Network Intrusion Detection for IoT Security based on Learning Techniques

Nadia Chaabouni, Mohamed Mosbah, Akka Zemmari, Cyrille Sauvignac and Parvez Faruki

***Abstract*—Pervasive growth of Internet of Things (IoT) is visible across the globe. The 2016 Dyn cyberattack exposed the critical fault-lines among smart networks. Security of Internet of Things (IoT) has become a critical concern. The danger exposed by infested Internet-connected things not only affects the security of IoT, but also threatens the complete Internet eco-system which can possibly exploit the vulnerable Things (smart devices) deployed as botnets. Mirai malware compromised the video surveillance devices and paralyzed Internet via distributed denial of service (DDoS) attacks. In the recent past, security attack vectors have evolved bothways, in terms of complexity and diversity. Hence, to identify and prevent or detect novel attacks, it is important to analyze techniques in IoT context. This survey classifies the IoT security threats and challenges for IoT networks by evaluating existing defense techniques. Our main focus is on Network Intrusion Detection Systems (NIDS); hence, this paper reviews existing NIDS implementation tools and datasets as well as free & open-source network sniffing software. Then, it surveys, analyzes and compares state-of-the-art NIDS proposals in the IoT context in terms of architecture, detection methodologies, validation strategies, treated threats and algorithm deployments. The review deals with both traditional and machine learning (ML) NIDS techniques and discusses future directions.**

In this survey, our focus is on IoT NIDS deployed via Machine Learning since learning algorithms have a good success rate in security and privacy. The survey provides a comprehensive review of NIDSs deploying different aspects of learning techniques for Internet of Things, unlike other top surveys targeting the traditional systems. We believe that, the paper will be useful for academia and industry research, first, to identify IoT threats and challenges, second, to implement their own NIDS and finally to propose new smart techniques in IoT context considering IoT limitations. Moreover, the survey will enable security individuals differentiate IoT NIDS from traditional ones.

I. Introduction

Internet of Things is considered as the third industrial revolution [1]. It is defined as "the interconnection, via the Internet, of computing devices embedded in everyday objects, enabling them to send and receive data" [2]. IoT market is growing at a breathtaking pace, starting with 2 billion objects in the year 2006 to projected 200 billion by 2020 [3], a rise of 200%. IoT sensors/devices often collect and process spatial and temporal information for specific events and environment tackling various challenges [4], [5]. The IoT objects or Things have become smarter, treatment is more intelligent and communications have turned instructive. Therefore, IoT is used in almost all fields: domestic, education, entertainment, energy distribution, finances, healthcare, smart-cities, tourism and even transportation [6]. Consequently, industry, academia and individuals are trying to integrate the flow of fast commercialization with seldom attention to the safety and the security of IoT devices and networks. Such a neglect can possibly endanger the IoT users and in turn disrupt the vibrant ecosystem. For example, Smart-homes can be remotely controlled by cyber-criminals, and Smart vehicles can be hijacked and remotely controlled to create panic among citizens.

The danger exposed by these Internet-connected Things not only affect the security of IoT systems, but also the complete eco-system including web-sites, applications, social networks and servers, via controlled smart device as robot networks (botnet). In other words, compromising a single component and/or communication channels in IoT-based systems can paralyze the part or complete Internet network. In 2016, the Dyn cyberattack harvested connected devices installed within smart-homes and conscripted them into "botnets" (also referred to as a "zombie army") via a malware called `Mirai`. In addition to IoT systems vulnerabilities, attack vectors are evolving in terms of complexity and diversity. Consequently, more attention should be paid to the analysis of these attacks, their detection as well as the infection prevention and recovery of systems after the attacks.

A. Scope of the Survey

Since security of pervasive IoT systems is critical, it is important to identify IoT threats and specify existing defense strategies. This survey starts with IoT threats classification to have a better vision for strategic investigations. For that, we propose a binary classification with: i) IoT layers; and ii) encountered challenges while developing the IoT systems. We believe that IoT networks are different from Wireless Sensors Networks (WSN) and Cyber Physical Systems (CPS) [7] due to heterogeneous composition of layers in terms of protocols, standards and technologies. Furthermore, variety of challenges encountered during the implementation of various use-cases mentioned in [8] have different context compared to WSN networks.

Traditional defense mechanisms for known attacks have varied use and may be efficient in specific situations; however,

N. Chaabouni is with LaBRI, Univ. Bordeaux, Bordeaux INP, CNRS France and Atos Innovation Aquitaine Lab, France.(e-mail: chaabouni.nadia14@gmail.com).

M. Mosbah and A. Zemmari are with LaBRI - Bordeaux Laboratory of Research in Computer Science, Univ. Bordeaux, Bordeaux INP and CNRS UMR 5800, F33405 France (email: mosbah@u-bordeaux.fr, zemmari@u-bordeaux.fr).

