybrid:** An Hybrid IDS combines the centralized and distributed IDS placement approaches, with the goal of enhancing their strengths and to avoid weaknesses. There are two methods of placement in this strategy. One is to divide the network in clusters, each having a cluster-head. The cluster-head hosts the IDS agent and has two functions: to perform normal operations like other nodes, and intrusion detection. The second method



(a) Centralized **(b)** Distributed **(c)** Hybrid

FIGURE 3. Different IDS placement methodologies in IoT.

for hybrid placement involves hosting IDS on both the border router and in network nodes. As compared to the former approach, in this case the presence of a central element is mandatory.

3) ON THE BASIS OF INFORMATION SOURCE

Another important classification of IDS considers its classification on the basis of the information source employed for intrusion detection. Some IDSs analyze network packets collected from the network backbone, in order to find attackers. Other IDS analyze information obtained from the operating system or application software. In general, we may consider the follow IDS strategies, in what respects the type of information source(s) considered:

- **Host based:** Host based IDS (HIDS) detects intrusions by collecting information from an individual system. HIDS may utilize information obtained either from the operating system or by the application software.
- **Network based:** Network based IDS (NIDS) detect intrusions by capturing and analyzing network packets. NIDS often consists of sensors deployed at strategic locations, which read and inspect network traffic in stealth mode in order to make it more difficult for attackers to determine their presence.
- **Hybrid:** A hybrid IDS relies on both information sources (network and host-based intrusion detection) for signs of intrusion.

The advantages and disadvantage of both strategies are listed in Table 3.

TABLE 3. Advantages and disadvantages of HIDS and NIDS.

Information Source	Advantages	Disadvantages
HIDS	Detects the exploitation of organization's internal equipment [42]	Incompatibility issues while monitoring different platforms like DLL in Windows OS or API calls in Linux [42]
	Can analyze end-to-end encrypted or obfuscated communications' activity	Consumes host resources
		Additional challenge in detection accuracy due to lack of context knowledge [43]
NIDS	Easy monitoring, if well-placed within the network	Utilize information only from packet headers and thus cannot detect attacks within encrypted traffic
	Easy configuration with little impact on an existing network	Limited support for scalable and fast networks [43]

4) ON THE BASIS OF USAGE FREQUENCY

This classification of an IDS is based on when audit data analysis for intrusion detection is performed. It is essentially the time elapsed between monitoring of events and their analysis, particularly by the detection component. Depending on how frequent the information flows from monitoring agents to detection components, IDS may be classified along the following strategies:

- **Offline Mode:** In offline mode, the detection component of IDS analyzes the information collected by all the monitoring agents at regular intervals. It is essentially batch-based detection, which is used to understand the attacker behavior. However, such IDS are precluded from generating active responses. Given that IoT devices generate huge amount of data continuously, this mode of detection can be useful for greater depth of coverage at non-peak times.
- **Real Time:** In real time IDS, information from the monitoring agents is continuously fed into the detection component which analyzes and detects it in real time, while the communication sessions are still in progress. Real time IDS raise an alarm as soon as intrusion is detected.

5) ON THE BASIS OF VALIDATION STRATEGY

Validation is defined as the activity of proving the validity or accuracy of proposed research. There has been a lot of emphasis on validity to ensure accuracy and correctness of work, as well as a mandate to publication content. This is also true for security proposals in the context of IoT communication protocols, as different validation strategies may be

employed and bring light to the effectiveness of a proposal in dealing with real-world challenges (e.g. security attacks). Thus, a deep insight of validation strategies also helps to analyze the maturity of works on IDS in IoT, and this motivates its consideration in the taxonomy previously discussed. The validation strategies can be classified as follows:

- **Experimentation:** This type of validation involves systematic experimental collection of data, under particular conditions and assumptions. It includes a series of experiments to test abstract models or proposed technique or system against reality [44].
- **Hypothetical:** Hypothetical validation means claiming accuracy by assuming unreal examples.
- **Theoretical:** Validating the system theoretically means to assess whether it actually measures what it purports to measure. This type of validation implies generality, whereby all the parameters that characterize the performance of system are measured. It can also be said as formal axiom-proof describing a new theory [44].
- **Simulation:** Simulation involves random generation of data according to a theoretical distribution, to determine the effectiveness of a given technique/system [44]. It is beneficial when the authors need to provide evidence of certain characteristics of their proposed work, which they cannot directly measure in real network environments.

6) ON THE BASIS OF SECURITY THREAT/ATTACKS

Abomhara and Koien [45] identify and discuss threats, exposure, attacks and vulnerabilities in the context of IoT applications. In general, a deliberate attempt to evade security services and violate security policies is referred to as an attack. In order to clarify our analysis on proposals targeting the detection of particular attacks in IoT environments, we find it useful to briefly discuss the main attacks.

Since all the devices in IoT are IP enabled with the usage of the standardized protocol stack (6LoWPAN-based), we need to consider that attacks can originate from within the network and also from the Internet. Thus, the attack surface is certainly larger and completely different from that considered in traditional proposals for security in closed and isolated WSN environments. Attacks that originate from within the network are termed as insider attacks whereas those that originate from outside the network are termed as outsider attacks.

Another useful classification for attacks is that they can also be either active or passive. Passive attacks in IoT gather data from the network without disrupting its operations. In such attacks, the attacker silently sniffs for information in the background and, in the IoT context, passive attacks may also target routing functions, where the attacker, while not disrupting routing operations, is able to infer important information on routing traffic and network topology. In active attacks, the attackers can disrupt the functionality of the network, which requires them to dispense some of its resources to carry out the attack. Active attackers in IoT tend to modify routing information to disrupt routing, launch DoS attacks by altering control messages or transmitting messages with incorrect routing information, among other situations. Thus, active attacks threaten integrity, confidentiality and availability of the data being transmitted, the devices and in general of the IoT application itself. Sometimes, an attacker launches a man-in-the-middle attack to relay or alter the communication messages between two parties. Next we identify the main attacks in this context.

- **Routing Attacks:** In routing attacks, the attacker either modifies or spoofs the information (messages) exchange in the context of the routing protocol. There are a number of routing attacks in IoT targeting the RPL protocol, a fundamental protocol for Internet-integrated WSN, as we proceed to identify:
  - *Wormhole Attack:* Wormhole attack involves at least two attackers that communicate with each other over a low-latency link, often called a tunnel, and transmit all the traffic through it [16]. This tunnel is faster than existing normal paths and creates an illusion that those two end points are actually close to each other affecting routes within the network.
  - *Selective Forwarding Attack:* In selective forwarding attacks, the attacker selectively forwards the packets, while dropping particularly-selected packets [46]. The attacker limits the suspicion on its behavior by forwarding certain control messages or packets from few non-victim nodes. Selective forwarding attacks are most effective when the attacker is explicitly included on the path of data flow [47]. An attacker emulates selective forwarding by either jamming or by causing collision on each forwarded packet. Thus, to achieve this the attacker follows the path of minimum resistance, and explicitly attempts to include itself on the data flow path [47].
  - *Clone ID and Sybil attacks:* Clone ID and Sybil attacks appear similar, since both target the creation of a number of logical identities [26]. Such attacks pose a threat to location-based routing protocols. In the case of Sybil attack, all the logical identities are copied on one physical node, whereby a large part of network gets affected with no need of deploying more physical nodes. In the case of clone ID attack, the attacker copies these logical identities on a number of physical nodes, such that they all act as clones, thereby gaining access to the traffic meant for victim node. To detect clones and minimize the effect of this attack, IDS modules must keep track of number of instances of each identity.
  - *Version Number Attack:* Version number attacks are specific to the RPL protocol, and consists on modifying the version number carried in DIO control messages [16]. These attacks target the global repair mechanism of RPL, which is based on version number itself. The attacker modifies the version number

and forces the nodes to exchange control messages on each global rebuild, thus draining their limited resources.
  - *Sinkhole Attack:* In the sinkhole attack, the attacker falsely claims to be the optimal routing path to its neighbors in order to route traffic through it. In RPL, the attacker performs this attack by faking a lower rank, usually the rank of the root node, in order to be selected as a parent by its neighbors [16]. This attack is often accompanied by a selective forwarding attack, where malicious node first attracts all traffic to itself and then selectively forwards it.
  - *Blackhole and Hello Flooding Attacks:* In the Blackhole attack, the attacker drops all the packets that are routed through it, with the goal of hindering routing or to intercept data [16]. On the other hand, Hello Flooding Attacks are triggered when an attacker broadcasts HELLO messages with optimal routing metrics and join a network.
  - *Rank, Neighbor and Local Repair Attacks:* As with Version number attacks, Rank attacks too are specific to the RPL protocol. In an RPL network, the rank of nodes increases strictly, from the root to the child nodes. In this attack, the attacker changes its rank to get selected as preferred parent, thus attracting the traffic. The effects of this attack may be sub-optimal routes, loops in routing, changes in topology and delayed packet delivery. An attacker, while performing a neighbor attack, broadcasts all DIO messages that it receives, so that its neighbor thinks that a new neighbor is in range and tries to connect to it. This attack can be considered a special case of a wormhole attack, with selective forwarding of DIO messages. This attack affects the network topology, QoS parameters and can be dangerous when combined with other attacks, as discussed in [23]. In the local repair attack, the attacker initiates the local repair mechanism and sends local repair messages periodically, in the absence of any problem in the link quality. This attack has more affect on the delivery ratio than any other attacks.
  - *Expected Transmission Count (ETX) Attack:* ETX is the expected number of transmission to transmit and acknowledge packets successfully [48], and in this attack an attacker may falsely advertise incorrect ETX values to gain access to the network, or to launch DoS attacks.
- **Denial of Service (DoS) Attacks:** The second most common and the easiest to launch is the DoS attack and when this attack is launched by multiple compromised systems on a single victim, it is termed as Distributed Denial of Service (DDoS) attacks. Also present in the IoT, DoS tries to put a node or a network out of operation by flooding it with (possibly incorrect) requests, preventing the node to accept and process legitimate requests. Since IoT devices are already resource-constrained, it becomes very easy to exhaust their resources and devoid them of the services. DoS can be initiated in almost all the layers of the communication stack. For example, in the transport layer, the attacker can send superfluous connection requests to the victim node. A detailed classification of DDoS attacks in IoT is given in [15].
- **Spoofing Attacks:** Various other classes of attacks detected in the IoT include impersonation or spoofing attack, physical tampering of devices, man-in-the-middle, data integrity, authentication and adversarial attacks.
- **Man-in-the-Middle (MiTM) Attacks:** Another class of attack is the MiTM, where the attacker actively wiretaps the communication between two nodes without their knowledge. The attacker intercepts and possibly modifies the messages between the nodes. Also, in recent times, attacking machines are usually a part of a large set of compromised machines (a botnet). Integrity attacks aim at modifying the information (routing or data) in the network.
- **Penetration Attacks:** Penetration attacks exploit vulnerabilities in the system and involve any unauthorized access or modifications to system's resources and data. In such attacks, attackers gain control of the system by exploiting a number of software flaws. In the IoT, an attacker may gain control of the device, either physically or via an application, which he is able to reverse engineer and examine for vulnerabilities. Most common types of penetration attacks are:
  - *User to Root (U2R):* In this attack, a user starts with normal user account and tries to gain privileges of administrator controls by exploiting vulnerabilities.
  - *Remote to User (R2U):* In this attack, an attacker remotely gains access to a user account on the target machine.

7) ON THE BASIS OF EVALUATION METRICS OF IDS FOR IoT

One of the first efforts to evaluate the accuracy of IDSs in a realistic environment was conducted at the MIT Lincoln Laboratory in 1998 [49]. Though the best way to evaluate IDS is to test it on real network traffic, the encryption and repeatability of real traffic makes it a non-viable solution. As already discussed, other ways of testing and validation of an IDS includes experimentation, simulation, theoretical and hypothetical. This section discusses about the various parameters (or metrics) that have been used to evaluate an IDS in the IoT. We classify these metrics into detection-related metrics and resource utilization metrics as follows:

- **Detection Related Metrics:** Intrusion detection usually involves a binary classification problem, whereby activities are either termed malicious or not malicious. Thus, statistical measures to evaluate the detection performance of an IDS are the same as that of a binary classification test. In terms of accuracy, these measures

are based on four possible states of observed activities (e.g. communications), in particular:

- True Positive (TP): A state when IDS correctly identifies an actual malicious activity as malicious.
- True Negative (TN): A state when IDS correctly rejects valid acceptable activities as being malicious.
- False positive (FP): A state when IDS incorrectly classifies an activity as malicious.
- False negative (FN): A state when IDS incorrectly classifies an activity as normal, when in reality it was an attack.

All these states are often fed to a confusion matrix, which is a table that describes the performance of a classification algorithm. Table 4 depicts a confusion matrix often used in classification algorithms employed in IDS.

TABLE 4. Confusion matrix.

		Predicted Class	
		Positive	Negative
Actual Class	Positive	TP*	FN*
	Negative	FP*	TN*

*TP, FN, FP, TN represent True Positive, False Negative, False Positive and True Negative respectively.

Intrusion detection is typically evaluated by its accuracy, detection and false positive rates. Nonetheless, other metrics of detection, in particular sensitivity, recall, precision, F1 score, time until detection and latency are also significant for the analysis. In this section, we define and discuss each of these metrics, and evaluate how such metrics and considered in the literature. The following evaluation metrics are often computed from the confusion matrix:

- *Accuracy:* termed as the correct identification of malicious activities and also correct exclusion of non-malicious activities. Thus, accuracy is defined as the ratio of true results (true positive and true negative) among all the activities examined.

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

- *Precision, Recall, Fallout and Specificity:* These metrics are also calculated from the confusion matrix, shown in Table 4, and provide more detailed analysis than just proportion of accurate classifications. Precision is the proportion of correctly detected malicious activities relative to the predicted number of malicious activities. The aim of an IDS is to achieve high precision by minimizing the number of false positives. It is calculated using the equation:

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

Recall or True positive rate (TPR) or Sensitivity or detection rate is the ratio of correctly detected malicious activities to the total number of real malicious activities and calculated as follows:

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

Fallout or false positive rate (FPR) is the ratio of incorrect detection of malicious activities relative to the total number of malicious activities.

$$Fallout = \frac{FP}{FP + TN} \tag{4}$$

Specificity or true negative rate (TNR) is the ratio of number of correctly rejected non-malicious activities to the total malicious activities.

$$Specificity = \frac{TN}{TN + FP} = 1 - FPR \tag{5}$$

Miss rate or the False Negative Rate (FNR) is the ratio of incorrectly rejected malicious activities out of all the malicious activities.

$$MissRate = \frac{FN}{FN + TP} \tag{6}$$

- *F-score:* F-score, also called F-measure or F1-score is the weighted average of the true positive rate recall) and precision. Its value ranges between [0,1], with 1 as its best value. This happen when precision and recall both are 100% and thus the IDS has zero false positives and detects 100% of the intrusions.

$$\begin{aligned} F_1 &= 2 \times \frac{1}{\frac{1}{Recall} + \frac{1}{Precision}} \\ &= 2 \times \frac{Precision \times Recall}{Precision + Recall} \end{aligned} \tag{7}$$

This metric is preferred when only one accuracy metric is desired for evaluation.

- *ROC Curve and AUC:* The Receiver Operating Characteristics (ROC) curves were created from the signal processing theory but were later extended to other domains like machine learning, data mining as well as artificial intelligence. In intrusion detection, these curves are used to visualize the relation between the TPR and FPR, and to compare the accuracy of two or more IDSs. As shown in Figure 4, this curve has false alarm rate (FPR) as the x-axis, and TPR as the y-axis. The curve R is better than the curves Q and P, as the ROC value is closer to 100%, which is the perfect detection rate. The benefits of using a ROC curve is its ability to separate error cost considerations from the IDS performance. However, a major disadvantage of ROC-curve is that small variations in FPR causes a drastic difference in TPR, when the normal traffic exceeds in large amount as compared to intrusions in the network traffic. Therefore, a new metric Area

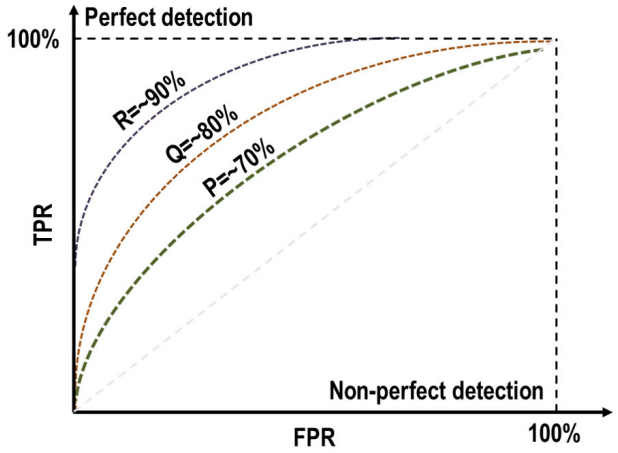


FIGURE 4. ROC curves - P, Q and R show levels of detection [50].

under ROC (AUC) was established to compare IDS. Area under ROC curve (AUC) sums up the total accuracy in a way that caters to increase in TP rate and decrease in FP rate. The range for AUC is from worst value of 0.5 to best value of 1.0.

- **Resource Utilization Metrics:** Resource utilization metrics are also referred to as performance metrics, given that they measure the performance of the system in which the IDS resides. These metrics allow to visualize whether the IDS has any detrimental effect on the system. Generally, resource utilization metrics are further classified into: CPU utilization and Memory utilization. However, in the area of wireless networks and IoT, another type of resource utilization metric that is used is the energy overhead. Since such networks are composed of constrained nodes with limited power resources, the evaluation of an IDS using this metric is quite significant. The following are the metrics considered in this context:
  - *CPU Utilization:* The percentage of CPU computation used by the IDS is referred to as CPU utilization.
  - *Memory Utilization:* The percentage of the total available memory, used by the IDS, is referred to as memory utilization. The memory, in this case, is either the RAM or ROM used in the device supporting intrusion detection.
  - *Energy Overhead:* Energy overhead is one of the prominent preconditions that the IDS must satisfy to be practically implemented in the IoT. The two metrics of CPU and memory utilization directly affect the energy overhead of an IoT node.

8) ON THE BASIS OF DATASETS USED FOR EVALUATION

In this section, we discuss different public and private datasets used for IDS evaluation in proposals related to IoT security. This classification is significant because datasets support the development and evaluation of the type of IDS being built, as such datasets contains network and host based features. The datasets for IoT, which we discuss in the next section, either use available benchmark public datasets from traditional networks or, on the other hand, create their own by installing sniffers in the network (Ad Hoc). We begin our discussion with popular network based datasets followed by host based datasets.

- **DARPA (Lincoln Laboratory 1998-99) [49], [51]**: The Defense Advanced Research Projects Agency (DARPA) 98 is one of the earliest datasets for network security analysis. The network traffic in DARPA is in the form of tcpdump format with online and off-line evaluations. The dataset which is made up of 4GB of binary data was produced in training period of 7 weeks. This dataset was enhanced to generate features in KDDCup'99 dataset.
- **KDDCup'99 [52]**: With an aim to develop machine learning algorithms for security, KDDCup'99 is a transformed version of tcpdump traces of DARPA into 42 network features. This dataset is injected with DoS, U2R, R2U and probing attacks. However, the different probability distributions and attack types in test and train data makes it suffer from unbalanced classification methods.
- **NSL-KDD [53]**: NSL-KDD is an upgraded version of KDDCup'99 to overcomes its limitations. Firstly, it removes redundant records from training and test data. Secondly, it selects variety of records from the original KDDCup'99 to achieve higher classification accuracy. Thirdly, unbalanced distributions from train and test data are removed. However, this dataset is not a perfect representative of real networks and lacks support for low footprint attacks scenarios.
- **ISCXIDS [54], [55]**: ISCX IDS dataset was designed at the Canadian Institute for Cybersecurity (CIC) in 2012, using realistic network and traffic captured over seven days. This dataset is based on the concept of profiles which contain descriptions of distribution models and multi-stage attacks. However, the dataset did not reflect the reliability of labelling process. Also, profiling of real networks can be tedious because of their complexity.
- **CICIDS2017 [56]**: Built on the abstract behaviour of 25 users based on the HTTP, HTTPS, FTP, SSH, and email protocols, a newer dataset by CIC is the CICIDS2017. B-Profile [57] approach is used to generate realistic traffic and it is analyzed with CICFlowMeter [58] with labeled flows based on source and final IP, ports, protocols and timestamps. This dataset also suffers from lack of labeling credibility and complex profiling of real networks.
- **UNSW-NB15 [59]**: Developed at the UNSW, UNSW-NB2015 dataset is created with the IXIA PerfectStorm tool. This tool generates 100GB of benign and attack traffic in the form PCAP files with a significant number of 49 novel features. This dataset includes nine attack types, namely, DoS, Exploits, Generic, Fuzzers, Analysis, Reconnaissance, Worms, Shellcode and Backdoors.