C. Sauvignac is with Atos Innovation Aquitaine Lab (email: cyrille.sauvignac@atos.net).

P. Faruki is with Computer Engineering Department, MNIT Jaipur, India (e-mail: parvezfaruki.kg@gmail.com)

they may not be completely secure. Despite the availability of traditional security with encryption, authentication, access control or data confidentiality, IoT networks still have been subject to network attacks necessitating a second line of defense [9], [10]. In such situations, the importance of Intrusion Detection Systems (IDSs) for IoT is relevant. One of the popular strategy deployed among IoT systems is IDSs or Network Intrusion Detection Systems (NIDSs) for connected smart Things. NIDSs have been subject to scrutiny to achieve secure traditional computer science systems since the 1980s [11]. Thus, NIDS is a mature scientific field. Unfortunately, traditional NIDS techniques may be less efficient and/or inadequate for IoT systems due to characteristic changes like constrained resources, limited power, heterogeneity and connectivity [9], [10]. Traditional systems have usually master nodes which are powerful in terms of computation resource and storage/memory space. These nodes monitor inbound and outbound flows with no major resource or network bandwidth constraints. However, IoT systems are distributed and composed of a large number of devices whose computing capacity, storage/memory space and battery life are mainly limited in resources. IoT is also limited by its network bandwidth capacity. Moreover, IoT allows interaction between the virtual and physical environment which is unpredictable. Every node has an IP address to guarantee its communication with Internet. This causes trust problems and particular vulnerabilities. In addition to these limitations, IoT is based on heterogeneity in terms of communication protocols and co-existing technologies. The protocols and technologies are either not employed in traditional networks such as IEEE 802.15.4, 6LoWPAN[1] and CoAP; or at least not at the same time within a single system [8]. Finally, IoT environment generate voluminous critical data that must be protected. Consequently, NIDSs are more challenging and restrictive in IoT networks compared to NIDSs in traditional computing systems.

Many IoT NIDS have been developed using attacks rules/signatures or normal behavior specification. Unfortunately, these NIDS have; i) high false positive and/or false negative attack recognitions (false alarms); ii) inability to detect unknown/zero day attacks. Hence, researchers explored artificial intelligence (AI) and machine learning (ML) with an emphasis on deep learning (DL) algorithms to improve systems security [13], [14], [15]. In fact, learning techniques have a significant impact in fraud detection, image recognition and text classification. The effectiveness of machine learning has encouraged the researchers to deploy learning algorithms among IDS to improve detection of cyber attacks, anomaly detection and identify abnormal behaviors among the IoTs. Therefore, this paper surveys and evaluates notable machine learning contributions for IoT NIDSs. In the recent past, academia and industry have shifted their focus towards developing ML based NIDSs. They accomplish interesting results; from 86.53% [16] to over 99% [17] in detection accuracy and a reduction in false positive (FP) from about 4% [18] to 0.01% [19].

[1]Kasinathan et al. in [12] define 6LoWPAN as "a standard protocol designed by IETF as an adaptation layer for low-power lossy networks enabling low-power devices (LLN) to communicate with the Internet".

To summarize, there are three important areas that the survey is targeting (Fig. 1): i) IoT; ii) security mechanisms; and iii) machine learning techniques. Our focus will be on the intersection of the above three domains as highlighted in Fig. 1: i) the intersection between IoT and security (IDS for IoT); and ii) the intersection between the three domain (IoT NIDS that are deployed via learning techniques).

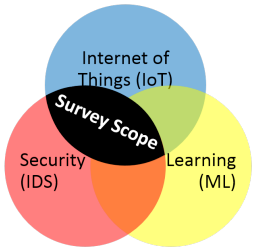

Fig. 1: Survey Scope

B. State-of-art Surveys

There are multiple surveys that treat the two other intersections. For the first intersection between IoT and learning domain, [20], [21], [22] and [23] have significant contributions. Chen et al. [20] review data mining (DM) for IoT in knowledge, technique and application points of view. The authors study big data algorithms and challenges when deployed in IoT environment. Tsai et al. [21] survey both, features of data for IoT and features for data mining for IoT with a discussion about changes, potentials, open issues, and future trends. However, Cui et al. [22] provide an overview of the application of machine learning in IoT domain. This survey is very recent and concentrates on progresses in machine learning techniques for IoT applications. Finally, Mahdavinejad et al. [23] present a taxonomy of machine learning algorithms while discussing how they can be applied to the data in order to extract higher level information. Furthermore, the authors explain the potential and challenges of machine learning for IoT data analytics.

Surveys about IDS based on learning techniques (not specially for IoT) such as [24], [25], [13], [15], [26] and [27] have performed comprehensive evaluation for traditional systems. Agrawal et al [24] review various data mining techniques for anomaly detection. Buczak et al. [25] discuss machine learning (ML) and data mining (DM) methods for cyber analytics in support of intrusion detection. Both surveys refer and summarize complexity and challenges for ML/DM cyber-security. However, Fadlullah et al. [13] concentrate on deep learning (DL) for network traffic control systems. Hodo et al. [15], proposed shallow and deep networks taxonomy and surveyed intrusion detection systems. Moreover, Wang and Jones survey [26] reviewed DM, ML, DL and Big Data works with an emphasis on subsequent evaluation criteria such as stream data characteristics, stream processing systems, feature dimension reduction and data reduction. Finally, Mishra et al. [27] analyze and compare limitations of machine learning techniques as well as constraints for deployment in intrusion

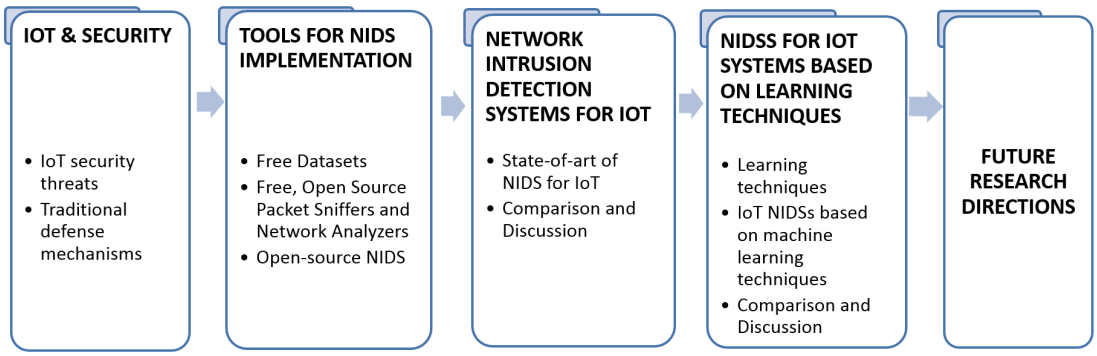

Fig. 2: Survey Workflow

detection. The key factor which differentiates Mishra et al.'s work from existing surveys is the evaluation that no particular intrusion detection technique can help detecting all types of attacks.