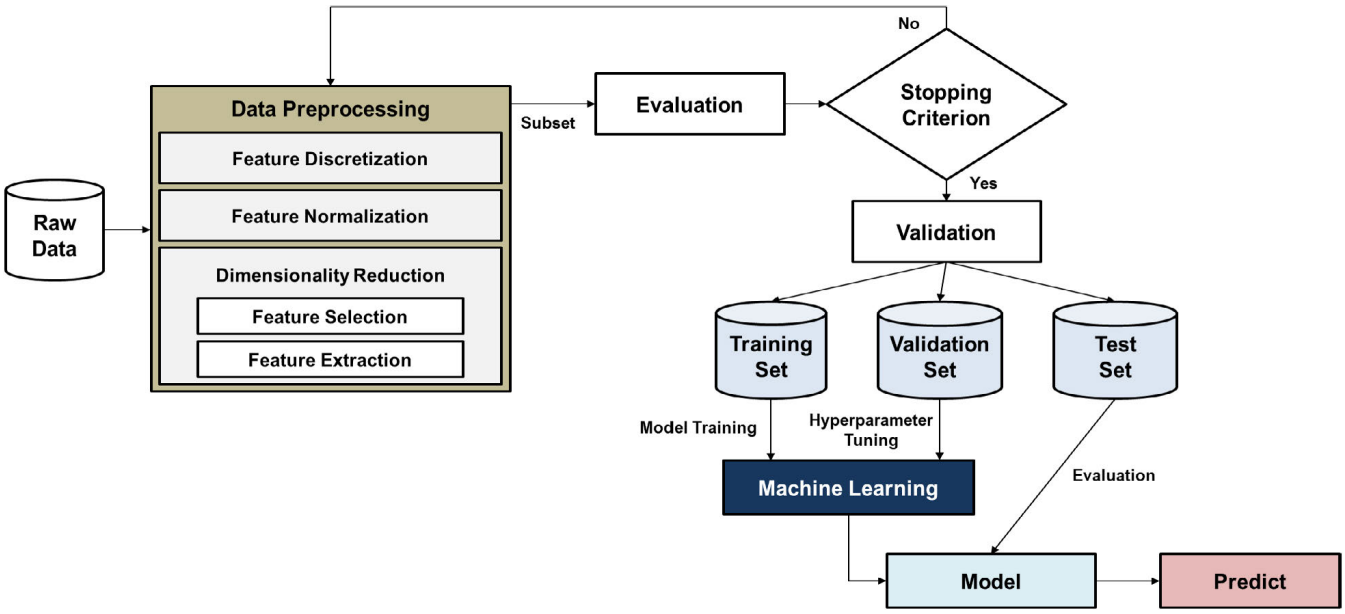


FIGURE 5. Data preprocessing in the machine learning process.

The synthetic environment used for attack generation and more complexity than KDD'99 are the major drawbacks of this dataset.

- **Sivanathan *et al.* [60] IoT Dataset**: This is one of the earliest public datasets for IoT, developed with traffic traces from a smart-campus which classifies IoT devices based on network characteristics. These IoT devices include health monitors, lights, camera and other appliances. The traffic traces are collected over a period of three weeks and analyzed to characterize statistical attributes such as signalling patterns, burstiness, data rates and activity cycles.
- **RPL-NIDDS17 [61]**: This dataset specific to 6LoWPAN networks in IoT is developed by collecting packet traces by simulating various routing attacks against RPL protocol. It is used for the training, testing and evaluation of network intrusion detection in 6LoWPAN based IoT network. This dataset is available for free use for academic research purposes and contains seven routing attacks on RPL like sybil, sinkhole, hello flooding, blackhole, clone id, local repair and selective forwarding.
- **Bezerra *et al.* [62]**: This public dataset is significant because it contains not only network traffic but also host-based features like memory and CPU usage. This dataset is generated from real IoT device profiles like surveillance camera and multimedia centre implemented on Raspberry Pi which are infected with botnet malware. The dataset covers a period with non-infected device and when infected with multiple botnet malware including Mirai, Doflo, Hajime/MC, Aidra/SC, BashLite/ST, Tsunami and Wroba [63].

9) ON THE BASIS OF DATA PRE-PROCESSING AND FEATURE SELECTION TECHNIQUES

Data pre-processing is perceived as fundamental stage in all anomaly based network intrusion detection systems. It is recognized as important because IDS deals with huge amount of data with irrelevant and redundant features, which causes slow training, higher resource consumption and poor detection rate. Thus, the effectiveness of an IDS not only depends on the detection algorithm but also on data pre-processing methods which are used to extract the features from audit data, therefore deeply influencing the quality of such data. The representation, quality and size of input data can have major impact on the accuracy and computational performance of the detection method. Specifically, utilizing a dataset with high dimensionality and redundant features can make training more difficult. Thus, pre-processing of input data prior to training phase is essential, and it involves mapping input data to feature vectors. In the context of machine learning, this process is also called feature engineering. As shown in Figure 5, data pre-processing is the basic stage in the machine learning process. Such data pre-processing techniques have been explained with analysis in [64]. We summarize the main aspects of data pre-processing techniques in our following discussion:

- **Feature Discretization:** Discretization involves conversion of continuous features into finite discrete sets. It facilitates some learning algorithms by simplifying information representation.
- **Feature Normalization:** Often called feature scaling, feature normalization standardizes the range of numerical features, with Min-max and Z-score being the most common normalization techniques.

- **Dimensionality Reduction:** The process of reducing the dimensionality of features in a dataset is termed as dimensionality reduction. It helps in removing redundant features and reducing storage and computation overhead. Dimensionality reduction can be further broken into feature selection and feature extraction, as we discuss next:
  - *Feature Selection:* The goal of feature selection is to find optimal subset of features that best characterize the data and remove irrelevant and redundant features. Feature selection enables to achieve higher accuracy of classification model, reduce computation and storage overhead and avoids over-fitting of the model. Feature selection has four different approaches:
    - Wrapper - Uses machine learning algorithms to evaluate the accuracy produced by selected features. These have high computational complexity.
    - Filter - Feature Subset selection by utilizing the characteristics of features.
    - Hybrid - The combination of the previous two approaches (wrapper and filter).
    - Embedded - Selects features during the training process.
  - *Feature Extraction:* The goal of feature extraction is to reduce the dimensionality of features by creating new features from existing ones, through linear or non-linear transformations using supervised or unsupervised learning methods. Two of the most common feature extraction algorithms are Principal Component Analysis (PCA), an unsupervised technique, and Linear Discriminant Analysis (LDA), a supervised technique.

B. CLASSIFICATION APPROACHES OF IPS

Traditionally, IDS tries to detect intrusions as they occur. Such systems are critical to defensive approach to security, but have shortcomings. IDS can only detect intrusions but cannot stop them. IPS can proactively block intrusions and focus on what an attack does which does not change even if the attack signature is changed. IPS basically deals with attack mitigation techniques and are of the following two types on the basis of deployment:

1) HOST BASED IPS (HIPS)

In HIPS, the intrusion prevention application is resident on the specific host (e.g. an IoT device) and is closely bound to operating system kernel and services, monitoring and intercepting system calls to kernel or other APIs in order to prevent intrusions as well as log them [65].

2) NETWORK BASED IPS (NIPS)

NIPS is essentially an inline IDS placed at the edge of the network to inspect incoming as well as outgoing traffic to the network. If an intrusion is detected, NIPS generates alerts and drops that packet or subsequent packets in that session [65].

Besides, both HIPS and NIPS can use either signature based or anomaly based prevention techniques. Signature based IPS inspects contents of traffic for particular signatures to detect and prevent known intrusions. On the other hand, anomaly based IPS intend to prevent intrusions by identifying abnormal rate of traffic and preventing them with various techniques like rate limiting specific protocol type, port numbers or IP addresses.

C. CLASSIFICATION APPROACHES OF IRS

Though IPS can prevent intrusions before their occurrence, an IRS responds to intrusions and tries to mitigate them [66]. IRS is crucial component in defensive security because, as with any security system, complete prevention of intrusions is impractical. Additionally, in situations when IPS misses intrusion, an alert by the IDS after detection will be futile in the absence of any reactive response. When IDS gathers intrusion information, the response component generates responses on the basis intrusion symptoms.

1) ON THE BASIS OF LEVEL OF AUTOMATION

On the basis of level of automation in generating the response, IRS is categorized as *notification, manual and automated* response systems.

- **Notification Response System:** These system notify the network/system administrator about the intrusion (typically via email messages or console alerts), who in turn select the best reactive intrusion response. However, the attacker may block and monitor email messages which makes this approach inappropriate. Also, there is a time difference between detection and response which further increases the challenges. Bro (now called Zeek) [67], a network IRS is an example of notification response system which generates alerts in the form of reports and emails.
- **Manual Response System:** Manual response systems have greater degree of automation than notification only response systems, and allow system/network administrator to trigger an action from predetermined set of responses to generate a manual response with respect to attack information. However, there is still a delay in detection time and the time that administrator takes to initiate a response. This approach also fails against fast attacks like DoS.
- **Automated Response System:** Automatic response system provide immediate response to intrusion through automated decision making process. The aim of these systems is to decrease the time gap between detection and response with high level of automation and without human intervention. Another important consideration is that an appropriate response must be generated with lower false positives and minimum uncertainty and response cost [66]. On the basis of complexity and ability to adjust, automated response

systems are further classified as *adaptive, association and expert based* response systems [68]. Adaptive-based IRS uses a feedback loop to evaluate the previous response. Association-based IRS are conventional automated response systems that generate a static response after an attack is detected. Expert IRS are more dynamic and involve series of if-then statement based on one or more metrics.

2) ON THE BASIS OF ACTIVITY OF TRIGGERED RESPONSE

On the basis of activity of triggered response, IRS is classified as being *passive and active* [69], as follows.

- **Passive:** Passive response systems notify the administrator about the intrusion and provide its information. Passive IRS does not attempt to minimize the damage caused by the attack or prevent it. Examples of passive responses include administration notifications of alarm and report generation, enabling any additional IDS, activity logging, or any other analysis tools [69].
- **Active:** The goal of active response systems is to minimize the damage done by the intruder and attempt to block/remove it. Examples of active responses on a host include denying access to files, deleting tampered file, restricting or eve disabling user account. On a network, active responses include restarting the target system, blocking ports or IP addresses and enabling additional firewall rules [69].

IV. STATE-OF-THE-ART (SoA) DEFENSIVE SECURITY PROPOSALS IN THE IoT

We start our analysis on the SoA on relevant proposals dealing with IDS in IoT with defensive security proposals. As previously discussed, our focus is on solutions targeting standard IoT protocols, illustrated in Table 2, which allow for developing IoT applications integrated with the Internet communications infrastructure. As such, the analyses proposals are candidate to support also open and inter-operable defensive security solutions. Researches on defensive security for IoT demonstrate that there are mainly three axes of works: i) IDS, ii) IPS and iii) IRS. We proceed by discussing each in detail.

A. STATE OF THE ART IN IoT-IDS

This part describes proposals of IDS in IoT, with special emphasis on the detection technique, placement methodology, information source, usage frequency, validation strategy, attacks, evaluation metrics, datasets and data pre-processing techniques. This is a novel work that considers nine different dimensions of an IDS in IoT. The reviewed proposals enable the readers to follow the evolution of this domain, from its inception to the present day. We augment our discussion with qualitative as well as quantitative analysis of proposals. Fig. 2 illustrates the proposed IDS taxonomy, we review the proposals along the taxonomy and Table 11 summarizes the relevant works on IDS in IoT, which we proceed to discuss.

1) ON THE BASIS OF DETECTION TECHNIQUE

- **Signature Based IDS** - The detection of intrusions using signatures in IoT has been discussed in [70]–[74]. The porting of open source rule-based IDS to IoT like Suricata [75] in [70], [71] and Snort [76] in [74] is not an efficient approach, since it is verified that such solutions demand a lot of storage, which frequently is not available on IoT devices. Pattern matching based detection in [72] focuses on reducing the unnecessary matching operations of the Wu-Manber algorithm, one of the fastest multiple pattern matching algorithms. Johannson and Olsson [77] implement Snort on a graphical processing unit to optimize the computationally expensive pattern matching algorithm. The authors compare the modified implementation with an unmodified version of Snort, in a closed simulation environment. As a result, a speedup of 1.3 is achieved with twice the throughput and lesser energy consumption. Using the complex event processing (CEP) technology, the authors in [73], [78] propose an IDS that filters and processes continuous data streams or events in real-time. Such a technology merges data from multiple sources to decode real time actions from complicated events. CEP suits IoT needs since it can cater to huge amounts of data with low latency. To detect the abnormal operation in CoAP, the work in [79] define verification and threshold-based rules on source IP addresses, number of CoAP requests per minute and messages from other protocols. Such preconfigured rules are implemented as policies in the memory of sensing devices and border router of the IoT network.
- **Anomaly Based IDS** - As previously discussed, anomaly IDS in IoT can be categorized as Statistical, Machine Learning, Data Mining, Signal Processing and Blockchain methods, which we proceed to analyze:
  - *Statistical IDS:* Statistical-based IDS for IoT, designed by [80]–[82], employ mathematical operations such as average and statistical tests like ANOVA and sequential probability ratio test (SPRT), to distinguish abnormal traffic from normal traffic patterns. The dependence on statistical operations and models is reduced by augmenting device normality profile with whitelist of incoming and outgoing traffic in [83]. In [84], the authors establish two energy prediction statistical models for route-over and mesh-under routing schemes in 6LoWPAN communication environments. By processing payloads as overlapping byte tuples, stronger detection is ensured in [85], as the attacker has to match normal payload statistical profile through byte substitution or padding. The works in [86] use Received Signal Strength Indicator (RSSI) which is measurement of the power in the received radio signal, convert it to distance values and compare the actual distance with the obtained values

in order to detect wormhole attacks. Based on the multi-instance feature of RPL, the authors devise algorithms to detect version number attacks in RPL networks using regular and monitoring instances of RPL in [87].

– *Machine Learning Methods:*
  * **Random Forests (RF):** Random forests are ensemble of decision trees, which merges multiple decision trees with the goal of getting more stable and accurate predictions. They can be used for both classification and regression problems. RF are simple and flexible since they do not always require much effort in hyper-parameters tuning and often produce good results even with default hyper-parameters. As an example of classification using RF in IoT, the research in [88] generate rules using random forests, which are then applied to features extracted from network traffic data with an objective to identify IoT device type from a whitelist of trustworthy devices. To reduce the probability of generating biased trees, the work in [89] divide the simple (IF condition) and complex rule generation (RF-based) into stages.
  * **SVM:** Defined by a separating hyperplane, SVM is another classification model based on supervised machine learning. Though the kernel function in SVM is its strength, choosing a good one is not always easy. To obtain an optimized model, the authors in [90] incorporate the LIBSVM library [91] and the Radial Basis Function (RBF) [92] kernel on a SDN-enabled IoT network. Bezerra *et al.* [93] perform one class classification with SVM by training the model on the server and then deploying it in IoT device. Assuming that smaller windows lead to faster identification of botnets, the authors evaluate the optimal time window size, number of instances needed for training and optimal SVM hyper-parameters. Jan *et al.* [94] compare the performance of their two-class SVM classifier on the three kernel functions (RBF, linear and polynomial) on Poisson distributed network traffic. The authors deduce that linear SVM outperform polynomial and RBF while accurately classifying traffic with four different input feature sets. The authors in [95] synthesize the literature by measuring binary and multi-class classification accuracy of SVM for anomaly detection in CoAP protocol. In this work, sigmoid kernel function achieves more accuracy in binary and linear kernel in multi-class SVM classification.
  * **FCM:** Clustering belongs to the class of unsupervised machine learning and divides the data into one or more clusters. Fuzzy clustering is a type of clustering in which one piece of data can belong to two or more clusters. FCM gives best results for overlapped datasets, and is the most widely used fuzzy clustering algorithm. FCM is deployed by Deng *et al.* [96] for detecting sybil attacks in IoT. A newer variant of FCM, Suppressed Fuzzy Clustering (SFC) which converges faster than FCM if parameters are selected reasonably is discussed in [97]. In this work, the authors divide the pre-processed IoT data as low-risk and high-risk, which is further detected using different frequencies to achieve better accuracy.
  * **Correlation Analysis:** Correlation analysis is employed to identify relationship between two variables, and can be used to detect abnormal behavior by comparing the consistency of network report with the information exchanged. To enable anomaly information exchange, a novel control message, Distress Propagation Object (DPO) for RPL is proposed in [98], [99]. In this proposal, nodes in the network monitor and grade their neighbors, isolate anomaly nodes, discard their packets and finally report to the parent node. On predefined intervals, the gateway correlates the reports gathered to check for inconsistency before raising any alert. A similar approach of correlating alerts from multiple devices to minimize the false positives is also discussed in [100]. Intelligent sensing data gatherers, Cognitive Tokens are induced in the network to deliberately attract malicious users in [101]. This work is based on error-based outlier filtering that looks only those users which fall under the category of potential anomalies.
  * **CI-Fuzzy Logic:** A Cognitive IDS based on CI paradigm developed in [102], and uses IP addresses to build IoT device profiles, but is only limited to TCP/IP based networks. The approaches in machine learning that take into consideration uncertainty and source trust are based on Subjective Logic, which is a specific type of probabilistic logic. Subjective Logic has been used to model trust networks in IoT in [103].
  * **CI-AIS:** AIS is a computational intelligence (CI) algorithm, and a major advantage of CI-based systems for intrusion detection in wireless networks is their tolerance to uncertainty and imprecision, while with less computational overhead. AIS seeks to capture aspects of the natural immune system, in which learning is aimed at differentiating self (body) and non-self (foreign pathogens) elements. The immune cells are responsible for this differentiation. Within the domain of intrusion detection in IoT, immune cells act as detectors, with attack signatures that

learn and adapt to dynamic changing environments [104], [105]. Clonal selection algorithms, a common technique in AIS has been used to evolve the detector for new and mutated attacks in [106]. Further amelioration of AIS involves delegating the training of AIS to the cloud platform [107] and grey prediction method for intrusion prediction and response [108].

* **CI-ANN and Deep Learning:** ANN, modeled on the human brain, learns by adjusting weights for correct prediction of class labels. An artificial neural network consists of connected set of input, hidden and output nodes. The algorithm is effectively a functional mapping from input values to output values. Learning in ANNs can be supervised, like multi-level perceptron (MLP), unsupervised like self-organizing maps (SOM) or even reinforced like random neural networks (RNN). Since the last decade, ANNs have been used extensively as anomaly detection techniques in the IoT network. A MLP model with three layers and a unipolar sigmoid transfer function is used to train the network with forward and backward learning algorithms in [109]. With this model, a overall accuracy of 99.4% is achieved in detecting DDoS attacks with over 10 Million packets. Roux *et al.* [110] detect intrusions by training a neural network with RSSI, reception timestamp and related radio activity of sensor nodes. This IDS is independent of underlying wireless technology, thus making it interoperable. In IoT, RNN-based IDS in [111] enjoy the advantages of lesser complexity and greater generalization, even for small training data sets. Finally, Prabavathy *et al.* [112] implement Online Sequential Extreme Learning Machine (OSELM) on distributed local fog nodes to intelligently interpret attacks from IoT traffic. The authors argue that ELM is more powerful than MLP neural network in terms of generalization power and convergence speed.
A subset of machine learning, Deep Learning (DL), often called as deep neural networks have more than one hidden layer, and delve deeper into the data. Each layer in deep learning networks train on features based on the previous layer's output. Since IoT data is heterogeneous and its scale is very large, deep learning may prove to be more appropriate to find patterns in IoT data [113]. Deep Belief Networks (DBN), Deep Neural Networks (DNN), Auto-Encoder (AE) and RNN are the most common deep learning algorithms. Contrary to ANN, RNNs are a closer representation of how signals (impulse) are transmitted between neurons. RNNs learn from past outputs and recursively call their hidden layer where the output of node is given as input to the same node. Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) Neural Network are the types of RNN. For the first time in IoT security, the author in [114] propose lightweight GRU with one hidden layer and one hidden unit and performed ten experiments for tuning each hyper parameter to an optimal value. In addition, the author in [115] use DBN with three hidden layers which are pre-trained with unsupervised learning to improve the model efficiency. The model later uses supervised learning and binary classification to detect malicious activities in the network. Deep auto-encoders (DAE) efficiently learn coding in unsupervised manner and thus have been used to pre-train the DFNN model in [116], which further classifies network observations. Meidan *et al.* [117] propose a model which learns from benign data by training DAE for each device in the network. These devices act as standalone tools for automatic detection of botnet attacks. Recently, a seminal work in [118] use deep learning for detection of routing attacks in RPL based IoT networks. A sequential model with mean squared error (MSE) loss function is used in this binary classification problem.
* **Multiple ML:** A combination of one or more machine learning algorithms may help to improve accuracy or allow detection of specific type of attacks. For example, to accurately detect low frequency attacks, two-tier classification [119] IDS is deployed where the traffic classified as normal by one classifier is fed into another classifier for accurate detection. Another approach is to implement an hybrid of supervised (C4.5 decision trees) and unsupervised (IW clustering) machine learning [120] algorithms. The author in [121] reduce the false positives of individual K-means (KMIDS) approach and decision trees-based IDs (DT-IDS) by combining them into a hybrid IDS. In step one, KM-IDS divides the network into safe zones and in step two, DT-IDS predicts a threshold on direct connections. The authors in [122] compare the accuracy of SVM, kNN and J48 classifiers over the KDD-Cup 99 dataset [52]. More recently, the work in [123] use decision tree for binary classification at layer 1 of their model and random forest for multi-class classification at layer 2 to know the type of anomaly. This work achieves 100% specificity for CICIDS [56] and UNSW-NB15 [59] datasets.