The final list of existing surveys target intersection between IDS and IoT. From recent state-of-art, two Surveys about IoT intrusion detection are identified: [28] and [8]. Zarpelo et al. [28] present an overview IoT specific IDS, and introduce a taxonomy to classify them. Moreover, they propose a detailed comparison between the different IDS for IoT with parameters like placement strategy, detection method and validation strategy. However, Ben Khelifa et al. [8] concentrate on advancements in intrusion detection practices in IoT. They review recent state-of art with a special focus on IoT architecture. The authors finish their survey with future directions for IoT NIDS. Ben Khelifa et al. survey is more detailed clear critical review. However, none of the above discussed surveys concentrates on the use of machine learning for IoT NIDS. To the best of our knowledge, our proposal is the first comprehensive review that evaluates deployment strategies, and compares the effectiveness of machine learning powered IDS for the security of IoT systems.

C. Paper Selection Criteria and Survey Workflow

The discussed papers in Section IV and V are based on the following criteria:

- The papers deal with intrusion detection in IoT.
- NIDSs target the IoT systems in general (e.g. not just WSN networks) with their heterogeneity, mobility and all the IoT specific challenges.
- Authors present their NIDS architecture in details.
- Discussions and comparisons are about traditional NIDs in Section IV. However, they concern NIDs based on learning techniques in Section V.
- The state-of-art articles are mainly from indexed and top IEEE, ACM, Elsevier and Springer journals, and top conference venues published between 2013 until October 2018.

Rest of the Survey is structured in four blocks as illustrated in Fig. 2. In Section II, IoT security threat categories are classified and traditional defense techniques are presented with a focus on IDSs types. Section III lists and discusses available tools that can be used to develop NIDS; free datasets, free and open source network sniffers and finally open source NIDSs. These tools can be also used to test and evaluate the performance of NIDS. Section IV discusses IoT powered NIDS, their architecture, deployment and implications for the heterogeneous systems. In Section V, learning techniques via machine learning classifiers are introduced. Then, NIDSs for IoT systems deployed via learning techniques are reviewed, compared and evaluated. We comprehensively discuss existing state-of-art, compare and evaluate performance of machine learning based IoT systems deployed to secure the networks. Finally, the survey is concluded with a summary and a list of possible future directions in Section VI.

II. IoT & Security

Diversity and heterogeneity makes IoT systems security more crucial. IoT systems differs from traditional systems security due to following reasons:

- IoT systems are constrained in terms of computational capability, memory capacity, battery life and network bandwidth. Hence, it is not possible to deploy existing traditional security solutions which are often resource intensive.
- IoT systems are heavily distributed and heterogeneous systems. Thus, centralized traditional solution may not be suitable. Moreover, the distributed aspect of IoT add more difficulties and constraints in their protection.
- IoT systems are deployed in a physical environment which is unpredictable. Thus, physical attacks have joined the list of traditional security threats.
- IoT systems are connected to Internet since each device has access with its IP address. Hence, there is one more panel of threats related to Internet.
- IoT systems are composed of a large number of constrained objects that generate huge amount of data. So it is easy to flood and attack these small devices on the one side, and the limited bandwidth of the networks on the other side.
- IoT systems cover a large number of heterogeneous protocols and technologies in the same system. Hence, the proposed IoT security solution must take into consideration the large panel of these protocols and technologies in the same proposal