– *Data Mining Methods:* To reduce the complexity for IoT, proximity measures like the Jaccard coefficient, instead of the conventional Euclidean

distance as judging criterion, for similarity-based mining, is used in [124]. Qin *et al.* [125] argue that Exponentially Weighted Moving Average (EWMA) control charts have excellent predictive performance on the dynamically changing statistical characteristics. EWMA sets the upper and lower bounds on network statistical characteristics and an alert is generated if these features exceed the threshold interval set by EWMA.

- *Signal Processing Model:* Discrete Wavelet Transform (DWT) based anomaly behaviors analysis in IoT is performed in [126], in which wavelet coefficients are extracted from sensor reading and reconstructed, by selecting only those signals that lie within the considered thresholds.
- *Blockchain IDS:* Golomb *et al.* [127] propose the first of its kind lightweight framework (CIoTA), using the blockchain concept to perform anomaly detection. In this proposal, a trusted anomaly model is built by self-attestation and consensus among IoT devices. The blockchain incrementally updates this model and enforces trust while updating devices. However, a separate chain must be developed for each IoT model which restricts the application of this model to large industrial setting only.

- **Specification Based IDS** - To detect topology attacks, Le *et al.* [128] propose a specification-based IDS which uses a deployed backbone of monitor nodes. The detection of intrusions is based on rules specified by a finite state machine for RPL, which is implemented on all the monitor nodes. Misra *et al.* [129] propose an IDS based on the learning automata concepts, together with Service Oriented Architecture (SOA) as a system model for IoT. SOA is used as a middleware, and a threshold for each layer is specified in this proposal. The function of learning automata is to identify which packets can be discarded, and the system generates an alert when a number of requests to a layer exceeds a predefined threshold. Amaral *et al.* [130] programme selective watchdog nodes to identify intrusions, by snooping the exchanged packets in its neighborhood. The monitored messages were then matched against a set of rules, and an alert message is sent to the Event Management System (EMS). However, each watchdog node is programmed separately that depended on its location and its neighborhood traffic patterns. Grgic *et al.* [131] assert that the local detection module deployed on all nodes of the network tracked the activity of their neighbors and gauged their malevolence probabilities. Then, all nodes exchanged their estimations to calculate the final probability at the cooperative detection module. The authors claim that this probability calculation provides an advantage over other solutions, since it does not classify a node as either malicious or legitimate, but instead depicts its level of malevolence. Le *et al.* [132] extend their work in [128], in order to detect more topology attacks against RPL, such as sinkhole, neighbor and DIS attacks. The proposed system specified the RPL profile in the form of an Extended Finite State Machine (EFSM), employed with the Inductive Logic Programming (ILP) technique. ILP is a technique formed by the integration of logic programming and machine learning. The profiling of RPL is based on traces of its implementation using the Cooja simulator [133], which the authors claim to be more accurate than manual profiling of protocol from its documents. The authors in [134] detect only sinkhole attacks, but improve some critical QoS parameters which are overlooked by SVELTE [46] and INTI [135]. The QoS parameters, which are improved by this technique, are Packet drop ratio, Packet delivery ratio, Normalized overhead, Throughput and Average Energy consumption. Medjek *et al.* [136] thrive to detect the sybil mobile attack by employing trust-based RPL routing. In this proposal, the authors modify the DODAG Information Object (DIO), DAO and DIS control messages in the RPL protocol, with the goal of incorporating two new fields: maximum response delay and node ID field. All network nodes maintain a list of malicious nodes and they collaboratively monitor and evaluate trust in its neighbors' packets. The trust is composed of three components: honesty, energy and mobility and its associated weights. Kawamura *et al.* [137] develop a statistical event detection module which employs information obtained from the Network Time Protocol (NTP), a time synchronization protocol. This work is based on the principle that NTP can be used to reset the system clock by calculating the clock offset. In the wake of DDoS attack, the authors claim that large fluctuations occur in the system clock, due to time synchronization processing being skipped. Statistical deviations, particularly by calculating the coefficient of variance of the request and response delay times, besides the offset of clock, are calculated and compared with specified legitimate limits. The advantage of this scheme is that to implement it, expensive equipment or technical knowledge is not required.
- **Hybrid** - We find various works [46], [138]–[145] combining the signature and anomaly-based detection approaches, with the aim of reducing the memory overhead required to store signatures, together with the computational overhead required by anomaly-based techniques. Amin *et al.* [138] implement a hybrid IDS, as a combination of anomaly-based and pattern classifier techniques, in which two packet analyzers, i.e. one for the Internet and other for the sensor network, were deployed. Besides, this work implements the concept of data caches on base stations or cluster heads, in order to reduce the number of transmissions of sensor nodes thus increasing its lifetime. The authors further extend their work in [139] by building RIDES (Robust IDS), where they incorporate distributed pattern matching (signature based) and CUSUM control charts

(statistical based anomaly detection) simultaneously. Liu *et al.* [140] propose a signature-based IDS based on Artificial Immune System mechanisms. Antigens, self and non-self elements were simulated as the signature of IoT datagram, normal datagram, and datagram with attacks, respectively. To detect the attacks, the immune cell is simulated as a detector, which can evolve by adapting itself to new conditions. However, this work does not examine how the approach would be deployed and attack signatures stored on low capacity IoT devices. The authors in [141] detect selective forwarding attacks by using a three layered hybrid IDS. In the first layer, a pool of MAC IDs is created by identifying malicious activity using certain message fields and assigning zero to such fields. This malicious pool is then consecutively passed on to the rule and anomaly detection layers, for accurate detection. However, the authors do not mention the exact techniques they employ for pattern matching and anomaly detection in the second and third layers, respectively. Raza *et al.* [46] implement SVELTE and integrate it with a mini-firewall system designed for IP-connected IoT devices. SVELTE is based on detection and correction of inconsistencies by a the 6Mapper module, implemented on the 6LoWPAN Border Router (6BR). In IoT, the 6BR connects the WSN with the IP world and can play the role of a router and a bridge. In SVELTE, the 6BR is responsible for the reconstruction of the RPL DODAG with each node's parent and neighbor information. Equally of notice is the distributed firewall in this proposal, which is embedded in SVELTE with modules at the 6BR and constrained nodes, to prevent network from global attackers. The work in [142] present a framework to evaluate rule, anomaly and neural network-based IDSs, and also discuss how they can be applied to CoAP-based IoT communication environments. However, the results shown by the authors are not well established. In [143], the authors modify SVELTE [46] with same time rank collection by DODAG root, and by attaching timestamps to each DIO message that is either sent or received.

To balance accuracy and resource consumption of the signature and anomaly detection techniques employed, Sedjelmaci *et al.* [146] propose to use Nash equilibrium, a game theory concept involving the interaction of different participants. In this proposal, this concept is implemented in the IDS agent, where anomaly detection is only enabled when new attack's signature is expected to occur. The training, classification and building of a new rule related to new attack's signature is only activated after Nash equilibrium predicted the state of launching of a new signature attack. Thus, this scheme consumes low energy on IoT devices. The technique in [147] is an extended and enhanced version of the work in [146]. Knowledge-driven adaptable lightweight IDS, abbreviated as Kalis, was developed by Midi *et al.* [144]. The architecture of Kalis is equipped with both signature and anomaly detection modules, and the authors compare the proposed architecture's performance with Snort [76]. However, the authors did not explicitly state the details of various algorithms deployed in detection modules. Gajewski *et al.* [145] propose an IDS for smart home systems where the local resource monitoring and preliminary log analysis was beholded to the Home Gateway device, whereas the processing of the long term anomaly analysis of the user's behavior is done at the ISP's premises. Shreenivas *et al.* [48] also extend SVELTE [46] and infer that timing inconsistency in receiving ETX values of nodes by the 6Mapper in SVELTE [46] can lead to ETX attacks, since the attacker can modify the ETX value and gain better position in the formation of DODAG. The authors in [148] assert that supervised and unsupervised optimal path forest (OPF) are parameter independent and fast classifiers that support multi-class classification. For this reason, the authors use supervised OPF as the misuse detection algorithm, for detecting external attacks from the Internet against the 6LoWPAN network. Secondly, they used unsupervised OPF as the anomaly detection algorithm, to detect internal attacks in 6LoWPAN. Napiah *et al.* [149] propose CHA-IDS for detecting combination attacks. CHA-IDS uses 6LoWPAN compression header as a feature in six machine learning algorithms, MLP, SVM, J48, Naive Bayes, Logistic regression and RF. These algorithms classify the type of anomaly and create certain rules. These rules are later updated at the 6BR for the detection of routing attacks. COSMOS, developed by Nespoli *et al.* [150] is equipped with an internal analyzer that uses Yara rules, in order to create the description of malware families, based on binary or textual pattern. The machine learning module in this proposal classifies unknown samples, using RF classification algorithm.

Some proposed works of the hybrid category employ specification and anomaly-based detection techniques. Cervantes *et al.* [135] propose the Intrusion detection of SiNkhole attacks on 6LoWPAN for InterneT of ThIngs (INTI) IDS solution, combining the anomaly and specification based method of intrusion detection, in the context of which monitoring the packet exchange between nodes is employed for anomaly-based detection, while reputation and trust extraction uses specification-based. Similarly, the authors in [151] deploy specification and anomaly agents in the router nodes and 6BR, respectively. At the router nodes, deviations from the packet receiving rate and rate of change of preferred parent are measured, and compared with predefined threshold values, for detection of selective forwarding and sinkhole attack, respectively. Thus, the router nodes analyze locally the raw packets, and send their analysis results to the 6BR. The 6BR analyzes the packets sent by router nodes for anomalies using unsupervised OPF algorithm.

Since projection and clustering are independent processes, parallel projection and clustering are performed using the MapReduce approach. Fu *et al.* [152] model and detect intrusion using Input/Output Labelled Transition System (IOLTS), an extended finite automata model that accentuates system's input and output interactions. To enhance accuracy, the final verification and decision on anomaly is taken manually.

2) ON THE BASIS OF PLACEMENT METHODOLOGY

Since IoT applications may depend on heterogeneous networks, supported by devices with varied capabilities, the decision of where to place IDS is a challenging one. A typical IoT network consists of constrained LLN nodes, and one or more resource-rich border routers. As shown in Figure 3a, such border routers are known as 6LoWPAN gateways, or 6BR, in 6LoWPAN-based IoT networks. We proceed our analyses on the existing proposals, in the context of the various placement strategies.

- **Centralized IDS** - A centralized IDS in IoT applications is, in most cases, implemented on the 6BR. As already discussed, such gateways are placed in close proximity of the IoT network while, on the other end, detection modules are placed on the 6BR in [80], [95], [121], [148], [149], [152]. In the absence of a 6BR, specialized routers, switches or other devices can act as IoT gateways, thus enhancing the inter-operability in IoT communications [73], [78], [83], [117]. Centralized IDS can also be implemented on dedicated centralized systems placed inline but beyond gateways and firewalls, as in [114], [115], [120]. Sometimes, to isolate IDS data communication from regular traffic, sensors are connected to a dedicated host using a wired connection [70], [71]. Specialized powerful IoT hardware like the GPU on ODROID-XU4 [153] is used by the authors in [77] for implementing SNORT IDS. In rare cases, the 6BR collaborates with the cloud and fog nodes as in [116], and SDN controller in [90], in order to gain storage for log databases and implementing resource intensive machine learning algorithms.
- **Distributed** - As for distributed IDS, we find two main implementation strategies. Firstly, specialized sensor nodes (known as watchdogs) are used, dedicated only to the task of monitoring and detection, distributed either equally or hierarchical. This approach is implemented in [87], [104], [125], [139], and can also be realized with the help of specialized fog and cloud nodes connected to the IoT network. In [112], [113], the authors deploy a distributed fog network where the edge nodes (known as fog nodes) train the models and host detection systems. In [150] the authors distribute the lightweight modules to the IoT devices and expensive operations in the cloud servers. The second approach is where the sensor nodes detect intrusions alongside their normal operations. This approach involves the process of continuous self and neighbor inspection, and is discussed in [72], [84], [93], [127], [131], [137], [145]–[147]. In [135], the network is composed of an hierarchical structure, where the nodes are classified as free, leader, and member or associated nodes. The role of each node changes over time, depending on the network configuration.
- **Hybrid** - In the hybrid IDS placement methodology, cluster heads act as a local base station, for nodes within the cluster, while detecting malicious activities within the cluster and from other cluster-heads (see Fig. 3c). Likewise, in [103], network nodes compute trust values of their neighbors and share such values with the cluster-head or border router responsible for the aggregation of the reputation values. In [128], the authors also adapt the method of dividing the network in regions. To cover the entire network, a backbone of trusted and resource-rich monitor nodes is created, which sniffs information from neighbors and characterize compromised nodes. In order to save resources, the authors replace monitor nodes with cluster heads in [132]. Instead of monitor nodes overhearing the entire communication, in this case the nodes exchange related information about themselves and its neighbors directly with the cluster head. This method allows the cluster head to cross verify the information about a cluster member shared by its different neighbors. Identically, Amaral *et al.* [130] deploy selected nodes as watchdogs and configure them with different rules, depending a pre-defined role group and its related application. In the system proposed in [134], after grouping of sensor nodes in cluster, maximum probability node became leader nodes which sent announcement message to its adjacent nodes. Apart from leader nodes, a set of observer nodes are also deployed to monitor and identify the packet drop count of adjacent nodes.

  The second method for hybrid placement involves hosting IDS on both the border router and in the network nodes supporting the IoT application. As compared to the first approach, in this case the presence of a central element is mandatory. Typically, resource demanding IDS modules are placed on the border router, while other more light-weighted computations are supported by the network nodes. This definition matches the theoretical model proposed by Liu *et al.* [105], where anomaly detection agents are deployed at several gateways to gather statistical data and share it with a central service. In [86], network nodes are accountable to detect any changes in their neighborhood, and also to send the same neighbor information to a centralized module which is installed in the 6BR. Similarly in [82], [98], [99], [103], the task of intrusion detection is divided among distributed nodes and edge routers. Although the authors in [98], [103] classify their work as distributed, the fact that they use edge or border router in making the final decision, makes it in fact more of an hybrid approach. In [82], the authors program the nodes to forward *Hello packets* on random paths,

in order to avoid dropping by malicious node. In [113], the proposed hybrid architecture is implemented using a Central Security System (CSS), but the aggregation of information such as RSSI is done by Radio Probes. In [108], the detection components are 6BR as a security awareness center (SAC) and security sensors (SS). The SS is viewed as immunity-based sensor which extracts, evaluates and predicts security situations. On the other hand, the SAC receives the results of security situations from SS and execute security responses in a timely fashion. Trust-based IDS (T-IDS), as proposed by Medjek *et al.* [136], is based on a three-tier hierarchical and cooperative architecture, involving monitoring nodes, 6BR and the Backbone Router (BR). The BR [154] acts as a backbone link and form a collection of LLNs as a single IPv6 subnet where 6BRs provide proxy-neighbor discovery services to their respective LLNs. Thus, multiple 6LoWPAN networks are federated by the BR, which manages a list of authorized nodes to access the entire network. In addition, the BR maintains a list of potential malicious nodes for all 6BR sub-networks. To comply with the constraints of typical IoT devices, the authors in [46] propose SVELTE, together with a mini-firewall module. This proposal implements the resource intensive modules in the 6BR, with the corresponding lighter modules supported by the constrained nodes. Similarly, the authors in [48], [143] aver the architecture of SVELTE [46]. For example, the authors in [48], extend SVELTE [46] by introducing a mechanism for geographic hints to locate malicious nodes inside 6LoWPAN networks. The authors in [126] propose two configurations for their IDS architecture, one centralized and other distributed. Arshad *et al.* [100] implement lightweight signature-based intrusion detection module on sensor nodes and anomaly based at the 6BR. The correlation agent implemented in the 6BR correlates intrusions at network and node levels, enabling improved visibility into the events within sensor nodes. A multi-agent model for intrusion detection is proposed in [96], which comprises an hierarchical architecture of host and network agents. Host agents manage test agents on network terminals, which are further managed by network agents placed over a certain latitude in the network.

We also note that some works in the literature in which architecture was not clear are [140], [144], [151]. In such works, though the authors differentiate between sensing and detection modules, but how and where such modules should be deployed for the implementation of the proposal at hand is not clear in their contribution.

3) ON THE BASIS OF INFORMATION SOURCE

Similarly to conventional IDS solutions, IDS proposals for IoT can also be used to either monitor network and host-based environments. A NIDS for IoT monitors traffic within an IoT network, and acts as a robust line of defense against intrusion before it can access the resources of IoT devices. On the other hand, host intrusions can get recorded in the audit trails of operating system or application software, as in traditional HIDS. It is evident that, in order to protect hosts from intrusions, they must be first prevented and detected at the network level. Due to this reason, nearly 82% of the works in literature focus on NIDS, while only 11% lay their attention on HIDS approaches (c.f.6b).

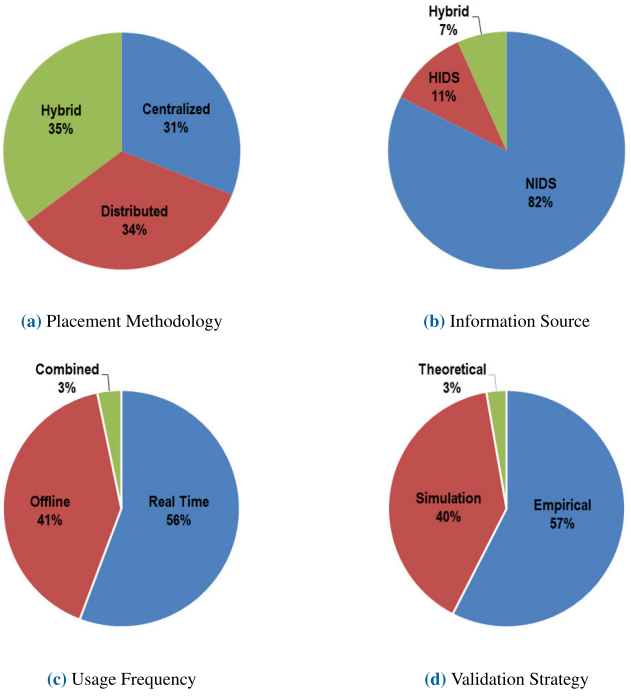
NIDS proposed for IoT are based on detecting protocol anomalies in IoT, in particular focusing on protocols such as CoAP [79], 6LoWPAN [149], RPL and IEEE 802.15.4. Such proposals utilize headers of these protocols to detect any deviation from the normal threshold on general network statistics, such as protocol type, packet length, size and arrival rate. After thoroughly analyzing the literature quantitatively for IoT NIDS proposals, we observe that current implementations do not deal with flow-based inspection, which looks at aggregated traffic information of related packets in the form of flow. Such implementation may help reduce the amount of already large data to be analyzed by the IDS in an IoT network. HIDS, on the other hand, in IoT are mostly based on the IoT device statistics like CPU and memory usage in [93], application's control flow in [127], packet arrival rate in [94], buffer overflows in [111], sensor data in [120] and finally, clock offsets in [137]. Consequently, hybrid of HIDS and NIDS have been implemented in [100], [102], [125], [132], [145] to monitor both network and host activities.

4) ON THE BASIS OF USAGE FREQUENCY

Though the resource constrained nature of IoT devices limit their ability to host an IDS that monitors it in real time, the security and time critical IoT applications in areas such as health care and industrial automation demand detection to be made in real time. Thus, researchers have made attempts to develop real time or near real time IDS for IoT. CEP-based processing [73], [78], offline training of machine learning model, an inline signature based IDS [150] and deploying fog nodes near the edge of the IoT network [112] are some of the approaches to ensure real time detection. Other offline or batch-based works as summarized in Table 5 gather network traffic on certain predefined time intervals using sniffers. In [74], the authors use tcpreplay, an open source suite for pcap editing and replaying previously captured traffic, to deal with the detection of intrusions in real time for TCP packet sizes less than 150 bytes. However, for real time detection in larger packets of at least 1000 bytes, the CPU utilization increases so much that it starts discarding the packets. Thus, they inspected deeply in offline mode. Similarly, Heimdall [83] perform two validations, namely maximum throughput in offline and real time. In real time validation, Heimdall holds the request when an IoT device communicates with new destination. Rather than immediately servicing the request, Heimdall evaluates and validates the destination firstly and then drops or services the request in real time. In maximum throughput validation, Heimdall [83] relies on audit trails by periodically querying the information

TABLE 5. Classification of IDS proposals based on information source and usage frequency.

		On the basis of Usage Frequency		
		Real Time	Offline	Combined
On the Basis of Information Source	HIDS	[111], [137]	[93], [120]	
	NIDS	[46], [48], [70]–[73], [77]–[79], [86], [89], [95]–[97], [107], [108], [112]–[115], [128], [129], [131], [134], [136], [143], [144], [148], [150]–[152]	[80]–[82], [84], [85], [87], [88], [98], [101], [103], [109], [110], [118], [119], [122], [123], [130], [135], [138], [146], [147]	[74], [83]
	Hybrid	[102]	[125], [132], [145]	



(a) Placement Methodology (b) Information Source

(c) Usage Frequency (d) Validation Strategy

FIGURE 6. Frequency of various IDS proposals using various dimensions in IoT.

aggregation system. A detailed tabular overview of the classification of IDS proposals based on information source and usage frequency for IoT is given in Table 5.