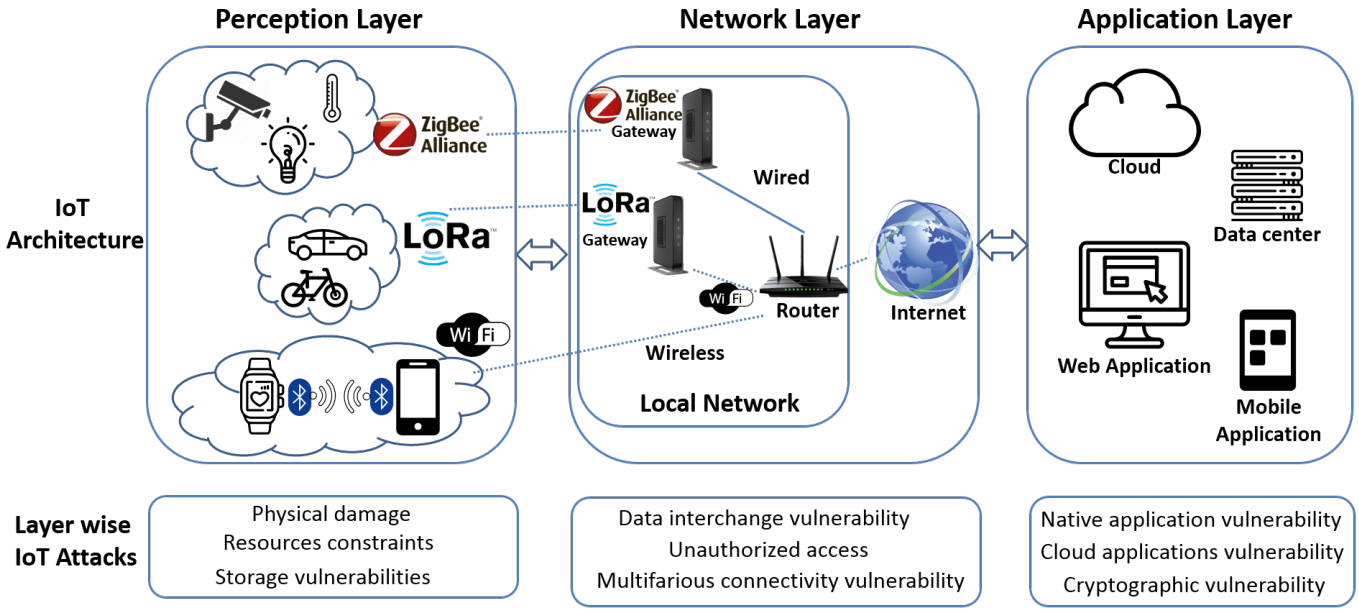

Fig. 3: IoT architecture & layer wise attacks

Consequently, IoT systems threats classification is discussed; then, traditional defense mechanisms employed against such threats are introduced.

A. Classification of IoT Threats

Since the IoT systems are varied and are facing multiple challenges, IoT threats can be classified into two types. The first type is about classification depending on the layers of the IoT systems' architecture, while the second one deals with IoT threats categorization based on their design challenges.

1) IoT threats classification by layers: IoT systems relate physical environment to the virtual one. A standard representation of IoT architecture is shown in Fig. 3. IoT consists of three main layers [29] which are perception / physical layer, network / transport layer and application layer.

First, the perception layer is the hardware layer. It is composed of the different sensors and actuators that send and receive data using different communication standards such as Bluetooth, RFID and 6LowPAN. Second, the network layer is the one which ensures the effective routing/transmission of data/information. It uses communication protocols like WiFi, 3G, GSM, IPv6, etc. Third, the application layer also called the software layer is the top layer that provides systems with the business logic and offers the user interfaces (UI) to the end users (traffic monitoring, smart classroom, etc.).

Each layer may represent multiple vulnerabilities as illustrated in Fig. 3 [30], [31]. Since devices are placed at different physical locations, they may be exposed to environmental hazards like abnormal rain/snow/wind or malicious attacks or unintentional damage. Also, stored data may be stolen via physical access. Sensors are tiny Things so, they suffer from resource constraint issues (computational resources, memory or energy, etc.). While data/commands are exchanged (network layer), they can face different network vulnerabilities such as data interchange vulnerabilities (data transfer can be shut down because of network floods or malicious gateway access), unauthorized access (impersonation attack, communication interception, password guessing attacks, etc.) and multifarious connectivity vulnerabilities (data integrity violation, bad Quality-Of-Service (QoS), etc.). Moreover, the application layer is mainly exposed to software problems such as account enumeration, insecure account credentials and lack of account suspension after a limited number of password guessing. Cloud applications [32], [33] can be attacked by viruses, trojan horses, worms, etc.. Since IoT is based on low computational capability devices, transport encryption is sometimes neglected or used in a weak version. Therefore, communications are easily traceable and easily discovered (Cipher text-only attack, Man In the Middle).

2) IoT threats classification by challenges: To understand IoT security attacks first, we introduce some IoT attack technical terms, then we present IoT challenges-based classification.

a) Technical terms of the attacks: First, **spoofing** [34], [35] or impersonation attack sneaks authentication credentials to gain unauthorized service access. Credentials can be stolen directly from a device, via eavesdropping the communication channel or by phishing. Spoofing can be categorized into: i) IP address spoofing; ii) ARP spoofing; and iii) DNS server spoofing. IP address spoofing refers to the falsification of content in the source IP header to mask sender's identity or to launch a reflected distributed denial of service (DDoS) attack. ARP spoofing attacks typically address resolution protocol (ARP). The spoofing attack resolves IP addresses to MAC (Media Access Control) addresses. When an attacker sends spoofed ARP messages across the Local Area Network (LAN), attacker's MAC address will be linked with the IP address of a legitimate member of the network. Consequently, malicious

parties can steal data, modify data in-transit or even stop traffic on a LAN. DNS server spoofing modifies a DNS server (a system that associates to each domain name an IP address) reroutes a specific domain name to unauthorized IP address of the infected server.

Second, **routing attacks** [36] target routing protocols where the exchanged routing information are spoofed, altered or replayed to generate fictitious routing behaviors (i.e., false network traffic attraction). **Sinkhole attack** [37] concerns a malicious node attracting huge traffic by presenting imaginary path as an optimal routing path. Concerning **selective forwarding attack** [38], is a data forwarding misbehavior where an attacker selectively forwards malicious packets while discarding genuine and important packets. Furthermore, **blackhole attack** [37] aims to disrupt normal data flow within a network. Initially, the attack fudges one or more faulty nodes as the best route(s); then, starts to drop data packets routed through the faulty path. On the other hand, **wormhole attack** [37] needs at least two faulty nodes connected via wired or wireless link. These malicious nodes tunnel the packets faster than normal track. Moreover, **replay attack** [34] considers the re-transmission or the delaying of valid data to gain unauthorized access within already established session.

Third, **tampering** attack [34] is classified as: i) Device tampering; and ii) Data tampering. The device tampering can be easily performed especially when an IoT device spends most of the time unattended. It can be easily stolen without being noticed and so used maliciously. The device can be stolen as hardware or just as software. The data tampering involves malicious modification of data for example data stored in databases or data transiting between two devices.

Fourth, **repudiation** [39] is about devices doing a malicious action and then denying performing it. It is the case when a device sends a virus on the network without leaving any trace to identify it.