5) ON THE BASIS OF VALIDATION STRATEGY

Regarding the validation strategy, two important and recurrent strategies employed to validate proposals for the IoT are simulation and experimentation. Experimentation involves creating duplicate target behavior of the original system operating in a different environment. In experimentation, authors develop and employ testbeds with specialized IoT hardware and software components. One of the commercial devices that closely resembles an IoT device with moderate resources and is often used by researchers in their experiments is the Raspberry Pi. This small single board computer consists of 1.4 GHz 64-bit quad-core ARM Cortex-A53 processor with 1GB RAM and support for WiFi connectivity. Oh *et al.* [72] integrate Raspberry Pi with Omnivision 5647 sensors to capture ten images of 640 by 480 pixels every second. For a smart home environment, Sforzin *et al.* [74] connect Raspberry Pi to home network router using Ethernet interface. In [110], the authors consider only the Zigbee protocol for experimentation, using Raspberry Pi 3 boards and a laptop with Zigbee transmitter. Controlled by a C&C server, Raspberry Pi has also been used as a part of botnet to launch DoS attacks against selected targets by Bezerra *et al.* [93]. In addition to Mirai and Bashlite botnet used by [117], this work also uses famous botnets like Hajime, Aidra, Tsunami and Dofloo for their experimentation. Similar approaches have been used by [78], [150]. Golomb *et al.* [127] create their own IoT testbed comprising of 48 Raspberry Pis which emulate IoT devices of smart cameras and smart lights. Other commercial IoT devices like Arduino UNO and Android phones have been used by authors in [126], [152]. Works on IDS in IoT also revolve around other development boards like Odroid xu3 development board running Java [144], ARM microprocessors running Raspbian OS [137] and devices like Nest thermostat [83] and baby monitors, smoke detectors [88]. In [70], the IDS and PenTest system is implemented on a Linux host connected to the 6LoWPAN via the IDS probes and pen test probes respectively. The works in [85] focus on collecting and evaluating traffic from two IoT devices, namely networked video camera by Foscam and a weather station WS-1001-WIFI Observer. Likewise, Han *et al.* [81] extract data from real vehicle traffic from a commercial car. IDS deployed on an open source hardware like TelosB motes can function with different IoT operating systems like Contiki in [79] and TinyOS in [130].

Some authors develop unique testbeds in terms of IoT software such as MATLAB in [94], [96], [101], [103], [112], [119], [120], [124], [146], R programming language in [116] and.NET framework in [148]. Google's Tensorflow is used in [114] to visualize network design. The authors in [73], [107], [109], [111] talked about deployment in sensor nodes and servers but what operational settings and devices were actually used was unclear.

The performance evaluation of IDS with network simulators predates the IoT, and in reality many existing simulators

TABLE 6. Overview of potential simulators used in IDS for the IoT.

Simulation Platform	Operating System	Programming Language	Targeted IoT Device Class[a]	Network Stack	Ref.
Cooja	Contiki	C	Class 0 - 1	uIP, RIME	[46], [48], [100], [108], [115], [118], [131], [132], [135], [136], [143], [149]
Qualnet	Linux, Windows, Solaris, and Mac OS X	C++	> Class 2	TCP/IP	[84]
ns-2	Linux	C++ and Otcl/tcl	Class 2	OSI, TCP/IP	[134], [139], [141]
OMNeT++	Linux, Mac OS/X, Windows	C++, NED, INET	> Class 2	TCP/IP	[142]
TOSSIM	TinyOS	nesC	Class 0	6lowpan/RPL IPv6 stack	[146], [147]
SENSE	-	C++	Class 1-2	TCP/IP	[138]
NetSim	Windows	C		6lowpan/RPL IPv6 stack.	

[a]As per [155]: Class 0 devices have <<100 kB ROM and <<10 kB RAM, Class 1 devices have $\sim$ 100 kB ROM and $\sim$ 10 kB RAM, Class 2 devices have $\sim$ 250 kB ROM and $\sim$ 50 kB RAM

for WSN or basic networks have been adapted to integrate IoT-related elements and protocols. The works considered in this survey employ WSN simulators like Cooja [133], Qualnet [156], NS-2 [157], OMNeT++ [158], TOSSIM [159] and SENSE [160], which can be potentially used as part of IoT research. The Cooja simulator [133] is available as part of Contiki OS [161], an open OS for programming the IoT sensors. Cooja [133] runs the uIP and RIME network stack and access the standard IoT protocols implemented by Contiki such as CoAP, 6LoWPAN, RPL and IEEE 802.15.4. The Cooja simulated firmware of Contiki running on virtual nodes can also be deployed to real physical hardware, with few modifications reducing the gap between proof of concept and prototyping phases. In reality, Cooja is not a simulator rather an emulator, thus producing realistic IoT scenarios. In order to reproduce such realistic scenarios, authors in [46], [48], [82], [86], [87], [108], [132], [135], [136], [143] run experiments on emulated Tmote sky nodes within the Cooja Simulator, running Contiki code. Cooja emulates real IoT traffic with additional features like PowerTrace [162], which helps to analyze the power consumption of a sensor node. With native support for 6LoWPAN over IEEE 802.15.4 networks, NS-2 simulator [157] is used to evaluate the IDS by authors in [134], [139], [141]. However, NS2 lacks support for the application layers protocols of IoT, which currently limits its applicability to validate proposals dealing with protocols such as CoAP, RPL or 6LoWPAN. Another well established and extensible simulator is OMNeT++ [158], used by [142] to deploy and evaluate the proposed IDS framework. OMNeT++ also lacks support of inbuilt IoT models and protocols, and the consolidation of missing IoT components remains a manual process. A commercial simulator, Qualnet [156], has additional sensor network library supporting IEEE 802.15.4 standard to achieve IoT specific simulations [84]. The TOSSIM simulator [159], a simulator for the TinyOS operating system [163], is used by the works in [146], [147] to evaluate the state for activating anomaly detection based on Nash equilibrium concept. Another simulator, SENSE ((SEnsor Network Simulator and Emulator) [160] with the design goals of reusability, extensibility and scalability has been used by Amin *et al.* [138] by working only with the IEEE 802.11 link layer component. A detailed tabular overview of the simulators used in IDS for IoT is given in Table 6.

In [71] the authors validate their proposal using an hypothetical example by developing an application integrated with the Frequency Agility Manager (FAM) while accounting for flooding attacks. The overall architecture also includes Scapy for penetration testing, Prelude for incident and event management and Suricata for intrusion detection. Theoretical analysis by creating a system model, lemma and proofs are done in [100], [101], [105].

6) ON THE BASIS OF SECURITY THREAT/ATTACKS

In this section we consider IDS proposals with respect to attacks and their categories, as expressed in the taxonomy previously discussed and illustrated in **Figure 2**. From our thorough survey of the literature, we find that the most common attacks in IoT are those against routing operations. In fact, out of the 68 works analysed in this survey, 23 [46], [48], [82], [86], [87], [98], [101], [103], [108], [115], [128], [131], [132], [134]–[136], [141]–[144], [151], [164] focus its research on routing attacks.

- Routing Attacks - The authors in [86] detect wormhole for two cases; one with packet encapsulation and the other with packet relay. This work hypothesizes that after wormhole attacks new neighbors are formed at the other end of the wormhole tunnel. As wormhole attack changes the hops in routing with tunneling, therefore the work in [151] add a new field, the rank of packet generating source node, and use it to compare the real and logical hop counts. The researcher in [115] uses multiple meta-features from the IPv6 header to detect wormhole, but the details of the exact mechanism is missing. Shukla [121] detects wormhole in two levels at the 6BR: level one detection is based on the request from a node to another node not belonging to the same cluster. In level two, the 6BR reports wormhole attack if distance between two nodes is greater than the safe permissible distance. Most of the works either gather probabilities from other nodes [82] or keep track of number of unanswered request or packets from child nodes [46], [151] to detect selective forwarding attackers. Likewise, Thanigaivelan *et al.* [99] execute a sub-routine at the gateway to check whether it receives contextual data from every active node in the network. This list of active nodes is prepared by the reception of DPO messages in RPL. Though no implicit defense for clone ID attack has been proposed in the literature, Raza *et al.* [46] propose to use location information at the 6BR to mitigate this attack. Other distributed mechanisms like distributed hash tables may also be used [23]. Thanigaivelan *et al.* [99] verify the node relationship by validating DPO messages reported by the neighbor nodes to localize the cloned node.
  Detection of Sybil attacks has mostly been based on trust calculation strategies [136] and consistency check in the difference in fair play points [101]. Nonetheless, the Sybil and Clone ID attacks on IoT routing protocol (RPL) have not been evaluated much. Digital signatures and MAC verification of the version number was done by VeRa scheme in [165]. In [103], the IDS nodes detect version attacks by substantiating new version number messages that they receive from their child and the root. Specialized monitoring nodes are deployed in IoT network which report increased version numbers in their neighborhood to the root [87]. The detection of sinkhole attacks focus on routing graph validation [46], trust experience in ranks [103] and change in preferred parent [151]. Detection of invalid topology state [128], same time rank [143] and increase in rank values [48] are the major approaches available in the literature to deal with this class of attacks. Authors in [128] deploy monitor nodes to record frequency of local repairs initiated by nodes. In the proposed model, if the number exceeds a threshold, monitoring system raised an alarm. The proposal in [132] consists in penalizing nodes which did not have latest DIO message. Authors in [48] address detection of ETX attacks by continuous verification where the ETX of parent must always be less than ETX of child.
- DoS Attack - In the case of high rate or flooding based DoS attack, the proposals usually define a threshold on the different network parameters. These techniques usually set an acceptable threshold on the number of packets per second [70], [71], [94], [129], on the total length of data packets [73] or on the energy consumption of the device [84]. Other methods include training, using packet traces [95], [109], [111] and measuring deviations from normal traffic [78], [79], [124], [125], [137], [139].
- MiTM Attack - Botnet attack in IoT has been detected in [80], [83], [93], [117], and other man-in-the-middle attacks, such as byte exchanges and bit flips, are dealt with in [142]. Meidan *et al.* [117] implement computationally expensive deep autoencoders to learn from benign data and use them for instantaneous detection of unknown and existing botnets. Bezerra *et al.* [93] claim that botnet attack causes unusual changes in the resources of IoT devices, like electric potential difference, number of simultaneous tasks, CPU and memory consumption which enables them to detect botnet malware.
- Spoofing Attack - Spoofing attack is detected in [46], [48], [108], [126], and is a multilayer attack in which attacker impersonates an authorized and legitimate node, either to make use of the resources or to disrupt the functioning of the network. Detection is based on authentication measures [126] and listing of nodes as either black or white [46].
- Penetration Attack - Out of bound read and write access, overflows and underflows in stacks or heaps [111], wrong password attempts and access attacks [81] have been detected in the literature. A handful of works in the literature [96], [112], [113], [116], [119], [122] detect access and DoS attacks from the KDD dataset [166], namely, Probe, DoS, U2R and R2L. Conventional attacks such as SQL injection, code injection and code reuse are pivoted in [85], [114], [123], [127] and malicious human intent in [164]. Besides, adversarial attacks detected in [88] aim at designing an input in a particular way so that model gives wrong result.
  Based on evaluation of event dependencies, a pioneer work in [100] explores the effect of multistage attacks in IoT infrastructure independent of any routing protocol. These attacks exploit multiple technologies deployed in multiple stages to penetrate an IoT network. Additionally, combined attacks of hello flood, sinkhole and wormhole are detected in [149]. The authors in [79] put limits on number of valid and invalid requests to detect any malformed CoAP messages. In [150], the authors propose COSMOS, an IoT sentinel, to detect malware attacks on IoT devices. In addition to detecting routing attacks, Chawla [115] detect opportunistic service attack where malicious node provides good services

to its neighbors initially to gain better reputation among peer nodes.

7) ON THE BASIS OF EVALUATION METRICS OF IDS

The evaluation of a typical IDS in IoT involves measuring detection performance with metrics like accuracy, TPR, FPR, precision etc. However, since IoT devices are constrained with respect to resources for operations of intrusion detection, the analysis of performance overhead caused by IDS is an important criteria of evaluation.

- Detection Related - In the literature, the metric of accuracy has been used to evaluate the performance of classification [88], [101], [144], pattern matching [72], [85], random forest [89], neural network based deep learning [109], [111], [113]–[115], clustering [120], [151] and game theory [146]. Although accuracy is a great start metric to measure performance of an IDS, it is not considered enough to measure performance. It is due to a phenomenon known as the accuracy paradox. Accuracy paradox refers to a situation when one IDS with given accuracy may have more predictive power than the other with higher accuracy [167]. Thus, comparing different IDS only on the basis of accuracy is avoided, in favor of other less misleading measures metrics as precision, recall, specificity and fallout. Precision and recall have been judiciously used to evaluate the IDS in [70], [80], [84]–[86], [107], [113]–[115], [120], [137]. The evaluation using metrics of accuracy, TPR, TNR, FPR, FNR and precision of anomaly based IDS using confusion matrix was done in [109], [111], [114], [120], [132]. Precision does not take into consideration the left out real intrusions (FN), while recall leaves out false positives in its calculation. Thus, an IDS can have good precision but it may not express the percent of correctly detected intrusions, versus all the intrusions that exist. Besides, an IDS may have a good recall, while high false positives at the same time. Hence, precision and recall do not completely define accuracy of an IDS. Also, they are not discussed in isolation, but either value of one metric is compared for a fixed value of other metric or both are combined into a single metric like F-score, a combination of precision and recall was calculated by the works in [113]–[115]. Finally, evaluation of IDS for IoT using ROC Curve and AUC was only done by authors in [139], where performance of CUSUM charts was presented with increase in number of attackers and traffic load. Apart from the above mentioned standard metrics, some other non-conventional evaluation metrics were used by the authors to evaluate their respective IDS. For example, the authors in [82], [86], [103] calculated the amount of packet overhead generated by the IDS, in order to detect the attacks. Another important metric that has been discussed by very few works is the detection delay or time until detection. Often, IDS employ accurate algorithms for detection but information about the time needed to detect a given intrusion is rarely presented. With this time, IDS are classified as being real time or non-real time. This time is the difference between the attack launch time and the attack detection time, and was discussed in [89], [113], [138]. In other works [72], [73], [83], [111], this time was indirectly termed as the execution time or the time taken by the IDS to generate appropriate output. It is clear through this analysis that almost all the works only evaluate IDS using basic detection metrics and lack the deeper analysis and evaluation of proposed intrusion detection algorithms. The advanced metrics of F-score, ROC and AUC not only enhance the comparison of available work with other works, but also quantify detection uncertainties. Hence, all these metrics must be considered intuitive and complemented with basic detection metrics when comparing different IDS.
- Resource Utilization - The memory utilized by the proposed IDS has been calculated in [73], [74], [101], [130]. The authors in [86] computed and compared energy statistics with that of a primitive application on the node. To run the proposed application, power consumption in [111] was measured as 10.45% more than the baseline power consumption. Network energy overhead and node power consumption using Contiki's Powertrace application were computed in [46], [48], [131], [132], [134]. This application depicts the time the various parts of a node were on and using this time, the energy consumption for various events is calculated. Similar method for calculating energy consumption were used to evaluate IDS in [147]. An average of 0.19% CPU was obtained by Kalis in [144], as compared with 6.3% for Snort [76]. Finally, in [139], [141], [146] the difference of initial energy set in sensor nodes and energy consumed was used to calculate average energy consumed at all nodes. The works [71], [81]–[85], [87]–[89], [98], [102]–[110], [113]–[115], [119], [120], [124], [126], [128]–[130], [135]–[138], [140], [142], [143], [145], [151], [152], [164], [168], [169] do not provide any performance metric for their proposed IDS solutions.

The classification of various proposals with evaluation metrics for routing attack detection is shown in Table 7.

8) ON THE BASIS OF DATASETS USED FOR EVALUATION

This section reviews existing datasets used in the state of the art proposals for designing IDS in IoT. Whilst Deng *et al.* [96] use all the available features from the KDDCup dataset with no feature selection technique, Putchala [114] extract six features out of the total 41 features using random forest classifiers. The author used 10% KDDCup dataset and split it into different layers based on attack types like buffer overflow, teardrop, smurf, IP and port sweep etc. Even though KDDCup has been used by proposals for the design and evaluation of IDS in IoT, the dataset and attacks is not correlated with IoT network traffic. The authors in [122] also employ KDDCup and NSL-KDD for evaluating their IDS. The authors considered all features from KDDCup and

TABLE 7. Classification of proposals for Routing attacks with their evaluation metrics.

Routing Attacks								
Wormhole	Selective Forwarding	Sybil	Sinkhole	Version Number	Blackhole	Rank	Neighbor	ETX
• **[23]**-94% TPR, 328mJ, 2.8KB RAM, 24.9KB ROM	• **[82]**-100% TPR	• **[101]**-99.2% Accuracy	• **[115]**-90% TPR, Precision (0.95), F1 Score (0.96)	• **[87]**-20% FPR	• **[115]**-0.9(Precision), 91% (TPR), 0.9 (F1-Score)	• **[118]**-0.95 (Precision), 96% TPR, 0.94 (F1 Score)	• **[132]**-100% TPR, 2.64% FPR	• **[48]**-90% TPR, 5570B RAM, 6B ROM
• **[115]**-95% TPR, Precision (0.91), F1 Score (0.9)	• **[99]**-3.3kB ROM, 864B RAM	• **[136]**-2.2 mW Energy Overhead	• **[132]**-100% TPR, 3.28% FPR	• **[103]**-5% FPR, 7% FNR				
• **[121]**-71-75% TPR	• **[46]**-80% TPR	• **[146], [147]**-92% TPR, 3% FPR, 8500 mJ	• **[46]**-90% TPR	• **[118]**-0.94 (Precision), 0.94 (TPR), 0.95 (F1 Score), 94.7 (AUC)		• **[132]**-100% TPR, 5.25% FPR		
	• **[143]**-6% FPR		• **[135]**-92% TPR, 28% FNR, 5% FPR			• **[143]**-5% FPR		
• **[151]**-96.02% TPR, 2.08% FPR, 100% Accuracy	• **[141]**-98% Accuracy, 98.4% TPR		• **[151]**-100% TPR, 2.98% FPR, 69.23% Accuracy			• **[48]**-0.05mW CPU Power		
	• **[144]**- 91% TPR, 100% Accuracy, CPU 0.19%, 13978kb Memory Overhead • **[148]**-95.48% TPR, 4.35% FPR • **[151]**-85.36% TPR, 10.12% FPR, 91.43% Accuracy							

19 important features from the NSL-KDD dataset. However, details of feature selection technique in this work is missing. The other work, conducted by AL-Hawawreh *et al.* [116], extract most significant six features using deep learning based neural network from NSL-KDD dataset. Likewise, in [148] the authors select 29 features using variant of binary gravitational search algorithm, and Pajouh *et al.* [119] select 35 features using PCA and LDA feature selection algorithms. Though advantageous over KDDCup, NSL-KDD lacks low foot print attacks and does not match the IoT network traffic and attacks. Hosseinpour *et al.* [107] test their IDS on SSH Brute Force attack from ISCXIDS dataset. To overcome the drawback of limit on number of user attempt in SSH protocol, the authors modify the dataset by incorporating SSH Brute Force attacks from number of bots thereby creating a distributed model. The other source which evaluates IDS based on capture files obtained from ISCXIDS is [77]. Though ISCXIDS is obtained from realistic network, the traffic is still not characteristic of IoT networks. A recent work by the authors in [123] deploy recursive feature elimination to select 14 features from CICIDS dataset. A similar approach is also used to extract important features from UNSW-15 dataset. The UNSW-15 dataset [59] reflects real normal behaviour and synthetic attack activities along with a large number of protocol and services and thus, the authors in [116], [123] used it to perform evaluation of IDS in IoT networks. The relevance and applicability of these aforementioned datasets with respect to IoT is listed in Table 8.

Since the relevance of aforementioned traditional datasets in IoT is minimalist, three new datasets- Sivanathan *et al.* [60], RPL-NIDDS17 [61] and Bezerra *et al.* [93] are specific to IoT and include particular protocols and attacks of IoT. The dataset by Sivanathan *et al.* [60] can be used for IDS evaluation of smart cities and campuses. Bezerra *et al.* [93] provides dataset on IoT devices infected with botnet malware whereas RPL-NIDDS17 [61] is strictly focused on routing attacks on RPL in 6LoWPAN networks. None of these datasets have been used so far for evaluating IDS in IoT. Table 9, compares these datasets according to the principal objectives for qualifying datasets.

9) ON THE BASIS OF DATA PRE-PROCESSING AND FEATURE SELECTION TECHNIQUES

As it can be inferred from the literature, there exists numerous proposals of IDS in IoT. Throughout the literature, the authors either optimize existing IDS solutions or propose new detection algorithms. Only few proposals focus their attention on data pre-processing and dimensionality reduction techniques which are proven to improve the accuracy of any IDS. Such techniques typically remove redundant and extract significant features from the data backed up with discretization and normalization. The initial reference of filtering events to remove redundant data and transform them into unified format before feeding them to intrusion analysis component is presented in [73]. In [81], the authors employ correlation feature analysis to combine several correlated features into one feature. With this approach, the authors reduced 41 features to 4 most significant features of vehicle data. Another simplistic approach by [85] uses bit pattern matching with binary AND operation and a conditional counter to select features from the data.

TABLE 8. Comparison of datasets used in the literature for evaluating IDS in IoT.