Fifth, **information disclosure** [34] deals with unauthorized information access. An attacker achieves the same by attaching snooping devices, by snooping the network channel or by getting physical access to a device; e.g Probe [40], [41] is when attackers try to gather information about a target node and its vulnerabilities by scanning connections (Port scanning, etc.). With information disclosure comes sensitive information leakage such as side channel attack [42].

Sixth threat is **DDoS** [29], the Distributed Denial of Service attack performed by multiple compromised nodes together from different geographic locations. Besides, DoS attack implicates a malicious attacker that attempt to consume network resources, target CPU time and/or bandwidth of legitimate users by flooding the system with rogue and amplified traffic. To conduct an efficient DDoS attack, botnets are used. They are networks of infected/controlled internet-connected devices. As mentioned in [43], DoS attacks are the most frequent attacks especially in IoT / Fog networks related to social IoT such as smart cities, etc. DDoS attacks can be categorized into following types [44], [45]:

1) Flooding attacks are based on bombarding a victim's system with a large number of packets, mainly UDP or ICMP packets [46], which causes compromise of network bandwidth. Flooding attack can be easily launched using botnets.
2) Amplification attacks can be established by exploiting reflection mechanism and spoofing the IP sources. Attackers send packets to reflector servers with a source IP address set to their victim's IP therefore indirectly overwhelming the victim with the response packets. To resume, hackers exploit vulnerabilities in different protocols to turn small queries into huge number of requests to slowdown and/or crash the victim's server(s). For examples, there are smurf, fraggle attacks [45] and DNS, SSDP amplification [47], [48] distributed attacks.
3) Protocol exploit attacks are built on malicious exploit of different protocols. As examples, there are SYN flood [47], TCP reset [47] and water torture attack [49], [50].
4) Malformed attacks are based on malformed network packets such as using same IP address for source and destination addresses [45], [47].
5) Logical/software attacks are attacks related to application protocols. For example Ping of Death [29] where an attacker sends simple fragmented ICMP ECHO request packet, larger than maximum IP packet size so that the victim fails to reassemble it. In Teardrop [46] attack, the adversary sends two fragments that do not reassemble by the offset value of the packet.

Seventh, **elevation of privilege** [39], [34] concerns obtaining or elevating privileges to access a device/service while not having a legal right. Such an attack can lead to dangerous situation especially when the attacker becomes a trusted part system. User-To-Root (U2R) and Remote-to-Local (R2L) [51], [40] are two examples of elevation of privilege attack. U2R is about gaining root privileges (superuser) on a node when the attacker initially has only a normal user account. However, R2L occurs when an attacker does not have an account on the victim node, hence exploits vulnerabilities to gain local access as a user via password guessing or breaking.

Eighth, **MITM** [34], [35], Man-In-The-Middle attack which represents interception of communication between two systems by an adversary to eavesdrop a conversation. MITM attacks are classified as ARP Cache poisoning, DNS spoofing, session hijacking, ICMP redirect, port stealing, etc.

Ninth, **user privacy** [34], [52] is like information disclosure. Besides, a hacker does not necessarily need to have access to unauthorized information to learn about a user. This can be done by analyzing metadata and traffic.

Tenth, **cloning Nodes** [53], [54], [55] concerns reintroducing a clone of a node in the network or a component in a system after capturing the credentials and the characteristics of the original one. Such an attack enables the malicious user to control the system, insert false information, disable functions, etc. Once an object is under the control of the attacker without the knowledge of its owner (botnet), the entire network can be infected.

b) IoT threats classification by design challenges: Because of different challenges related to IoT systems' design, developers as well as industries should pay attention

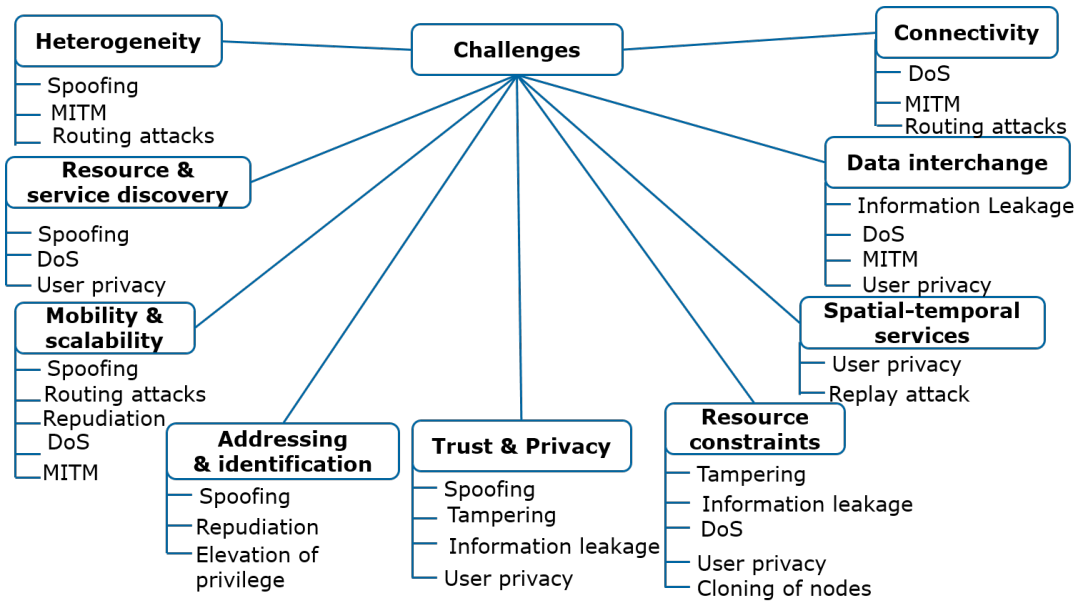

Fig. 4: IoT threats classification by design challenges

to many potential threats. Many research papers surveyed IoT security challenges and research opportunities such as Zhang et al. [56]; they detail IoT security challenges such as Object identification, Authentication and IoT privacy, etc. in IoT networks. A classification based on IoT systems' design challenges is presented in Fig. 4 and detailed below.

- Heterogeneity and Interoperability
  The backend IoT solutions consider the use of sensors, actuators and gateways provided by different vendors and may have different versions. To do so, the use of a dispositive managing interoperability between heterogeneous devices is needed. Such a component can be bombarded with fake requests that can lead to DoS attacks. In such a heterogeneous environment, spoofing, routing attacks as well as MITM are more likely to occur compared to the homogeneous systems. It is easier for a malicious node to impersonate a genuine Thing, gain unauthorized access to data and/or relay communication between two nodes message injection. As we can see from Fig. 3, IoT might be considered as the term which references a world of large variety of heterogeneous protocols and standards [57]. Their consideration makes IoT security solution more and more complex. Al-Fuqaha et al. provide a good survey about these technologies in [58].
- Connectivity
  In IoT, connectivity between different components of the system is required whether physical or the one in terms of services. For the first case, data from peripheral devices (sensors for example) have to be connected to an IP network with bridging devices which may be the cause for routing attacks as well as MITM attacks. For the connectivity in terms of services, changes in the availability of services should be notified to the respective devices so that the latter do not flood the system unknowingly with repetitive and non-available requests. Such a flood can lead to a DoS attack. Moreover, the QoS in IoT networks can be crucial specially in emergency situations. Thus, robust packet routing and a good QoS in data delivery should be ensured even in highly dynamic topologies [59].
- Mobility and Scalability
  Devices of IoT systems can be in continuous mobility in the field area; hence, they can change bridges they are connected to. This often causes disruption of discontinuity and/or connections to unauthorized services. Attacks like repudiation, MITM, DoS, sinkhole and wormhole become potentially possible. To mitigate such risks, security solutions not only consider mobile devices, but also network components such as the switches and the routers [60].
- Addressing and Identification
  Field devices in IoT applications use usually low power radios for short distance connection (less than 1 kilometer). For that, coordinator nodes allocate local addresses that do not follow a common standard, to peer devices. Consequently, these addresses remain hidden behind the FAN gateway/bridge; hence, malicious behaviors become untraceable. As a result, isolation of malicious node(s) and detection of spoofing and repudiation attacks is difficult. Further, the node can attempt to access unauthorized privileges without being screened from the outside network (elevation of privilege).
- Spatio-temporal services
  Events in IoT can be characterized by the amplitude of spatio-temporal impulse. As a result, data from IoT devices of same systems should have reasonable temporal behavior and spatial geolocation. However, these spatio-temporal tags must be protected from malicious users to avoid replay attacks. Also, the user's location data must

not be revealed to unauthorized users.

- Resource constraints
  Most peripherals IoT devices are tiny, which means that they are resource constrained in terms of computing power, onboard memory, network bandwidth, and energy availability. Tampering, information leakage and node cloning are possible attacks since the smart devices and sensors are resource constrained. The constrained resources limit the deployment of cryptographic solutions hence, lightweight solutions are foremost concern. For example, in [61], authors overcome such limitations and propose a novel error correction and detection technique entitled "Low Complexity Parity Check (LCPC)", to improve the quality of futuristic IoT networks.
- Data Interchange
  Before data interchange begins, it must be encrypted at the source IoT nodes. The encryption mechanisms, depends on the type of hardware, its computational capability and storage capacity. Inappropriate selection leads to security vulnerabilities such as information leakage (i.e., keys are being shared between multiple devices when encrypted packets are decrypted and repacked at multiple points in the communication chain). Additionally, nodes that encrypt data can be attacked via denial of service or resource exhaustion attacks. For this reason, end to end encryption is desirable.
- Resource and service discovery
  In IoT systems, mechanisms of resource and service discovery should be deployed to enable autonomy and self-discovery of the devices. These mechanisms should be protected with two way authentication to avoid spoofing or restrict the malware component from flooding the system with feigned requests to thwart DoS attacks.
- Trust and privacy
  IoT smart sensor devices manage private / sensitive user informations (e.g. user habits, patients data, civil protection data etc.); hence, confidentiality and data protection is extremely important. In fact, trust and privacy [62], [63] are fundamental issues for IoT based networks. Users, Things and devices are required to authenticate via reliable services for mitigating spoofing, tampering and information leakage attacks. Trust and privacy are getting more attention with smart phones e.g. Android OS [64], [65] and [66].

B. Traditional defense mechanisms

After detailing and classifying IoT threats, in the following we discuss attack mitigation techniques which protects existing IoT systems and networks. Over time, conventional IT security solutions have covered servers, networks and cloud storage. Most of these solutions can be deployed for security of IoT systems. Defense mechanisms can be separate, or combined depending on the treated threats [67]. In this Section, traditional mechanisms that can be used to protect IoT are described.

First, **filter packets** [68], with firewalls and proxies for example, represents an important defense against IP spoofing attacks (and consequently DDoS attacks). Two types of filtering are possible: i) ingress filtering; and ii) egress filtering. Ingress filtering on incoming packets is about blocking packets from outside the network with a source address inside the network to protect against outside spoofing attacks. However, egress filtering on outgoing packets is about blocking packets within the network with a source address that is not inside to prevent an internal hacker from attacking external machines.