Datasets	Ref	Applicable to IoT
KDDCup	[96], [114], [122]	×
NSL-KDD	[112], [113], [116], [119], [122], [148]	×
ISCXIDS	[77], [107]	×
CICIDS	[123]	×
UNSW-NB15	[116], [123]	✓
[60]	-	✓
RPL-NIDDS17 [61]	-	✓
[62]	-	✓

Data pre-processing also involves feature discretization and normalization to convert continous features to discrete ones and also standardize and normalize the date within a particular range. As a result, the calculations are speed up in the machine learning algorithms that follow. Putchala [114] discretize three categorical features of KDD dataset, protocol_type, service and flag into numerical values before inputting them to the model. Besides, feature normalization using Min, Max and Median of IoT data is discussed in [93], [94]. Bezerra *et al.* [93] gathers host data comprising of CPU and memory usage of the devices, number of concurrently running tasks and electric potential. This gathered data is fed into a remote server where the data instances are grouped and normalized according to Min-Max values within the range of 0 and 1. Jan *et al.* [94] generate four features from a single attribute, packet arrival rate, with multiple combinations of mean, max, median values of the attribute.

In the context of machine learning, the conventional feature selection methods are filter and wrapper as discussed in Section III. For feature selection in IoT, filter method is employed in [151] to select four features from the raw packet, each having a characteristic traffic related information. By reconstructing session from pcap files, Meidan *et al.* [88] extract session level features from network, transport and application layer data. The authors even added 60 new features of their own for the purpose of whitelisting the traffic. Though filter methods have low execution time, they have lesser accuracy than wrapper methods which use ML algorithms to evaluate the accuracy of selected features. Thus, unsupervised clustering method is used in [107] to divide the traffic into normal and intrusion packets. The classified packets are then provided to AIS based IDS for

TABLE 9. Comparison of IoT applicable datasets.

Datasets	Realistic Network Configuration	Realistic Traffic	Labeled Dataset	Total Interaction Capture	Complete Capture	Diverse/ multiple attack scenarios	Format	No. of Features	No. of Instances
(+) UNSW-NB15	True	True	True	True	True	True	CSV	49	2 Million and 540044
(-) Sivanathan	True	True	True	True	True	-	CSV	12	-
(-) RPL-NIDDS17	True	True	True	True	True	True	CSV	20	465318
(+) Bezerra	True	$True^1$	False	True	True	$False^2$	CSV	4	-

(+) Indicates that the Dataset is publicly available. (-) Indicates that the Dataset is not publicly available. 1. Host based dataset. 2. Only DDoS attack

online training. In a similar manner, Putchala [114] employ random forest classifier to select important features from the KDDCup dataset. Another wrapper based technique by authors in [148] perform feature selection using MI-BGSA algorithm. The authors evaluate the performance of this method using a two-objective fitness function based on false alarm rate and detection rate. Yavuz *et al.* [118] select, normalize and extract features from Wireshark pcap files and generate their own IoT Routing Attack Dataset (IRAD) with 64 million entries. The authors implement automatic feature extraction algorithm in Python using Pandas and Numpy libraries. Recently, the authors in [123] employ Recursive Feature Elimination (RFE) to analyze the significance of each feature through feature importance and obtain an optimal set for three metrics-accuracy, recall and precision.

Likewise, subset features are not only selected but new features are also extracted from existing ones using feature extraction. Feature extraction further reduces the computational complexity of the classifier using techniques like PCA, LDA, SFC etc. In IoT, unsupervised PCA is used by authors in [95]–[97], [119], [125] for reducing the dimensionality of features. With main characteristics of data retained, PCA is used to reduce IoT data to two dimensions in [125]. Granjal *et al.* [95] compare PCA and LDA for multi-class classification using various kernels of the SVM classifier. The authors deduce that LDA is better than PCA in terms of accuracy and clustering the data according to each of the classes. As a result, the authors employ only LDA with SVM classifier for binary classification of intrusions. Pajouh *et al.* [119] perform a two-layer dimension reduction by implementing unsupervised PCA succeeded by supervised LDA. The authors claim that the output of PCA transformed dataset is not ideal for classification, thus LDA's linear computation is used to further speed up the intrusion detection. As a result of dimension reduction the authors reduced the complexity to half by using only four of 35 features from the NSL-KDD dataset. Finally, Liu *et al.* [97] aim to improve effectiveness of IDS by using PCA followed by SFC. In this work, the PCA algorithm reduce the number of variable and eliminate correlated features. The resultant dimension reduced data is divided into high-risk and low-risk data with SFC algorithm.

B. STATE OF THE ART IN IoT-IPS

Apart from IDS, other defensive approaches of securing network in IoT are Intrusion Prevention Systems (IPS). The aim of IPS is to ensure guaranteed access to resources, trace the attacker and minimize the intensity of the attack. The early approaches to IPS include blocking malicious and spoofed IP addresses, rate limiting certain protocols and ingress/egress filtering. All these approaches can be implemented either on a single host (as in HIPS) or the network (as in NIPS).

Regarding HIPS proposals for IoT, the prevention of UDP flooding attack in the Contiki operating system of IoT has been proposed by Kamaldeep *et al.* [170]. The authors implement an ICMP rate limiting mechanism on the Tmote sensor nodes which effectively reduces the consumption of bandwidth on the reverse path of already low power and lossy networks of the IoT. Authors in [111] propose another HIPS on smart controller IoT system to stop receiving data from the sensor nodes in wake of any intrusion. The evaluation of proposed HIPS is performed with Runtime Intrusion Prevention Evaluator (RIPE) [171]. Considering that sampling all the packets is both exhausting and time consuming, Misra *et al.* [129] propose DALERT, an HIPS employing the concepts of Learning Automation (LA) and establishing an optimum sampling rate at sensor nodes to prevent DDoS attacks in IoT. The mitigation of malware using Yara rules is discussed in [150]. Similarly, Nobakht *et al.* [90] uses OpenFlow rules to quarantine or limit accesses to the infected IoT device. High speed digital signal processing hardware chip-based IPS using statistical and high order spectrum focusing characteristics of Sybil intrusion is presented in [96].

A preliminary reference on NIPS in IoT is discussed in [164], where the authors propose a hash-based mechanism to generate entity behavioral proofs of devices, consisting of their normal behaviour. If an anomalous behavior from the normal pattern is measured, the IPS blocks those users. However, it is interesting to note that the mechanism is proposed

TABLE 10. Classification of IRS according to level of automation and activity of triggered response.

IRS		Activity of Triggered Response	
		Passive	Active
Level of Automation	Notification	[109]	-
	Manual	-	-
	Automated	RIDES [139],KALIS [144], [152]	IMLADS [125], [169]

altough not evaluated for preventing any attack. In [172], the authors introduced the REATO NIPS system, which is based on the usage of a cross domain middleware, identified as Networked Smart Objects (NOS), which aims to avoid DoS attacks taking place in short time intervals. The NOS keeps track of connection requests and sends packets to data sources (IoT sensor nodes) in the network. The authors of [77] suggested faster pattern matching in SNORT to actively block packets or source of transmission. The prevention of DoS attacks using predefined threshold on a number of CoAP requests from a particular source over a particular time frame to drop further communication from that source is presented in [79], [95]. The IPS proposed in [83] is an hybrid IPS, where IoT devices prevent botnet attacks by maintaining their own statistical profile of traffic patterns, such as the number of packets per minute categorized by packet type such as UDP, TCP, Address Resoultion Protocol (ARP) and Domain Name Service (DNS). On the other hand, the centralized traffic manager prevents devices from communicating with illegitimate destinations, by white listing both incoming and outgoing traffic.

C. STATE OF THE ART IN IoT-IRS

In this section, we will discuss the existing proposals of Intrusion Response Systems (IRS) for the IoT, analyzed along the proposed taxonomy. The initial reference to response system in IoT is discussed in [139], where the authors introduce the idea of a gateway-initiated automated attack response, with the goal of subverting the attack and to traceback the attack source for future packet filtering. In spite of being an important security domain, the authors only mentioned the idea and did not explore it further. To avoid network disruptions, a notification IRS for IoT that warns the response team at an early stage of attack is proposed in [109]. A similar notification-based IRS that produces reports and sends them to the network administrator is proposed in [152]. The response system built by [144] is an example of an association-based automated IRS. To simulate a response action, in this work the authors revoke the attacking node from the network. All the previously discussed IRS are passive, and as such do not attempt to minimize the damage in the IoT application and devices. However, Greensmith [169] propose a responsive version of a deterministic Dendritic Cell Algorithm (DCA), an AIS technique which incorporates T-cell effectors for automated responses to detection of an intrusion. The response involves disconnecting the connection, throttling the packet rate or delaying the connection. Recently, an active and automated adaptive IRS built by the authors in [125] aims to reduce the anomaly influence by classifying different anomalies and its respective response strategies. Likewise, system maintenance agents perform patching and traffic filtering, anti-virus transfers the infected node and quarantine management agents control anomaly behavior with access control. Table 10 highlights the IRS proposals in IoT on the basis of level of automation and activity of triggered response.

V. ANALYSIS AND DISCUSSION

In this section, we compare and analyze the previously identified and discussed proposals, and for the purpose of comparison we employ the particular taxonomy of IDS, IPS and IRS proposed in Section III. To envisage the proposals reviewed in this survey, Table 11 enlists all the research contributions towards developing IDS for IoT. In the context of this analysis, we must note that all the proposals apply to one or more IoT protocol from the standardized communication stack and other open source, commercial and proprietary protocols are kept out of the scope of this review.

We note that research in IoT IDS has seen tremendous growth, especially in what respect the application of machine learning. In this regard, researchers either optimize algorithms for implementation in IoT systems, or on the other hand deploy powerful nodes to perform intrusion detection in IoT. With respect to the challenges posed by the huge amount of IoT data, researchers also divert towards deep learning algorithms for intrusion detection. However, both ML and DL encounter challenges in obtaining realistic and high-quality datasets consisting of the various types of attacks. We observe that such comprehensive and diverse datasets are in fact rare. We also note that IoT systems continuously stream highly heterogeneous data, leading to the possibility that data may be noisy and corrupted. In such a case, effective ML and DL models should be proposed that can learn from low-quality data. As for information source and usage frequency, we observe that most proposals focus on real time and offline NIDS, with few exceptions of HIDS and hybrid audit data. On the other hand, we can see that the placement strategy is balanced, with equally distibuted empirical and simulation validations. In general, proposals focus more on routing attacks with a few addressing pre-processing of raw data.

Regarding the prevention and response to intrusions, we observe that further work in this domain is required, in order to prevent and ascertain the impact of intrusions with

TABLE 11. Summary of proposed IDS for IOT. Reviewed works are grouped based on the proposed taxonomy in Fig.2.

Detection Algorithm		Placement	Information Source	Usage Frequency	Validation Strategy	Security Attack	Evaluation Metric	Dataset	Data Preprocessing & Feature Selection	Year	Ref
Signature based											
		Centralized	NIDS	Real Time	Empirical (Metaspoilt, Scapy)	DoS (UDP Flooding)	Detection Metrics- TPR & FPR	Self Generated Dataset	-	2013	[70], [71]
		Distributed	NIDS	Real Time	Empirical (Raspberry Pi)	-	Detection Metric-Accuracy	Packet Capture from Internet	-	2014	[72]
		Centralized	NIDS	Real Time	Empirical	DoS (Land Attack)	Resource Utilization - CPU and Memory	-	Event Filtering	2014	[73]
		Distributed	NIDS	Combined	Empirical	-	Resource Utilization - CPU and Memory	Public PCAP from NETRESEC	-	2016	[74]
		Centralized	NIDS	Real Time	Empirical	-	Power,Exec Time	ISCXIDS 2012, Appneta	-	2018	[77]
	Threshold	Hybrid	NIDS	Real Time	Empirical	DDoS - Internal and External	Energy, Memory	Ad Hoc	-	2018	[79]
	CEP	Centralized	NIDS	Real Time	Empirical	DDoS	TPR, FPR, Accuracy	Ad Hoc		2018	[78]
Anomaly based											
Statistical		Centralized	NIDS	Offline	Simulation	MiTM (Botnet on 6LoWPAN)	Detection Metrics-TRP & FPR	Self Generated Dataset	-	2009	[80]
		-	NIDS	Offline	Empirical	Access attack	-	Dataset from Commercial Car	Factor Analysis	2015	[81]
		Hybrid	NIDS	Offline	Simulation (Cooja)	Routing attack (Selective Forwarding)	Detection Metric -TPR	Self Generated Dataset	-	2017	[82]
		Centralized	NIDS	Combined	Empirical	DoS and Botnet	Execution Time	Self Generated Dataset	-	2017	[83]
		Distributed	NIDS	Offline	Simulation (Qualnet)	DoS	Detection Metric-TPR	Self Generated Dataset	-	2014	[84]
		Distributed	NIDS	Offline	Empirical	Tunneling, Worm propagation, SQL code injection	Detection Metric-Accuracy	Generic Web	Bit Pattern Matching (LiSP)	2015	[85]
		Hybrid	NIDS	Real Time	Simulation (Cooja)	Routing Attack (Wormhole)	FPR, Memory, Energy	Simulated	-	2015	[86]
		Distributed	NIDS	Offline	Simulation (Cooja)	Routing Attack (Version Number)	Detection Metric-FPR	Self Generated Dataset	-	2017	[87]
Anomaly Based - Machine Learning	**Random Forest**	Centralized	NIDS	Offline	Empirical	Spoofing Attack	Detection Metric-Accuracy	Self Generated Dataset	Filter Method	2017	[88]
		Hybrid	NIDS	Real Time	Empirical	-	Detection Metric-Accuracy	Ad Hoc	-	2017	[89]
	SVM	SDN-Centralized	HIDS	-	Empirical		Precision, Recall	Ad Hoc	Wrapper-Heuristic	2016	[90]
		Distributed	HIDS	Offline	Empirical	Botnet	Accuracy, Precision, Recall, AUC, F1, Specificity	Self Generated Dataset	Min-Max Scaling	2018	[93]
		Centralized	NIDS	Real Time	Empirical	DoS	Accuracy, RoC, Recall, F-Measure	Ad Hoc	PCA, LDA	2018	[95]
		Centralized	HIDS	-	Simulation (MATLAB)	DDoS	Accuracy, TPR, FPR, TDR, FDR, CPU Time	Ad Hoc	Mean, Max,Median	2019	[94]
	FCM	Hybrid	NIDS	Real Time	Simulation (MATLAB)	-	DR,FPR	KDDCUP99	PCA	2018	[96]
	Fuzzy Clustering	-	NIDS	Real Time	Empirical	-	Accuracy, FPR	-	SFC, PCA	2018	[97]
	Correlation	Distributed-Hierarchical	NIDS	Offline	-	Routing and DoS attack	-	-	-	2016	[98]
		Hybrid	NIDS	Offline	Empirical and Theoretical	Routing attack (Sybil)	Accuracy, Execution Time, Memory Overhead	Synthetic and Real-time Email Dataset		2017	[101]
		Hybrid	Hybrid	-	Simulation (Cooja)	Multistage attacks	Power, Memory Overhead	Simulated	-	2018	[100]
		Hybrid	NIDS	-	Empirical and Simulation (Cooja)	Routing Attacks (Selective Forwarding, Clone, Packet Flooding)	Memory, Power	Simulated	-	2018	[99]
	Computational Intelligence	Distributed	Hybrid	Real Time	-	-	-	-		2013	[102]
		Hybrid	NIDS	Offline	Simulation (MATLAB)	Routing attack (Selective Forwarding, Sinkhole, Version Number)	TPR, FPR, TNR, FNR	-	-	2017	[103]
	Artificial Immune System	Distributed	NIDS	-	-	-	-	-	-	2012	[104]
		Hybrid	NIDS	-	Theoretical	-	-	-	-	2012	[105], [106]

TABLE 11. *(Continued.)* **Summary of proposed IDS for IOT. Reviewed works are grouped based on the proposed taxonomy in Fig.2.**

Detection Algorithm		Placement	Information Source	Usage Frequency	Validation Strategy	Security Attack	Evaluation Metric	Dataset	Data Preprocessing & Feature Selection	Year	Ref
Deep Learning		Distributed	NIDS	Real Time	Empirical	MiTM (Botnet)	Accuracy, TPR, FPR, TNR, Precision	KDD-Cup 99, ISCX	Wrapper Method	2016	[107]
		Hybrid	NIDS	Real Time	Simulation (Cooja)	Spoofing and Routing Attack(Sinkhole and Sybil)	-	Ad Hoc	-	2017	[108]
	Artificial Neural Network	Centralized	NIDS	Offline	Empirical	DDoS	Accuracy, TPR, FPR, TNR, FNR	Ad Hoc		2016	[109]
		Centralized	HIDS	Real Time	Empirical	Data integrity and DoS	Accuracy, TPR, FPR, TNR, FNR, Execution Time,Energy Overhead	BugBench [173]	-	2016	[111]
		Hybrid	NIDS	Offline	Empirical	DoS(Jamming)	-	Ad Hoc	-	2017	[110]
	Deep Learning	Centralized	NIDS	Real Time	Simulation (Cooja)	Routing (Sinkhole, Blackhole, Wormhole) and QoS attacks	Accuracy, TPR, FPR, FNR, Precision, AUC and F-score	Scapy Penetration Testing	-	2017	[115]
		Distributed and Parallel	NIDS	Real Time	Empirical	Penetration Attack(U2R, R2L),DoS	Accuracy, TPR, FPR, Precision, AUC and F-score	NSL-KDD	Deep Unsupervised Learning-Sparse Auto Encoder	2017	[113]
		Centralized	NIDS	Real Time	Empirical	Multilayer	Accuracy, TPR,FPR, TNR,FNR, Precision, AUC and F-Score	KDD 99'Cup/DARPA	Feature Normalization and Wrapper method for Feature Selection	2017	[114]
	OS-ELM	Distributed	NIDS	Real Time	Empirical	Penetration attack	Accuracy, TPR, TFR	NSL-KDD	-	2018	[112]
	Deep Learning	Centralized	NIDS	-	Empirical	Penetration Attacks	Accuracy, DR, FPR, CPU	NSL-KDD, UNSW-NB15	-	2018	[116]
	DAE	Centralized	NIDS	-	Empirical	Botnet	TPR, FPR	Ad Hoc	-	2018	[117]
	MLP, Naive Bayes	-	NIDS	Offline	Simulation (Cooja)	Routing Attack (Rank, Version Number, Hello Flood)	Precision, Recall, F1 Score	Simulated (IRaD)	Wrapper	2018	[118]
	Multiple ML	Centralized	NIDS	Offline	Empirical	Penetration attack (U2R,R2L)	TPR, FPR	NSL-KDD	PCA and LDA	2016	[119]
		Centralized	HIDS	Offline	Empirical (MATLAB)	Application Layer attacks	Accuracy, TPR, FPR, TNR,FNR, Precision	Intel Lab IoT Dataset	-	2017	[120]
	K-Means, DT	Centralized	NIDS	-	Empirical	Routing (Wormhole)	TPR	Ad Hoc	-	2017	[121]
	kNN, C4.5, SVM	-	NIDS	Offline	Empirical	U2R, R2L	Accuracy, DR,FPR	NSL-KDD, KDD	Incremental Clustering	2018	[122]
	DT, RF	-	NIDS	Offline	Empirical	Conventional	Precision, Recall, F-Score	CICIDS2017, UNSW-15	Feature Selection-RFE	2019	[123]
Data Mining	**Jaccard Coefficient**	-	-	-	Empirical (MATLAB)	DoS	-	Ad Hoc	-	2014	[124]
Data Mining	**Clustering-DBSCAN**	Distributed-Autonomous	Hybrid	Offline	Empirical	DDoS	DR,FPR,FNR	Ad Hoc	PCA	2019	[125]
Signal Processing	**DWT**	-	-	Real Time	Empirical	DoS	Accuracy	Ad hoc	-	2018	[126]
	Blockchain-EMM	Distributed	HIDS	-	Empirical	Code Reuse and Code Injection	-	Ad Hoc	-	2018	[127]
Specification based											
Specification Based		Hybrid	NIDS	Real Time	-	Routing attack(Local Repair and Rank Attack)	-	-	-	2011	[128]
		Distributed	NIDS	Real Time	Empirical	DDoS	-	Ad Hoc	-	2011	[129]
		Hybrid	NIDS	Offline	Empirical	-	Memory Overhead	Ad Hoc	-	2014	[130]
		Distributed	NIDS	Real Time	Simulation (Cooja)	Routing attack	Energy Overhead	Simulated	-	2016	[131]
		Hybrid	Hybrid	Offline	Simulation (Cooja)	Routing attack (Rank attack, sinkhole, neighbor attack)	TPR, FPR, TNR, FNR, Energy Overhead	Simulated	-	2016	[132]
		Hybrid	NIDS	Real Time	Simulation (ns-2)	Routing attack (Sinkhole attack)	Energy Overhead	Simulated	-	2016	[134]
		Hybrid	NIDS	Real Time	Simulation (Cooja)-only attack	Routing attack (Sybil)	-	-	-	2017	[136]
		Distributed	HIDS	Real Time	Empirical	DDoS	Precision and Recall(TPR)	Apache Bench	-	2017	[137]
Hybrid											

TABLE 11. ***(Continued.)*** **Summary of proposed IDS for IOT. Reviewed works are grouped based on the proposed taxonomy in Fig.2.**