Second, **adopt encryption** with cryptographic protocols, data storage encryption or virtual private networks (VPNs). Using cryptographic network protocols (i.e., Transport Layer Security (TLS), Secure Shell (SSH), HTTP Secure (HTTPS), etc.) leads to the encryption of data/code/updates before sending and authenticating them. The defense is based on digital signatures/certificates (pair of public and private keys) to ensure, in one hand, that data/code/update was sent by the legitimate device/service and never modified. On the other hand, it guarantees that data/code/updates are encrypted and cannot be read or used by unauthorized individual. Cryptographic network protocols can be used to protect things against IP spoofing, tampering, repudiation, MITM, user privacy compromising attacks and node cloning. Moreover, encrypting data storage helps prevent information disclosure and maintains user privacy. Concerning VPN (Virtual Private Network), it is a secure communication tunnel between two or more devices. It encrypts the communication by creating a virtual private link over the existing unsecure network. Encryption is a good solution to preserve confidentiality and privacy. However, IoT networks are vulnerable since the resource limits the devices. Therefore, the use of light cryptographic solutions proposal from Al-Turjman et al. [69] is an interesting approach. They propose a confidential cloud assisted WSN based framework maintaining confidentiality, integrity and access privileges (CIA). The proposed agile framework ensures integrity of collected sensor data with elliptic curve cryptography.

Third, **employ robust password authentication schemes**. Moreover, limit data access by assigning the resources with appropriate privileges. The use of One-Time Password (OTP) can be an interesting solution. Spoofing, tampering, information disclosure, elevation of privileges and MITM can be avoided by the above mechanisms.For IoT networks, authentication strategies need to be lightweight such as in Al-Turjman et al. solutions. In [70] authors propose a light weight framework to strengthen the security of IoT networks. They introduce a cloud supported mobile-sink authentication, an elliptic-curve based seamless secure authentication and key agreement (S-SAKA). However in [71], authors propose a "Hash" and "Global Assertion value" based authentication scheme for the evolving 5G technology. Their proposal considers context-sensitive seamless identity provisioning (CSIP) framework for futuristic Industrial Internet of Things (IIoT).

Fourth, **audit and log activities** on web servers, database

servers, and application servers. Due to these traces, outliers can be detected. More specifically, log key events such as transaction, login/logout, access to file system or failed resource access attempt(s) can detect anomalous behavior. A good practice to protect these files is to back up them, regularly analyze them for detection of suspicious activity and relocate system log files from their default locations. Further, secure the log files by using restricted ACLs (Access Control List: a list of permissions attached to an object) and ecrypt the transaction log. These techniques prevent IoT systems from repudiation and privilege elevation attacks.

Fifth, **detect intrusions using IDS (Intrusion Detection System)**. An **IDS** [72] is a combination of software and hardware which monitors network or systems to identify malicious activities and gives immediate alerts. They have been adopted [73] since 1970 [74]. IDSs are generally categorized according to i) deployment; and ii) detection methodology.

IDS deployment is categorized as i) HIDSs; and ii) NIDSs. Host-based Intrusion Detection Systems (HIDSs) are installed on a host machine (i.e., a device or a Thing). They monitor and analyze activities related to system application files and operation system. HIDSs are preferred against insider intrusion deterrence and prevention. Network-based Intrusion Detection Systems capture and analyze packet flow in the network. In other words, they are scanning sniffed packets. NIDSs are strong against external intrusion attacks. Since our interest is towards security of resource constrained IoT systems, the rest of the paper will focus on NIDSs solutions.

In the following, we discuss scenario after the intrusion has happened. A good detection system is the one which identifies the compromised situation and minimizes the loss by quickly identifying the attack(s).

There are a variety of IDSs. In [12], detection methodologies are classified as i) misuse detection; ii) anomaly detection, iii) specification detection; and iv) hybrid detection.

- Misuse detection or signature detection (knowledge based) is a set of predefined rules (such as bytes sequence in network traffic or known malicious instructions sequence used by a malware) that are loaded and matched with events. When a suspicious event is detected, an alert is triggered. This type of IDS is efficient for known attacks; unfortunately it cannot detect zero-day [41] / unknown / unseen attacks [75] due to lack of signatures. Cyber security solutions prefer signature based detection as it is simple to implement and effective for identifying known attacks (high detection rate with low false alarm rate).
- Anomaly detection (behavior based) compares a normal recorded behavior with current input. Initially, normal network and system behavior are modeled. In case of deviation from normal behavior, the detector considers it an attack. Anomaly is identified with statistical data analysis, mining and algorithmic learning approaches. Anomaly detector is successful in preventing unknown attacks. However, they tend to generate high false positive rate since previously unseen (yet legitimate) behaviors may be categorized anomalous. Another advantage is that the normal profile activities are customized for every system, every application and every network, which makes things difficult for the attacker. It is difficult to know exactly which activities can be undetected.
- Specification detection has the same logic as anomaly detection. It defines anomaly as deviation from normal behavior. This approach is based on manually developed input specifications to capture legitimate (rather than those previously seen) behavior and its deviations. However, specifications require the user to give input. This method reduces high false alarm rate as compared to anomaly detectors.
- Hybrid detection is a combination of previous methods, especially signature and anomaly based detection. Hybrid detector improves accuracy by reducing false positive events. Most of the existing anomaly detection systems are in reality hybrid one. They start with an anomaly detection, then try to relate it with the correspond signature.

Sixth, **prevent intrusions with IPS (Intrusion Prevention System)**. An **IPS** is an IDS which respond to a potential threat by attempting to prevent it from succeeding. An IPS responds immediately and stop malicious traffic to pass before it responds by either dropping sessions, resetting sessions, blocking packets, or proxying traffic. However, an IDS responds after detecting passed attacks. There are many types of IPS [72] mainly in-line detection, layer seven switches, deceptive systems, application firewalls, and hybrid switches. To get more details about IPS types, please refer to Patil et al. paper [72].