Detection Algorithm		Placement	Information Source	Usage Frequency	Validation Strategy	Security Attack	Evaluation Metric	Dataset	Data Preprocessing & Feature Selection	Year	Ref
Anomaly and Signature	Pattern Classifier and Anomaly Detector	Hybrid	NIDS	Offline	Simulation (SENSE)	DDoS	FP, FN and Detection Time	Simulated	-	2009	[138]
	Snort Signature and Cumulative Sum Control Chart	Distributed	NIDS		Simulation (ns-2)	DoS (UDP Flooding)	FPR, TPR, ROC Curve, Energy Consumption	Simulated	-	2009	[139]
		Hybrid	NIDS	Real Time	Simulation (Cooja)	Routing attack (Sinkhole, sybil, spoofing, clone ID and selective forwarding)	TPR, Memory and Energy Overhead	Simulated	-	2013	[46]
	Rule and Neural Network	-	NIDS		Simulation (OMNeT++)	MiTM and routing attacks	-	-	-	2014	[142]
		Hybrid	NIDS	Real Time	Simulation (Cooja)	Routing attack (Selective Forwarding and Rank)	FPR	Simulated	-	2014	[143]
	Rule and Anomaly	Distributed	HIDS		Simulation (ns2)	Routing attack (Selective forwarding)	Accuracy, TPR, Energy Consumption	Simulated	MAC Filtering	2015	[141]
	Rule and Game Theory	Distributed	NIDS	Offline	Simulation (TOSSSIM)	Routing Attack (Wormhole, Hello Flood, Spoofing, Black Hole, Sybil, Sinkhole), DoS	TPR, FPR and Accuracy, Energy Overhead	-	-	2016	[146], [147]
		Hybrid	NIDS	Real Time	Empirical	DoS and routing attacks (Selective Forwarding, ICMP Flood, Smurf attack, SYN flow)	TPR, Accuracy, CPU,Memory Overhead	-	-	2017	[144]
	Rules and Correlation	Distributed	Hybrid	Offline	-	DDoS and Botnet	-	-	Filter and Embedded Feature Selection	2017	[145]
		Hybrid	NIDS	Real Time	Simulation (Cooja)	Routing Attack (Rank, ETX)	TPR, Memory Overhead, Power	Simulated	-	2017	[48]
	Misuse and OPF	Hybrid	NIDS	Real Time	Simulation (MATLAB)	Routing Attacks (Collaborative of Selective Forwarding and Sinkhole)	-	NSL-KDD	MI-BGSA Wrapper	2017	[148]
	6 ML	Centralized	NIDS		Simulation (Cooja)	Combination	Accuracy, Energy, Memory	Simulated	Correlation Based	2018	[149]
		Distributed	NIDS	Real Time	Empirical	Malware	Memory, CPU	Koodous, apkmirror, Drebin	-	2019	[150]
Anomaly and Specification	Specification and Reputation Based	Distributed	NIDS	Offline	Simulation (Cooja)	Routing attack (Sinkhole)	TPR, FPR, FNR	Simulated	-	2015	[135]
	Specification and Unsupervised OPF	Hybrid	NIDS	Real Time	Simulation (MATLAB)	Routing Attack (Selective forwarding, Sinkhole, Wormhole and their collaboration)	TPR, FPR and Accuracy	Ad Hoc	Filter	2016	[151]
	-	Centralized	NIDS	Real Time	Empirical	DoS and routing attacks (Jamming, false-attack and replay attack)	-	-	-	2017	[152]

IPS and IRS, respectively. This is relevant particularly from the viewpoint of growing number of attacks on IoT networks. In the next section, we extend our discussion on the research challenges and opportunities to address intrusions in IoT.

VI. RESEARCH CHALLENGES AND OPPORTUNITIES

We proceed by discussing research challenges and opportunities in the area, which infer from our previous analysis of the various IDS, IPS and IRS techniques applies to IoT.

Since research for IDS in IoT is still in its nascent stage, this section also introduces ideas and possibilities of future research in this domain.

A. INTRUSION DETECTION IN IoT

1) DETECTION TECHNIQUE

The choice of detection technique is critical, since it decides how other characteristics of an IDS will be deployed. Anomaly-based detection are proposed more over misuse and specification based techniques, due to their more demanding memory requirements, and as such can detect zero day attacks. For these, there is a need to develop, analyze and compare more lightweight and optimized anomaly detection algorithms, particularly based on ML and DL for IoT networks. Additionally, the training phase is an important phase of learning in anomaly detection algorithms, which usually suffers from the problem of low accuracy and high false positives in attack detection. Thus, IDS should be trained for longer duration before deploying the network, in order to improve its accuracy and reduce false positives. In other situations, unsupervised anomaly detection algorithms also must be researched, since there is a lack of properly labelled datasets for IoT, which can be used for training beforehand. Besides, the integration of rule, anomaly and specification based detection techniques should be employed to avoid the drawbacks and get the benefits of all of them. This integration was equally emphasized for CPS by the authors in [174], [175], as well as for IoT in [29], [176]. As highlighted in [29], though it is true that the techniques must be tested on different IoT scenarios and applications, novel works that aim integration of anomaly and specification-based techniques must be researched. In this survey on IDS for IoT, it is evident that only a few works [135], [151], [152] focus on this integration and that, as such, more effort is required to refine this integration by enhanced network behavior modelling and by defining and thresholding network parameters to detect intrusions. In the case of hybrid methodology, Sedjelmaci *et al.* [146] argues to use anomaly detection only when a new attack is about to occur, and which was not detected by the signature based detection method. More such approaches need to be investigated, since these reduce the energy consumption of IoT devices.

2) INFORMATION SOURCE AND PLACEMENT STRATEGY

As is evident from the reviewed literature, information source of IDS is crucial, since it governs the visibility level of activities, an IDS can offer within the monitored system. For instance, an HIDS is limited to monitoring the host and therefore cannot monitor any sub-versions in the network traffic. While reviewing [100], [102], [125], [132] and [145], we observe that hybridized information is suitable to detect internal as well as external attacks on IoT devices and networks.

Regarding the placement strategy, a centralized approach acts as a single point of failure. Also, IoT networks are distributed networks, since they comprise of geographically distributed nodes. Secondly, the IDS placement also depends on the type of algorithm being implemented. If the algorithm simply matches attack signatures and specifications, a distributed approach on IoT devices can work. But, if the algorithm is any compute intensive machine learning operation, the training of such an algorithm may not be feasible on resource limited nodes. As a result, both these criterion plays an important role in deciding the placement strategy. The best strategy is to process compute intensive operations in resource rich gateways/server or cloud/fog nodes, with lightweight parts being processed in the IoT edge devices. Fog computing-based IDS have been discussed in [112], [113], [116]. Such IDS take advantage of unlimited cloud resources for training machine learning model, and also lesser detection latency than cloud based IDS. Consequently, a collaborative architecture utilizing information from IoT hosts and networks placed strategically on edge devices and gateways should be deeply investigated and utilized for future IDS IoT.

3) VALIDATION STRATEGY AND DATASETS

The best possible approach to test any IoT IDS is by using experimenting and simulating on IoT networks. With a number of devices available in the market, experimenting with real IoT devices is not a challenge. Regarding simulation, IoT network simulators like Cooja [133], which support standardized IoT protocols, have been used extensively in the recent years. A commercial alternative is NetSim [177], supporting the RPL protocol, altough it has not been explored by researchers. From our analysis, IoT simulators offer varying degree of support for protocols. For example, popular simulator like NS-2 [157] only provide native support for 6LoWPAN over IEEE 802.15.4 standard while lacking support for application layer protocols. Though improvements to base simulation models are possible, they only add to additional overheads. Thus, simulators trying to establish themselves in IoT research should thrive built-in support for all IoT protocols. Researchers are encouraged to go through the works in [178], which provides a deeper analysis of various simulation tools and open IoT testbeds for effective simulations and prototype evaluation.

The evaluation of IDS also requires appropriate data sets. Through our research, we identified that researchers seldom use datasets applicable for IoT networks, and this certainly raises the need for IoT benchmarked datasets. Due to the absence of such datasets, researchers have evaluated their IDS on contemporary datasets such as KDDCup, NSL-KDD and ISCXIDS, which do not represent true IoT systems and applications. To facilitate a throughout evaluation of IDS for IoT, we believe future research in recent IoT datasets like [60]–[62] and [59] is required. Rigorous analysis of these datasets for accurate labelling, attack diversity and credibility is envisaged. Such datasets will enable training and validation of ML models and allow real and logical comparison between different proposals.

TABLE 12. Summary of open research challenges for integrating second line of defense mechanisms with IoT.

Research Challenge	Description
Detection Technique, Placement and Validation Strategy of IDS	•Refined integration of specification, anomaly and signature based detection techniques to detect known, unknown and zero day attacks •Ensure automated analysis and continous learning with incremental machine learning techniques for streaming detection in real time •Need of collaborative architecture with strategic placement of IDS modules between the gateways, fog and cloud devices and IoT devices •Lack of support for end-to-end IoT service simulations
Multistage and Simultaneous Attacks on IoT Networks	•Design and development of multi-layered IDS for cross layer intrusion detection •IoT Intrusion datasets do not cater to traffic with simultaneous attacks
IoT Datasets and Pre-processing Techniques	•Lack of public IoT Datasets covering broad range of correctly labelled normal and malicious traffic •Irrelevant and noisy features in the dataset •Efficient feature reconstruction and dimension reduction algorithms based on deep learning techniques like auto encoders •Room for research on experimentation with different datasets, feature selection and detection techniques to analyze their applicability for IDS evaluation
Real Time Intrusion Prevention	•Designing an autonomous, adaptive and scalable prevention system that can handle large size packets and high speeds of IoT network systems.
Attack Mitigation with Intrusion Response Systems	•Mechanisms which handle false alarms generated by IDS •Real time response categorization according to nature of attack
IoT Device Constraints	•Lightweight and robust security modules for resource constrained IoT devices •Need of optimized computational and storage techniques in IoT devices and edge gateways like software accelerators.

4) MULTISTAGE AND SIMULTANEOUS ATTACKS

A quintessential attacker can perform an attack in multiple stages. Such advanced attacks are called multistage attacks and target conventional networks as well as advanced technologies as IoT. Except the works by Arshad *et al.* [100], existing proposals simply pivot on the detection of individual attacks, irrespective of relationships between them. The widely heterogeneous and vast nature of IoT networks makes the detection of these attacks more challenging. Explicit mechanisms that seek dependencies between different malicious incidents, irrespective of underlying technology, are required to protect IoT against multistage attacks. The initial efforts in [35], [73], [100] can help security researchers to develop such explicit mechanisms.

Further, it is possible that attackers launch simultaneous attacks in the IoT to completely bring the network down. For example, in a network two attackers can simultaneously launch selective forwarding attack and sinkhole attacks. Such simultaneous attacks were discussed by authors in [149], which can act as baseline to develop and evaluate IDS that can detect the occurrence simultaneous attacks. Also, application and adaptation layer attacks still need more attention, because adaptation layer is novel to IoT, and application layer communications are seldom secured using cryptographic techniques. According to [179], the assumption that application layer messages are usually encrypted, and that thus face less intrusions as compared to network layer, is a fallacy. In fact, all layers of the network stack require appropriate intrusion detection mechanisms. Also, due to the constrained nature of IoT devices, application layer is often not secured using cryptographic measures of encryption, authentication and integrity. Such a multi-layered IDS is proposed in [114], which can be used as a basis for proposing other multi-layered IDSs.

5) EVALUATION METRICS

Several metrics as accuracy, TPR, FPR, memory overhead have been used by researchers to evaluate IoT IDS, but if one has to compare the metrics of different proposals then they must be trained and tested using the same data and operational setting. Even if proposals use the same data set as NSL-KDD, while comparing, the authors did so in a vague way by using different subsets from the same dataset. Thus, the authenticity of these proposals is uncertain, as future research efforts may focus on similar operational conditions for evaluating and comparing IDS IoT proposal with other existing proposals in the literature.

6) DATA PREPROCESSING-DIMENSIONALITY REDUCTION

Whether it is intrusion detection or prevention, both are incomplete without the selection of accurate network features, which are fed into machine learning algorithms

during training process. Feature selection algorithms identify and select features from raw data that are needed to learn a task. In addition, to reduced computation and accurate results, the advantage of feature selection is that it reduces the amount of training data and time significantly especially on already constrained IoT devices. For signature based techniques, data pre-processing aims to minimize the storage space for signatures in IoT devices. It also tries to speed up the continual updates and analysis of attack signatures. In addition, protocol specific pre-processing involves parsing and normalizing fields of network and host traffic for signature generation. In the case of specification based techniques, the first stage of pre-processing involves modeling of network protocol as a finite state automation with all the possible states and transitions. In the second stage, packets are mapped to properties of state automation. Despite the number of proposals being signature and specification based, none of them discuss data pre-processing for these techniques, and as such may be explored by security researchers in IoT.

As for anomaly detection techniques, a number of works discuss data pre-processing spanning multiple methods and algorithms. Suitable feature extraction and selection algorithms, particularly based on deep learning, need to be proposed, in order to reduce the training and testing time required, and also to ensure accurate classification by the IDS. Algorithms like deep autoencoders used in [116], [117] are efficient in feature selection, and can act as baseline for developing further approaches. Additionally, software accelerators for deep learning as DeepX [180] help to control and optimize resources on constrained devices. In this regards, researchers can take inspirations from [181], [182]. Researchers may also compare and analyze different feature selection algorithms and its effect on the performance of detection algorithms in IoT applications and environments.

7) REAL TIME IDS

With the growing articulation of IoT with the physical world and thus with human lives, a real-time IDS is crucial. Real time IDS is significant since recent IoT devices include medical devices like pacemakers, inhalers etc. and if attackers compromise such devices and intrusion detection is not done in real time, it can prove to be fatal. The development and adaptation of lightweight detection modules for IoT increase its detection time and FPR. To address this, researchers have proposed two solutions: firstly, offline training of machine learning model and detecting intrusions on-line. However, this might not always be possible, considering the dynamic nature of IoT data streams and networks. In such a case, researchers may focus on **incremental ML**, which updates models in real-time, ensures continuity in learning and adapts to new data without the need to retrain the model. Such an approach can reduce time frame of detection and make IDS more intelligent. The second solution is based on implementing ML techniques on resource-rich network gateways or fog nodes, which helps to minimize delays and conceive near real time detection. However, fog computing for IoT IDS is still in its infancy, and needs further attention by researchers.

8) MULTI-CLASS CLASSIFICATION

Most of the IDS in the literature classify the activities either as malicious or normal, i.e. the classifications were binary classifications. Only the works in [113] evaluate their IDS on four classes of attack categories, namely DoS, probe, R2U, U2R and normal. However, IDS for IoT must not only detect malicious activity, but also identify the type of attack triggered. More works should be proposed that enable multi-class classification and is a promising approach.

9) OTHER ISSUES

IoT devices are also susceptible to physical attacks, and this certainly highlights the need for tamper resistant detection techniques. Additionally, research on agent security, and its placement in the physical domain, is also essential. In the IoT, since encrypting all the traffic between agents is both impractical and resource consuming, encryption and validation of intrusion alerts with hardware tamper resistant technologies can be utilized by IDS primarily to ensure authentication of alert generating IDS components. These technologies are based on TrustZones [183], Trusted Execution Environment (TEE) [184], physical unclonable functions and power fingerprinting [185]. Throughout a system, TrustZone technology isolates the software of the device from the hardware. In this technology, regions of memory designated as secure will not be compromised if other separated regions of memory are compromised. Though TrustZone is designed primarily to prevent software attacks, it also protects against simple hardware attacks, like shack attacks. In shack attacks, the attacker gain physical access to the device and attempt to connect to the device by boundary scan input/output and other built-in self test facilities [186]. Later, the attacker may force pins/lines to high or low voltage or even reprogram memory units. On the other hand, Trusted Execution Environment is made of a virtualized or separate security processor dedicated to run secure applications. In particular, Physical Unclonable functions are unique entities derived from manufacturing variations to ensure authentication in communications. Power fingerprinting [185] involves analysis of side channels like consumption of power or emissions of electromagnetic radiations to detect hardware Trojans or other integrity attacks. Additionally, identification of the time when the device is turned off or opened can help to detect tampering attacks, because a device has to be turned off to reverse engineer and add features to it. To the best of our knowledge, no paper is published which integrates IDS with these approaches that will detect and prevent tampering attacks in IoT.

B. INTRUSION PREVENTION IN IoT

Since IPS are placed inline, they introduce fundamental performance and stability issues in network that they are designed to protect. The inspection of data takes time to prevent intrusion and often comes at a price of slower

responsiveness and throughput of the network. Keeping such aspects in mind, IPS for IoT may be deployed and evaluated on gateways or critical hosts containing sensitive information with only IDS on the remaining sensor nodes. IPS techniques like pushback protocol, real time packet filtering, deep packet investigation for IoT systems need further investigation. A combination of IPS and IDS commonly referred to as IDPS may also be developed.

C. INTRUSION RESPONSE IN IoT

The IRS responds to intrusions and attempts to identify and trace the attack source, in order to stop malicious flows of traffic. Such techniques either track the complete lineage of data and actions on it, or trace IP packets from the destination back to the source, despite spoofed IP addresses. The former is provenance technique and the latter is termed as IP traceback technique. In IoT, provenance based RPL packet path tracing is proposed in [187]. In their novel lossless provenance scheme, the authors set a bit in IPv6 extension header, to acquire data generating node and path provenance. This scheme ensures data trustworthiness and integrity during packet traversal and in auditing and forensic analysis. IP Traceback in IoT has been briefly discussed by Amin *et al.* [138]. More information on its implementation can be found in [188].

D. NEXT GENERATION FIREWALL AND HONEYPOTS FOR IoT

A firewall protects a network from malicious activities that originate from outside the network. It filters the incoming and outgoing packets based on predefined set of rules. A firewall can be host based or network based, and can either run on hardware or software. A seminal work on firewall for IoT devices and networks was proposed by authors in [139]. Later, authors in [28] attempted to define a baseline framework of IDS augmented with a firewall. Frameworks such as these are significant promising approaches for researchers and, as such, firewalls for IoT should be comprehensively evaluated.

Honeypots, a popular deception tool has an important role to play in all the three areas of security which are prevention, detection and response. Some limited prevention by deception using honeypots in IoT can deter attackers by wasting their time and resources. They can also address challenges of false positives, false negatives and data aggregation in detection [189]. For response, honeypots may prove useful since they can be easily taken offline for forensic analysis. Finally, we can say that there is no real prevention, detection and response done by honeypots in IoT and further research is required.

E. SUMMARY OF LESSONS LEARNED

In this survey, we have covered the recent research contributions made towards securing the IoT with second line of defense systems particularly, intrusion detection, prevention and response systems. While such systems have great potential in establishing robust security for the Internet of things, it is not without its restrictions that need to be researched upon. In this section, we will summarize and revisit the knowledge we have gained in the context of IoT security, as discussed in this review.

The second line of defense mechanisms in IoT networks provide security against internal and external attacks even when cryptographic measures to secure communications are present. In providing security for the IoT, we learned how IDS, IPS and IRS enable detection, prevention and response of intrusions in IoT networks. An efficient IDS addresses both known and unknown intrusions in real time efficiently by consuming minimum resources of IoT device. Such systems must defend the IoT network and its resources without impacting network's performance. Also, the IoT consists of a broad range of devices, thus inter-operability issues must be kept in mind while designing such systems for IoT. The IoT networks have inherent power constraints thus, lightweight IDS mechanisms that incur minimum overhead on the delivery of necessary IoT service is needed. Due to resource limitations of IoT devices, the resource related metrics apart from detection related metrics are important performance metrics. Research efforts to ensure automated analysis with hybrid detection technique to detect known, unknown and zero day internal and external attacks are required.

Due to the inherit heterogeneity of data and the lack of IoT datasets, feature selection in IoT is a non-trivial task. Feature selection is challenging in IoT not only owing to the data heterogeneity but also due to the novel networking and communication technologies like MQTT, TSCH, BLE etc. that continuously produce ephemeral IoT data streams with redundant features. Following [114], [148], [151], better classification performance can be achieved by implementing a hybrid feature selection technique combining filter and wrapper methods. Lastly, according to [95]–[97], [119], [125], feature extraction is also a potential way to improve detection accuracy and reduce computational complexity of the intrusion detection model.

It can be learned from the literature that datasets from the perspective of IDS evaluation in IoT seldom exists. We observed that data sets currently do not comprehensively reflect the heterogeneous IoT network traffic patterns and modern attacks. For example, none of the most commonly used datasets like the KDDCup, NSL-KDD, ISCXIDS and CICIDS include attacks specific to IoT protocols of the standardized stack shown in Table 1. In such a situation, it is prudent to design adequate data sets for IoT like the ones in [60], [93] and [61] for design, development and evaluation of a reliable IDS. The researchers are encouraged to search or devise realistic IoT dataset with richer features that matches real world operational IoT environments.

First, it is learned that very few proposals discuss appropriate preventive measures to prevent IoT networks from internal and external attacks. These measures include limiting the number of request/response packets in a protocol and analyzing the ingress traffic either to a host or a network

for malicious behavior. Being inline, the efficiency of an IPS relies on its real time analysis with minimum network and system overhead. Following this, Misra *et al.* [129] propose to improve performance by sampling few packets and Saeed *et al.* [111] perform real time analysis by evaluating their IPS on RIPE [171].

The research in IRS for IoT is still in its nascent stage and need more attention of researchers. In the literature, [169] propose preliminary self-healing responses like disconnecting and delaying the connection. The characterization of responses according to anomalies as discussed by Qin *et al.* [125] is a progressive way of responding to intrusions. However, it must be ensured that this countermeasure does not reduce the network's availability, thus providing powerful mechanism for critical networks.

VII. CONCLUSION

The rapid development and integration of IoT technologies with physical world enable transformation of multiple application domains such as smart cities, smart homes and buildings, intelligent transportation system etc. However, the secure operation of IoT applications is also fundamental, since resource-constrained devices (things) are connected to the unreliable and untrusted Internet and hence, security at both the device and network levels becomes critical. In this context, intrusion detection, prevention and reaction systems act as second line of defence, after cryptographic measures, to provide security as sensing and actuating devices are progressvely integrated the the Internet communications architecture. In fact, such systems play a significant role to detect and prevent malicious activities from within or outside of the network.

To the best of our knowledge, our survey is the first proposal with comprehensive discussion of defensive security solutions for IoT systems. We present a taxonomy and perform an exhaustive analysis of existing IDS, IPS and IRS in IoT while focusing, in particular, in how these can be implemented to guarantee security in IoT. Given its importance, we structure our study only along the standardized protocols for IoT. As noted earlier, defensive proposals for conventional networks can be revisited, optimized and evaluated for its integration with IoT. Highlighting challenges surrounding these mechanisms, this survey attempts to motivate researchers to address challenges recognized in this article to secure IoT systems with defensive systems.

REFERENCES

[1] F. Javed, M. K. Afzal, M. Sharif, and B.-S. Kim, "Internet of Things (IoT) operating systems support, networking technologies, applications, and challenges: A comparative review," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2062–2100, 3rd Quart., 2018.

[2] I. Markit, "The Internet of Things: A movement, not a market," IHS Markit, London, U.K., Tech. Rep., 2018. Accessed: Jan. 2, 2020. [Online]. Available: https://cdn.ihs.com/www/pdf/IoT_ebook.pdf

[3] S. E. Deering and R. M. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, RFC, document 2460, Dec. 1998.

[4] OWASP. (2018). *Owasp-IoT-top-10-2018*. Accessed: May 2, 2019. [Online]. Available: https://www.owasp.org/images/1/1c/OWASP-IoT-Top-10-2018-final.pdf

[5] O. Garcia-Morchon, S. Kumar, S. Keoh, R. Hummen, and R. Struik, *Security Considerations in the ip-Based Internet of Things, Internet-Draft Draft-Garcia-Core-Security-06*, IETF Secretariat, New York, NY, USA, Sep. 2013.

[6] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[7] K. Panetta. (2016). *Gartner's Top 10 Security Predictions 2016*. Accessed: Mar. 2, 2019. [Online]. Available: https://www.gartner.com/smarterwithgartner/top-10-security-predictions-2016

[8] E. Rescorla and N. Modadugu, *Datagram Transport Layer Security Version 1.2*, RFC, document 6347, Jan. 2012.

[9] J. Granjal, E. Monteiro, and J. S. Silva, "Security in the integration of low-power wireless sensor networks with the Internet: A survey," *Ad Hoc Netw.*, vol. 24, pp. 264–287, Jan. 2015.

[10] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia, and M. Dohler, "Standardized protocol stack for the Internet of (Important) Things," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1389–1406, 3rd Quart., 2013.

[11] *IEEE 802.15.4-2015/Cor 1-2018–IEEE Standard for Low-Rate Wireless Networks corrigendum 1*, I. S. Association, Standard 802.15.4-2015, 2015.

[12] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, *Transmission of IPv6 Packets Over IEEE 802.15.4 Networks*, RFC, document 4944, Sep. 2007.

[13] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, *Rpl: Ipv6 Routing Protocol for Low-Power and Lossy Networks*, RFC, document 6550, Mar. 2012.

[14] Z. Shelby, K. Hartke, and C. Bormann, *The Constrained Application Protocol (CoAP)*, RFC, document 7252, Jun. 2014.

[15] M. Malik, Kamaldeep, and M. Dutta, "Defending DDoS in the insecure Internet of Things: A survey," in *Artificial Intelligence and Evolutionary Computations in Engineering Systems*. Heidelberg, Germany: Springer, 2018, pp. 223–233.

[16] A. Mayzaud, R. Badonnel, I. Chrisment, and I. G. Est-Nancy, "A taxonomy of attacks in RPL-based Internet of Things," *Int. J. Netw. Secur.*, vol. 18, no. 3, pp. 459–473, 2016.

[17] J. Postel, *Transmission Control Protocol*, IETF, Standard 7, Sep. 1981. [Online]. Available: http://www.rfc-editor.org/rfc/rfc793.txt

[18] J. Postel, *User Datagram Protocol*, IETF, Standard 6, Aug. 1980. [Online]. Available: http://www.rfc-editor.org/rfc/rfc768.txt

[19] J. P. Anderson, "Computer security threat monitoring and surveillance," James P. Anderson Co., Washington, DC, USA, Tech. Rep. 215 646-4706, Apr. 1980. Accessed: Feb. 2, 2020. [Online]. Available: http://seclab.cs.ucdavis.edu/projects/history/papers/ande80.pdf

[20] A.-S. K. Pathan, *Securing Cyber-Physical Systems*. Boca Raton, FL, USA: CRC Press, 2016.

[21] L. Wallgren, S. Raza, and T. Voigt, "Routing attacks and countermeasures in the RPL-based Internet of Things," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 8, Aug. 2013, Art. no. 794326.

[22] A. Le, J. Loo, A. Lasebae, M. Aiash, and Y. Luo, "6LoWPAN: A study on QoS security threats and countermeasures using intrusion detection system approach," *Int. J. Commun. Syst.*, vol. 25, no. 9, pp. 1189–1212, Sep. 2012.

[23] P. Pongle and G. Chavan, "A survey: Attacks on RPL and 6LoWPAN in IoT," in *Proc. Int. Conf. Pervasive Comput. (ICPC)*, Pune, India, Jan. 2015, pp. 1–6.

[24] A. Rghioui, A. Khannous, and M. Bouhorma, "Denial-of-service attacks on 6lowpan-RPL networks: Issues and practical solutions," *J. Adv. Comput. Sci. Technol.*, vol. 3, no. 2, pp. 143–153, 2014.

[25] L. Zhang, G. Feng, and S. Qin, *Intrusion Detection System for Low-Power and Lossy Networks*, Internet-Draft Draft-Zhang-roll-RPL-Intrusion-Defence-00, IETF Secretariat, New York, NY, USA, Nov. 2013.

[26] K. Zhang, X. Liang, R. Lu, and X. Shen, "Sybil attacks and their defenses in the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 5, pp. 372–383, Oct. 2014.

[27] A. A. Gendreau and M. Moorman, "Survey of intrusion detection systems towards an end to end secure Internet of Things," in *Proc. IEEE 4th Int. Conf. Future Internet Things Cloud (FiCloud)*, Aug. 2016, pp. 84–90.

[28] H. B. Patel, D. C. Jinwala, and D. R. Patel, "Baseline intrusion detection framework for 6LoWPAN devices," in *Proc. Adjunct 13th Int. Conf. Mobile Ubiquitous Syst., Comput. Netw. Services (MOBIQUITOUS)*, New York, NY, USA, 2016, pp. 72–76.

[29] B. B. Zarpel ao, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *J. Netw. Comput. Appl.*, vol. 84, pp. 25–37, Apr. 2017.

[30] M. Ali Al-Garadi, A. Mohamed, A. Al-Ali, X. Du, and M. Guizani, "A survey of machine and deep learning methods for Internet of Things (IoT) security," 2018, *arXiv:1807.11023*. [Online]. Available: http://arxiv.org/abs/1807.11023

[31] J. Arshad, M. Ajmal Azad, K. Salah, W. Jie, R. Iqbal, and M. Alazab, "A review of performance, energy and privacy of intrusion detection systems for IoT," 2018, *arXiv:1812.09160*. [Online]. Available: http://arxiv.org/abs/1812.09160

[32] E. Benkhelifa, T. Welsh, and W. Hamouda, "A critical review of practices and challenges in intrusion detection systems for IoT: Toward universal and resilient systems," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3496–3509, 4th Quart., 2018.

[33] M. F. Elrawy, A. I. Awad, and H. F. A. Hamed, "Intrusion detection systems for IoT-based smart environments: A survey," *J. Cloud Comput.*, vol. 7, no. 1, p. 21, Dec. 2018.

[34] S. Choudhary and N. Kesswani, "A survey: Intrusion detection techniques for Internet of Things," *Int. J. Inf. Secur. Privacy (IJISP)*, vol. 13, no. 1, pp. 86–105, 2019.

[35] K. A. P. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, "Internet of Things: A survey on machine learning-based intrusion detection approaches," *Comput. Netw.*, vol. 151, pp. 147–157, Mar. 2019.

[36] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for IoT security based on learning techniques," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2671–2701, 3rd Quart., 2019.

[37] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 266–282, 1st Quart., 2014.

[38] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, p. 15, 2009.

[39] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2016.

[40] F. Vancea and C. Vancea, "Some results on intrusion and anomaly detection using signal processing and NEAR system," in *Proc. 38th Int. Conf. Telecommun. Signal Process. (TSP)*, Jul. 2015, pp. 113–116.

[41] P. Brutch and C. Ko, "Challenges in intrusion detection for wireless ad-hoc networks," in *Proc. Symp. Appl. Internet Workshops*, Orlando, FL, USA, Jan. 2003, pp. 368–373.

[42] D. Moon, S. B. Pan, and I. Kim, "Host-based intrusion detection system for secure human-centric computing," *J. Supercomput.*, vol. 72, no. 7, pp. 2520–2536, Jul. 2016.

[43] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, Jan. 2013.

[44] M. V. Zelkowitz, "Techniques for empirical validation," in *Empirical Software Engineering Issues. Critical Assessment and Future Directions*. Heidelberg, Germany: Springer, 2007, pp. 4–9.

[45] M. Abomhara, "Cyber security and the Internet of Things: Vulnerabilities, threats, intruders and attacks," *J. Cyber Secur. Mobility*, vol. 4, no. 1, pp. 65–88, 2015.

[46] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," *Ad Hoc Netw.*, vol. 11, no. 8, pp. 2661–2674, Nov. 2013.

[47] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," in *Proc. 1st IEEE Int. Workshop Sensor Netw. Protocols Appl.*, May 2003, pp. 113–127.

[48] D. Shreenivas, S. Raza, and T. Voigt, "Intrusion detection in the RPL-connected 6LoWPAN networks," in *Proc. 3rd ACM Int. Workshop IoT Privacy, Trust, Secur. (IoTPTS)*, New York, NY, USA, 2017, pp. 31–38.

[49] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 DARPA off-line intrusion detection evaluation," *Comput. Netw.*, vol. 34, no. 4, pp. 579–595, Oct. 2000.

[50] N. Moustafa, J. Hu, and J. Slay, "A holistic review of network anomaly detection systems: A comprehensive survey," *J. Netw. Comput. Appl.*, vol. 128, pp. 33–55, Feb. 2019.

[51] M. V. Mahoney and P. K. Chan, "An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection," in *Proc. Int. Workshop Recent Adv. Intrusion Detection*. Heidelberg, Germany: Springer, 2003, pp. 220–237.

[52] S. Hettich and S. D. Bay. (1999). *KDD Cup 1999 Data*. Accessed: Jul. 4, 2019. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[53] C. I. for Cybersecurity. (2009). *NSL-KDD Dataset*. Accessed: Nov. 10, 2019. [Online]. Available:https://www.unb.ca/cic/datasets/nsl.html

[54] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Comput. Secur.*, vol. 31, no. 3, pp. 357–374, May 2012.

[55] C. I. for Cybersecurity. *Intrusion Detection Evaluation Dataset (ISCXIDS2012)*. Accessed: Jun. 10, 2019. [Online]. Available: https://www.unb.ca/cic/datasets/ids.html

[56] C. I. for Cybersecurity. *Intrusion detection evaluation dataset (CICIDS2017)*. Accessed: Sep. 29, 2019. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html

[57] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. ICISSP*, 2018, pp. 108–116.

[58] A. H. Lashkari, G. Draper-Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," in *Proc. ICISSP*, 2017, pp. 253–262.

[59] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2015, pp. 1–6.

[60] A. Sivanathan, D. Sherratt, H. H. Gharakheili, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Characterizing and classifying IoT traffic in smart cities and campuses," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, May 2017, pp. 559–564.

[61] A. Verma and V. Ranga. *RPL-Nidds17—A Data Set for Intrusion Detection in RPL Based 6Lowpan Networks (Internet of Things)*. Accessed: May 2, 2019. [Online]. Available: https://zenodo.org/record/1406034#.XLfrI-gzbIU

[62] V. H. Bezerra, V. G. T. da Costa, R. A. Martins, S. B. Junior, R. S. Miani, and B. B. Zarpelao, "Providing IoT host-based datasets for intrusion detection research*," in *Proc. SBSeg*, 2018, pp. 15–28.

[63] K. Angrishi, "Turning Internet of Things (IoT) into Internet of vulnerabilities (IoV): IoT botnets," Feb. 2017, *arXiv:1702.03681*. [Online]. Available: https://arxiv.org/abs/1702.03681

[64] K. A. Al-Utaibi and E.-S.-M. El-Alfy, "Intrusion detection taxonomy and data preprocessing mechanisms," *J. Intell. Fuzzy Syst.*, vol. 34, no. 3, pp. 1369–1383, Mar. 2018.

[65] E. Carter and J. Hogue, *Intrusion Prevention Fundamentals*. London, U.K.: Pearson, 2006.

[66] Z. Inayat, A. Gani, N. B. Anuar, M. K. Khan, and S. Anwar, "Intrusion response systems: Foundations, design, and challenges," *J. Netw. Comput. Appl.*, vol. 62, pp. 53–74, Feb. 2016.

[67] V. Paxson. (2018). *The Zeek Network Security Monitor*. [Online]. Available: https://www.zeek.org

[68] S. Anwar, J. Mohamad Zain, M. F. Zolkipli, Z. Inayat, S. Khan, B. Anthony, and V. Chang, "From intrusion detection to an intrusion response system: Fundamentals, requirements, and future directions," *Algorithms*, vol. 10, no. 2, p. 39, Mar. 2017.

[69] N. Stakhanova, S. Basu, and J. Wong, "A taxonomy of intrusion response systems," *Int. J. Inf. Comput. Secur.*, vol. 1, nos. 1–2, pp. 169–184, 2007.

[70] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits, "Denial-of-Service detection in 6LoWPAN based Internet of Things," in *Proc. IEEE 9th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2013, pp. 600–607.

[71] P. Kasinathan, G. Costamagna, H. Khaleel, C. Pastrone, and M. A. Spirito, "DEMO: An IDS framework for Internet of Things empowered by 6LoWPAN," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Berlin, Germany, 2013, pp. 1337–1340.

[72] D. Oh, D. Kim, and W. Ro, "A malicious pattern detection engine for embedded security systems in the Internet of Things," *Sensors*, vol. 14, no. 12, pp. 24188–24211, Dec. 2014.

[73] C. Jun and C. Chi, "Design of complex event-processing IDS in Internet of Things," in *Proc. 6th Int. Conf. Measuring Technol. Mechatronics Autom.*, Jan. 2014, pp. 226–229.

[74] A. Sforzin, F. G. Mármol, M. Conti, and J.-M. Bohli, "RPiDS: Raspberry Pi IDS—A fruitful intrusion detection system for IoT," in *Proc. Int. IEEE Conf. Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People, Smart World Congr. (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, Toulouse, France, Jul. 2016, pp. 440–448.

[75] O. I. S. Foundation. (2010). *Suricata: Open Source IDS/IPS/NSM Engine*. [Online].https://suricata-ids.org/
[76] T. S. Project. (2018). *The Open Source Network Intrusion Detection System*. [Online].http://www.snort.org
[77] L. Johansson and O. Olsson, "Improving intrusion detection for IoT networks," M.S. thesis, Dept. Comput. Sci. Eng., Univ. Gothenburg, Gothenburg, Sweden, 2018.
[78] A. M. da Silva Cardoso, R. F. Lopes, A. S. Teles, and F. B. V. Magalhães, "Real-time DDoS detection based on complex event processing for IoT," in *Proc. IEEE/ACM 3rd Int. Conf. Internet–Things Design Implement. (IoTDI)*, Apr. 2018, pp. 273–274.
[79] J. Granjal and A. Pedroso, "An intrusion detection and prevention framework for Internet-integrated CoAP WSN," *Secur. Commun. Netw.*, vol. 2018, pp. 1–14, Apr. 2018.
[80] E. J. Cho, J. H. Kim, and C. S. Hong, "Attack model and detection scheme for botnet on 6LoWPAN," in *Proc. Asia–Pacific Netw. Oper. Manage. Symp.* Jeju, South Korea: Springer, 2009, pp. 515–518.
[81] M. L. Han, J. Lee, A. R. Kang, S. Kang, J. K. Park, and H. K. Kim, "A statistical-based anomaly detection method for connected cars in Internet of Things environment," in *Proc. Int. Conf. Internet Vehicles*. Springer, 2015, pp. 89–97.
[82] F. Gara, L. Ben Saad, and R. Ben Ayed, "An intrusion detection system for selective forwarding attack in IPv6-based mobile WSNs," in *Proc. 13th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2017, pp. 276–281.
[83] J. Habibi, D. Midi, A. Mudgerikar, and E. Bertino, "Heimdall: Mitigating the Internet of insecure things," *IEEE Internet Things J.*, vol. 4, no. 4, pp. 968–978, Aug. 2017.
[84] T.-H. Lee, C.-H. Wen, L.-H. Chang, H.-S. Chiang, and M.-C. Hsieh, "A lightweight intrusion detection scheme based on energy consumption analysis in 6LowPAN," in *Advanced Technologies, Embedded and Multimedia for Human-Centric Computing*. Amsterdam, The Netherlands: Springer, 2014, pp. 1205–1213.
[85] D. H. Summerville, K. M. Zach, and Y. Chen, "Ultra-lightweight deep packet anomaly detection for Internet of Things devices," in *Proc. IEEE 34th Int. Perform. Comput. Commun. Conf. (IPCCC)*, Dec. 2015, pp. 1–8.
[86] P. Pongle and G. Chavan, "Real time intrusion and wormhole attack detection in Internet of Things," *Int. J. Comput. Appl.*, vol. 121, no. 9, pp. 1–9, Jul. 2015.
[87] A. Mayzaud, R. Badonnel, and I. Chrisment, "A distributed monitoring strategy for detecting version number attacks in RPL-based networks," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 2, pp. 472–486, Jun. 2017.
[88] Y. Meidan, M. Bohadana, A. Shabtai, M. Ochoa, N. O. Tippenhauer, J. D. Guarnizo, and Y. Elovici, "Detection of unauthorized IoT devices using machine learning techniques," 2017, *arXiv:1709.04647*. [Online]. Available: http://arxiv.org/abs/1709.04647
[89] M. Domb, E. Bonchek-Dokow, and G. Leshem, "Lightweight adaptive random-forest for IoT rule generation and execution," *J. Inf. Secur. Appl.*, vol. 34, pp. 218–224, Jun. 2017.
[90] M. Nobakht, V. Sivaraman, and R. Boreli, "A host-based intrusion detection and mitigation framework for smart home IoT using OpenFlow," in *Proc. 11th Int. Conf. Availability, Rel. Secur. (ARES)*, Aug. 2016, pp. 147–156.
[91] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, 2011.
[92] M. J. Orr, "Introduction to radial basis function networks," Centre Cogn. Sci., Univ. Edinburgh, Edinburgh, U.K., Tech. Rep., 1996.
[93] V. H. Bezerra, V. G. T. da Costa, S. B. Junior, R. S. Miani, and B. B. Zarpelao, "One-class classification to detect Botnets in IoT devices*," in *Proc. SBSeg*, 2018, pp. 43–56.
[94] S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo, "Toward a lightweight intrusion detection system for the Internet of Things," *IEEE Access*, vol. 7, pp. 42450–42471, 2019.
[95] J. Granjal, J. Silva, and N. Lourenço, "Intrusion detection and prevention in CoAP wireless sensor networks using anomaly detection," *Sensors*, vol. 18, no. 8, p. 2445, Jul. 2018.
[96] L. Deng, D. Li, X. Yao, D. Cox, and H. Wang, "Mobile network intrusion detection for IoT system based on transfer learning algorithm," *Cluster Comput.*, vol. 22, pp. 9889–9904, Jan. 2018.
[97] L. Liu, B. Xu, X. Zhang, and X. Wu, "An intrusion detection method for Internet of Things based on suppressed fuzzy clustering," *EURASIP J. Wireless Commun. Netw.*, vol. 2018, no. 1, p. 113, Dec. 2018.
[98] N. K. Thanigaivelan, E. Nigussie, R. K. Kanth, S. Virtanen, and J. Isoaho, "Distributed internal anomaly detection system for Internet-of-Things," in *Proc. 13th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2016, pp. 319–320.
[99] N. K. Thanigaivelan, E. Nigussie, S. Virtanen, and J. Isoaho, "Hybrid internal anomaly detection system for IoT: Reactive nodes with cross-layer operation," *Secur. Commun. Netw.*, vol. 2018, pp. 1–15, Aug. 2018.
[100] J. Arshad, M. A. Azad, M. Mahmoud Abdellatif, M. H. Ur Rehman, and K. Salah, "COLIDE: A collaborative intrusion detection framework for Internet of Things," *IET Netw.*, vol. 8, no. 1, pp. 3–14, Jan. 2019.
[101] V. Sharma, I. You, and R. Kumar, "ISMA: Intelligent sensing model for anomalies detection in cross platform OSNs with a case study on IoT," *IEEE Access*, vol. 5, pp. 3284–3301, 2017.
[102] A. Gupta, O. J. Pandey, M. Shukla, A. Dadhich, S. Mathur, and A. Ingle, "Computational intelligence based intrusion detection systems for wireless communication and pervasive computing networks," in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res.*, Dec. 2013, pp. 1–7.
[103] Z. A. Khan and P. Herrmann, "A trust based distributed intrusion detection mechanism for Internet of Things," in *Proc. IEEE 31st Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Mar. 2017, pp. 1169–1176.
[104] R. Chen, C. M. Liu, and C. Chen, "An artificial immune-based distributed intrusion detection model for the Internet of Things," *Adv. Mater. Res.*, vol. 366, pp. 165–168, Apr. 2012.
[105] C. M. Liu, S. Y. Chen, Y. Zhang, R. Chen, and K. L. Guo, "An IoT anomaly detection model based on artificial immunity," *Adv. Mater. Res.*, vol. 424, pp. 625–628, Feb. 2012.
[106] C. M. Liu, Y. Zhang, R. Chen, L. X. Xiao, and J. D. Zhang, "Research on intrusion detection for the Internet of Things based on clone selection principle," *Adv. Mater. Res.*, vol. 562, pp. 1982–1985, Sep. 2012.
[107] F. Hosseinpour, P. Vahdani Amoli, J. Plosila, T. Hämäläinen, and H. Tenhunen, "An intrusion detection system for fog computing and IoT based logistic systems using a smart data approach," *Int. J. Digit. Content Technol. Appl.*, vol. 10, no. 5, pp. 34–46, 2016.
[108] Y. Shi, T. Li, R. Li, X. Peng, and P. Tang, "An immunity-based IOT environment security situation awareness model," *J. Comput. Commun.*, vol. 5, no. 7, p. 182, 2017.
[109] E. Hodo, X. Bellekens, A. Hamilton, P.-L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of IoT networks using artificial neural network intrusion detection system," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, May 2016, pp. 1–6.
[110] J. Roux, E. Alata, G. Auriol, V. Nicomette, and M. Kaaniche, "Toward an intrusion detection approach for IoT based on radio communications profiling," in *Proc. 13th Eur. Dependable Comput. Conf. (EDCC)*, Sep. 2017, pp. 147–150.
[111] A. Saeed, A. Ahmadinia, A. Javed, and H. Larikani, "Intelligent intrusion detection in low-power IoTs," *ACM Trans. Internet Technol.*, vol. 16, no. 4, p. 27, 2016.
[112] S. Prabavathy, K. Sundarakantham, and S. M. Shalinie, "Design of cognitive fog computing for intrusion detection in Internet of Things," *J. Commun. Netw.*, vol. 20, no. 3, pp. 291–298, Jun. 2018.
[113] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," *Future Gener. Comput. Syst.*, vol. 82, pp. 761–768, May 2018.
[114] M. Putchala, "Deep learning approach for intrusion detection system (IDS) in the Internet of Things (IoT) network using gated recurrent neural networks (GRU)," M.S. thesis, Graduate School, Wright State Univ., Dayton, OH, USA, 2017.
[115] S. Chawla, "Deep learning based intrusion detection system for Internet of Things," M.S. thesis, Dept. Comput. Softw. Syst., Univ. Washington, Seattle, WA, USA, 2017.
[116] M. AL-Hawawreh, N. Moustafa, and E. Sitnikova, "Identification of malicious activities in industrial Internet of Things based on deep learning models," *J. Inf. Secur. Appl.*, vol. 41, pp. 1–11, Aug. 2018.
[117] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-BaioT—Network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Comput.*, vol. 17, no. 3, pp. 12–22, Jul./Sep. 2018.
[118] F. Y. Yavuz, D. Ünal, and E. Gül, "Deep learning for detection of routing attacks in the Internet of Things," *Int. J. Comput. Intell. Syst.*, vol. 12, no. 1, pp. 39–58, 2018.
[119] H. H. Pajouh, R. Javidan, R. Khayami, A. Dehghantanha, and K.-K.-R. Choo, "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks," *IEEE Trans. Emerg. Topics Comput.*, vol. 7, no. 2, pp. 314–323, Apr. 2019.

[120] A. Alghuried, "A model for anomalies detection in Internet of Things (IoT) using inverse weight clustering and decision tree," Ph.D. dissertation, School Comput., Dublin Inst. Technol., Dublin, Ireland, Feb. 2017.
[121] P. Shukla, "ML-IDS: A machine learning approach to detect wormhole attacks in Internet of Things," in *Proc. Intell. Syst. Conf. (IntelliSys)*, Sep. 2017, pp. 234–240.
[122] S. A. Aljawarneh and R. Vangipuram, "GARUDA: Gaussian dissimilarity measure for feature representation and anomaly detection in Internet of Things," *J. Supercomput.*, vol. 76, pp. 4376–4413, May 2018.
[123] I. Ullah and Q. H. Mahmoud, "A two-level hybrid model for anomalous activity detection in IoT networks," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2019, pp. 1–6.
[124] Y. Liu and Q. Wu, "A lightweight anomaly mining algorithm in the Internet of Things," in *Proc. IEEE 5th Int. Conf. Softw. Eng. Service Sci.*, Jun. 2014, pp. 1142–1145.
[125] T. Qin, B. Wang, R. Chen, Z. Qin, and L. Wang, "IMLADS: Intelligent maintenance and lightweight anomaly detection system for Internet of Things," *Sensors*, vol. 19, no. 4, p. 958, Feb. 2019.
[126] J. Pacheco and S. Hariri, "Anomaly behavior analysis for IoT sensors," *Trans. Emerg. Telecommun. Technol.*, vol. 29, no. 4, p. e3188, 2018.
[127] T. Golomb, Y. Mirsky, and Y. Elovici, "CIoTA: Collaborative IoT anomaly detection via blockchain," 2018, *arXiv:1803.03807*. [Online]. Available: http://arxiv.org/abs/1803.03807
[128] A. Le, J. Loo, Y. Luo, and A. Lasebae, "Specification-based IDS for securing RPL from topology attacks," in *Proc. IFIP Wireless Days (WD)*, Oct. 2011, pp. 4–6.
[129] S. Misra, P. V. Krishna, H. Agarwal, A. Saxena, and M. S. Obaidat, "A learning automata based solution for preventing distributed denial of service in Internet of Things," in *Proc. Int. Conf. Internet Things 4th Int. Conf. Cyber, Phys. Social Comput.*, Oct. 2011, pp. 114–122.
[130] J. P. Amaral, L. M. Oliveira, J. J. P. C. Rodrigues, G. Han, and L. Shu, "Policy and network-based intrusion detection system for IPv6-enabled wireless sensor networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, Jun. 2014, pp. 1796–1801.
[131] K. Grgic, D. Zagar, and V. Krizanovic Cik, "System for malicious node detection in IPv6-based wireless sensor networks," *J. Sensors*, vol. 2016, Jul. 2016, Art. no. 6206353.
[132] A. Le, J. Loo, K. Chai, and M. Aiash, "A specification-based IDS for detecting attacks on RPL-based network topology," *Information*, vol. 7, no. 2, p. 25, May 2016.
[133] C. Simulator. (2016). *Running Cooja Simulator*. [Online]. Available: [Online]. Available: http://anrg.usc.edu/contiki/index.php/Cooja_Simulator
[134] M. Surendar and A. Umamakeswari, "InDReS: An intrusion detection and response system for Internet of Things with 6LoWPAN," in *Proc. Int. Conf. Wireless Commun., Signal Process. Netw. (WiSPNET)*, Mar. 2016, pp. 1903–1908.
[135] C. Cervantes, D. Poplade, M. Nogueira, and A. Santos, "Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, Ottawa, ON, Canada, May 2015, pp. 606–611.
[136] F. Medjek, D. Tandjaoui, I. Romdhani, and N. Djedjig, "A trust-based intrusion detection system for mobile RPL based networks," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Jun. 2017, pp. 735–742.
[137] T. Kawamura, M. Fukushi, Y. Hirano, Y. Fujita, and Y. Hamamoto, "An NTP-based detection module for DDoS attacks on IoT," in *Proc. IEEE Int. Conf. Consum. Electron.-Taiwan (ICCE-TW)*, Jun. 2017, pp. 15–16.
[138] S. O. Amin, Y. J. Yoon, M. S. Siddiqui, and C. S. Hong, "A novel intrusion detection framework for IP-based sensor networks," in *Proc. Int. Conf. Inf. Netw.*, Chiang Mai, India, Jan. 2009, pp. 1–3.
[139] S. O. Amin, M. S. Siddiqui, C. S. Hong, and S. Lee, "RIDES: Robust intrusion detection system for IP-based ubiquitous sensor networks," *Sensors*, vol. 9, no. 5, pp. 3447–3468, May 2009.
[140] C. Liu, J. Yang, R. Chen, Y. Zhang, and J. Zeng, "Research on immunity-based intrusion detection technology for the Internet of Things," in *Proc. 7th Int. Conf. Natural Comput.*, vol. 1, Jul. 2011, pp. 212–216.
[141] N. Alajmi and K. Elleithy, "Multi-layer approach for the detection of selective forwarding attacks," *Sensors*, vol. 15, no. 11, pp. 29332–29345, Nov. 2015.
[142] J. Krimmling and S. Peter, "Integration and evaluation of intrusion detection for CoAP in smart city applications," in *Proc. IEEE Conf. Commun. Netw. Secur.*, San Francisco, CA, USA, Oct. 2014, pp. 73–78.
[143] T. Matsunaga, K. Toyoda, and I. Sasase, "Low false alarm rate RPL network monitoring system by considering timing inconstancy between the rank measurements," in *Proc. 11th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Barcelona, Spain, Aug. 2014, pp. 427–431.
[144] D. Midi, A. Rullo, A. Mudgerikar, and E. Bertino, "Kalis—A system for knowledge-driven adaptable intrusion detection for the Internet of Things," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Atlanta, GA, USA, Jun. 2017, pp. 656–666.
[145] M. Gajewski, J. M. Batalla, G. Mastorakis, and C. X. Mavromoustakis, "A distributed IDS architecture model for Smart Home systems," *Cluster Comput.*, vol. 22, pp. 1739–1749, Aug. 2017.
[146] H. Sedjelmaci, S. M. Senouci, and M. Al-Bahri, "A lightweight anomaly detection technique for low-resource IoT devices: A game-theoretic methodology," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.
[147] H. Sedjelmaci, S. M. Senouci, and T. Taleb, "An accurate security game for low-resource IoT devices," *IEEE Trans. Veh. Technol.*, vol. 66, no. 10, pp. 9381–9393, Oct. 2017.
[148] M. Sheikhan and H. Bostani, "A security mechanism for detecting intrusions in Internet of Things using selected features based on MI-BGSA," *Inf. Commun. Technol. Res.*, vol. 9, no. 2, pp. 53–62, 2017.
[149] M. N. Napiah, M. Y. I. B. Idris, R. Ramli, and I. Ahmedy, "Compression header analyzer intrusion detection system (cha-ids) for 6lowpan communication protocol," *IEEE Access*, vol. 6, pp. 16623–16638, 2018.
[150] P. Nespoli, D. U. Pelaez, D. D. López, and F. G. Mármol, "COSMOS: Collaborative, seamless and adaptive sentinel for the Internet of Things," *Sensors*, vol. 19, no. 7, p. 1492, Mar. 2019.
[151] H. Bostani and M. Sheikhan, "Hybrid of anomaly-based and specification-based IDS for Internet of Things using unsupervised OPF based on MapReduce approach," *Comput. Commun.*, vol. 98, pp. 52–71, Jan. 2017.
[152] Y. Fu, Z. Yan, J. Cao, O. Koné, and X. Cao, "An automata based intrusion detection method for Internet of Things," *Mobile Inf. Syst.*, vol. 2017, May 2017, Art. no. 1750637.
[153] HardKernel. (2015). *Odroid-XU4 User Manual*. Accessed: Mar. 28, 2018. [Online]. Available: https://magazine.odroid.com/wp-content/uploads/odroid-xu4-user-manual.pdf
[154] P. Thubert, C. Perkins, and E. Levy-Abegnoli, "IPv6 backbone router," IETF, Fremont, CA, USA, Tech. Rep., Feb. 2019. Accessed: Oct. 20, 2019. [Online]. Available: https://tools.ietf.org/html/draft-ietf-6lo-backbone-router-11
[155] C. Bormann, M. Ersue, and A. Keranen, *Terminology for Constrained-Node Networks*, RFC, document 7228, May 2014.
[156] I. SCALABLE Network Technologies. (2018). *Qualnet*. [Online]. Available: https://web.scalable-networks.com/qualnet-network-simulator-software
[157] T. V. Project. *The Network Simulator, NS-2*. Accessed: May 2, 2019. [Online]. Available: https://www.isi.edu/nsnam/ns/
[158] O. Ltd. (2018). *Omnet++*. [Online]. Available: http://www.omnetpp.org/
[159] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and scalable simulation of entire TinyOS applications," in *Proc. 1st Int. Conf. Embedded Netw. Sensor Syst.*, 2003, pp. 126–137.
[160] G. Chen, J. Branch, M. Pflug, L. Zhu, and B. Szymanski, "SENSE: A wireless sensor network simulator," in *Advances in Pervasive Computing and Networking*. Heidelberg, Germany: Springer, 2005, pp. 249–267.
[161] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki–a lightweight and flexible operating system for tiny networked sensors," in *Proc. 29th Annu. IEEE Int. Conf. Local Comput. Netw.*, Nov. 2004, pp. 455–462.
[162] A. Dunkels, J. Eriksson, N. Finne, and N. Tsiftes. (2011). *Powertrace: Network-Level Power Profiling for Low-Power Wireless Networks*. [Online]. Available: http://soda.swedish-ict.se/4112/
[163] P. Levis, S. Madden, J. Polastre, and R. Szewczyk, "TinyOS: An operating system for sensor networks," in *Ambient Intelligence*. Heidelberg, Germany: Springer, 2005, pp. 115–148.
[164] S. A. P. Kumar, B. Bhargava, R. Macedo, and G. Mani, "Securing IoT-based cyber-physical human systems against collaborative attacks," in *Proc. IEEE Int. Congr. Internet Things (ICIOT)*, Honolulu, HI, USA, Jun. 2017, pp. 9–16.
[165] A. Dvir, T. Holczer, and L. Buttyan, "VeRA-version number and rank authentication in RPL," in *Proc. IEEE 8th Int. Conf. Mobile Ad-Hoc Sensor Syst.*, Hangzhou, China, Oct. 2011, pp. 709–714.
[166] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 53–58.

[167] T. Afonja. (2017). *Accuracy Paradox*. Accessed: Jun. 28, 2019. [Online]. Available: https://towardsdatascience.com/accuracy-paradox-897a69e2dd9b
[168] I. Butun, B. Kantarci, and M. Erol-Kantarci, "Anomaly detection and privacy preservation in cloud-centric Internet of Things," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, Jun. 2015, pp. 2610–2615.
[169] J. Greensmith, "Securing the Internet of Things with responsive artificial immune systems," in *Proc. Genetic Evol. Comput. Conf. (GECCO)*, 2015, pp. 113–120.
[170] M. Malik and M. Dutta, "Contiki-based mitigation of UDP flooding attacks in the Internet of Things," in *Proc. Int. Conf. Comput., Commun. Autom. (ICCCA)*, May 2017, pp. 1296–1300.
[171] J. Wilander, N. Nikiforakis, Y. Younan, M. Kamkar, and W. Joosen, "RIPE: Runtime intrusion prevention evaluator," in *Proc. 27th Annu. Comput. Secur. Appl. Conf.*, 2011, pp. 41–50.
[172] S. Sicari, A. Rizzardi, D. Miorandi, and A. Coen-Porisini, "REATO: REActing TO denial of service attacks in the Internet of Things," *Comput. Netw.*, vol. 137, pp. 37–48, Jun. 2018.
[173] S. Lu, Z. Li, F. Qin, L. Tan, P. Zhou, and Y. Zhou, "Bugbench: Benchmarks for evaluating bug detection tools," in *Proc. Workshop Eval. Softw. Defect Detection Tools*, vol. 5, 2005, pp. 1–5.
[174] R. Mitchell and I.-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," *ACM Comput. Surv.*, vol. 46, no. 4, p. 55, 2014.
[175] S. Han, M. Xie, H.-H. Chen, and Y. Ling, "Intrusion detection in cyber-physical systems: Techniques and challenges," *IEEE Syst. J.*, vol. 8, no. 4, pp. 1052–1062, Dec. 2014.
[176] M. A. Rassam, M. Maarof, and A. Zainal, "A survey of intrusion detection schemes in wireless sensor networks," *Amer. J. Appl. Sci.*, vol. 9, no. 10, p. 1636, 2012.
[177] Tetcos. (2019). *Netsimv12.1*. [Online]. Available: https://www.tetcos.com/netsim-pro.html
[178] M. Chernyshev, Z. Baig, O. Bello, and S. Zeadally, "Internet of Things (IoT): Research, simulators, and testbeds," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1637–1647, Jun. 2018.
[179] A. Saha and S. Sanyal, "Application layer intrusion detection with combination of explicit-rule-based and machine learning algorithms and deployment in cyber-defence program," 2014, *arXiv:1411.3089*. [Online]. Available: https://arxiv.org/abs/1411.3089
[180] N. D. Lane, S. Bhattacharya, P. Georgiev, C. Forlivesi, L. Jiao, L. Qendro, and F. Kawsar, "DeepX: A software accelerator for low-power deep learning inference on mobile devices," in *Proc. 15th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Apr. 2016, pp. 1–12.
[181] D. Ravi, C. Wong, B. Lo, and G.-Z. Yang, "Deep learning for human activity recognition: A resource efficient implementation on low-power devices," in *Proc. IEEE 13th Int. Conf. Wearable Implant. Body Sensor Netw. (BSN)*, Jun. 2016, pp. 71–76.
[182] D. Ravi, C. Wong, B. Lo, and G.-Z. Yang, "A deep learning approach to on-node sensor data analytics for mobile or wearable devices," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 1, pp. 56–64, Jan. 2017.
[183] ARM. (2018). *ARM Trustzone*. [Online]. Available: https://www.arm.com/products/silicon-ip-security
[184] G. Platform. (2018). *Introduction to Trusted Execution Environments*. [Online]. Available: https://globalplatform.org/wp-content/uploads/2018/05/Introduction-to-Trusted-Execution-Environment-15May2018.pdf
[185] P. C. Security. (2016). *Embedding Security in the Internet of Things*. [Online]. Available: https://www.pfpcybersecurity.com/assets/IoT_Whitepaper.pdf
[186] A. S. Technology. (2009). *Building a Secure System Using Trustzone Technology*. Accessed: Jun. 28, 2019. [Online]. Available: http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD 29-GENC-009492C_trustzone_security_whitepaper.pdf
[187] S. Suhail, R. Hussain, M. Abdellatif, S. R. Pandey, A. Khan, and C. S. Hong, "Provenance-enabled packet path tracing in the RPL-based Internet of Things," *Comput. Netw.*, vol. 173, May 2020, Art. no. 107189.
[188] Kamaldeep, M. Malik, and M. Dutta, "Implementation of single-packet hybrid IP traceback for IPv4 and IPv6 networks," *IET Inf. Secur.*, vol. 12, no. 1, pp. 1–6, Jan. 2018.
[189] A.-S. K. Pathan, *The State Art Intrusion Prevention Detection*. 1st ed. Boca Raton, FL, USA: CRC Press, Jan. 2014.

KAMALDEEP was born in India, in 1990. He received the B.E. degree in computer science and engineering from Chitkara University, in 2012, and the M.E. degree in computer science and engineering from Panjab University, in 2015. He is currently pursuing Ph.D. degree in computer science engineering with the Panjab University's Affiliated Institute, National Institute of Technical Teachers' Training and Research (NITTTR), Chandigarh, India.

During his post-graduation, he developed keen interest in the area of cyber security and forensics and has delivered a number of lectures for teachers of polytechnic and engineering colleges. He is also working on security of the Internet of Things. His research interests include communication and network security in the IoT. His current research activities involve the design and implementation of lightweight cryptographic primitives for the next-generation IoT networks.

MAITREYEE DUTTA was born in Guwahati, India. She received the B.E. degree in electronics and communication engineering from Assam Science and Technology University, and the M.E. degree in electronics and communication engineering and the Ph.D. degree with specialization in image processing from Panjab University.

She is currently a Professor with the Computer Science and Engineering Department, National Institute of Technical Teachers' Training and Research, Chandigarh, India. She has more than 18 years of teaching experience. Her research interests include digital signal processing, advanced computer architecture, data warehousing and mining, image processing, and the Internet of Things. She has more than 100 research publications in reputed journals and conferences. She completed one sponsored research project—Establishment of Cyber Security Lab—funded by the Ministry of IT, Government of India, New Delhi, amounting Rs. 45.65 lac.

JORGE GRANJAL (Member, IEEE) received the Ph.D. degree, in 2014. He is currently an Assistant Professor with the Department of Informatics Engineering, Faculty of Science and Technology, University of Coimbra, Portugal, and also a Researcher of the Laboratory of Communication and Telematics, Centre for Informatics and Systems of the University of Coimbra. His main current research interests are computer networks, network security, and wireless sensor networks. He is also a member of ACM communications groups.