The above presented mechanisms can be used to protect IoT systems. Some of them like encryption and authentication are insufficient [9] to protect IoT, therefore; IDS are necessary and are more suitable for this case of systems. They can be considered as the last line of defense when other tools are broken. Another advantage of IDS is that they are variated and adaptable depending on needs. They can be doted with learning logic such as machine learning and artificial intelligence techniques in addition to other advanced technologies. This subject will be discussed in the next section.

From the different types and categories of IDSs, this survey concentrates on Anomaly and Hybrid Network IDSs (ANIDSs - HNIDSs) for IoT systems. This choice was made due to the power and the ability of anomaly and hybrid IDSs to detect unknown attacks. Moreover, the paper focuses on the network deployment since it offers more freedom in solution development unlike host deployments in IoT which necessitate low-power consumption and are resource constrained. IoT systems are heterogeneous and too big in term of number of devices. Therefore, having a single/multiple system(s) monitoring the entire network rather than analyzing each host separately (i.e.; the approach of HIDS is per-device security) is more suitable for the case of IoT networks security. After all, IoT is by

definition about the inter-connection of heterogeneous Things (devices).
The next Section presents different free, and open-source tools to develop and implement novel NIDS.

III. Tools for NIDS Implementation

NIDSs analyze network traffic to detect malicious behaviors. To build a NIDS, these are the needed basic steps [76] :

1) Collect the traffic data from the network.
2) Analyze the collected data.
3) Identify relevant security events.
4) Detect and report malicious events.

To perform these steps, researchers have two choices; use of the existing tools to facilitate implementation of their own NIDS; or the develop novel detection strategy. About existing tools, a person can choose between i) free datasets in an off-line mode (since it is difficult to test proposals on real networks and that datasets are a good solution for benchmarking); ii) free open source network sniffers to capture his own network traffic data; or iii) free open source NIDS that can be used and adapted for desired goals. To help researchers get a clear about available tools, we start with free datasets for NIDS; then, we discuss free and open source network sniffers and NIDS. These three types of tools are correlated. The network sniffers are used to collect network traffic data that will be stored in dataset. The Input is unlabeled; therefore, NIDS are needed to differentiate the instance as an attack or normal behavior. NIDS are generally larger than network sniffers. They use network sniffers to capture data which is subsequently used to differentiate attacks from normal behaviors.

A. Free Datasets

Free datasets can be used for NIDS implementation and/or validation. Unfortunately, there are no datasets created specifically for IoT networks. Hence, two strategies are possible: download an available dataset targeting traditional systems or deploy sniffing software in networks.
The most widely adopted datasets for NIDS are KDDCUP99 (KDD99), and NSL-KDD which is an improved version of KDD99. UNSW-NB15 [77] seems to be an interesting dataset for NIDS. Public datasets like PREDICT, CAIDA, DEFCON, ADFA IDS, KYOTO, ISCX 2012 and ICS attack datasets are available for evaluation and testing. The latest are either composed of unlabeled data, or are inaccessible from some countries or are specific domain data. Moreover, datasets suffer from i) privacy issues; ii) the heavy inputs anonymization; and iii) the non reflection of current security attacks.

- KDD99 [51]: is a dataset used for detection of “bad” connections from the “good” ones at the Third International Knowledge Discovery and Data Mining Tools Competition [78] for building the robust NIDS. The dataset is the feature extracted version of DARPA dataset (DARPA is a base raw dataset). KDD99 contains records from military network environment with injected attacks which can be categorized into: i) Denial of Service; ii) Remote to User; iii) User to Root; and iv) Probing. KDD99 is based on 41 features for each connection along with the class label using Bro-IDS tool (presented lately). The features are grouped into 4 types [51]:
  - 1-9: Basic features of individual TCP connections.
  - 10-22: Content features within a connection suggested by domain knowledge.
  - 23-31: Traffic features computed using a two-second time window.
  - 32-42: Host features are designed to assess attacks which last for more than two seconds.

  KDD99 is popular and is the most used by the researchers for experimental analysis. Different works [79], [80], [81], [40], [82], [83] were established to reduce the number of features by selecting the most relevant ones from the initial 41 features. However, many researches have reported disadvantages of KDD99 like [84], [85]. Some of the important ones are [73], [77], [86]:
  - The probability distribution of the testing and training sets are different. In other words, KDD99 suffers from unbalanced classification methods. Because of added new attack records to the testing set, balance between the types of attacks and normal traffic is not maintained anymore.
  - The dataset is out of date.
  - There is evidence of simulation artifacts that could result in over-estimations of anomaly detection performances.
- NSL-KDD [77], [87]: is the upgraded version of KDD99 to overcome its limitations. First, duplicated records in the training and test sets are removed. Second, there are a variety of records selected from the original KDD99 to achieve reliable results from classifier systems. Third, the problem of unbalanced probability distribution is eliminated. The major problem that persists in this dataset is the lack of modern low foot print attack scenarios.
- UNSW-NB15 [77]: was created in 2015 by the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) with IXIA PerfectStorm tool. Its goal is to generate hybrid real modern normal activities and synthetic contemporary attack behaviors. It is about two million and 540,044 records which are stored in four csv files. Those records are generated from 100GB captured raw traffic with tcpdump tool [91] (in pcap files). This dataset has nine types of attacks, namely, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. Fig. 5 illustrates steps to generate UNSW-NB15 dataset.

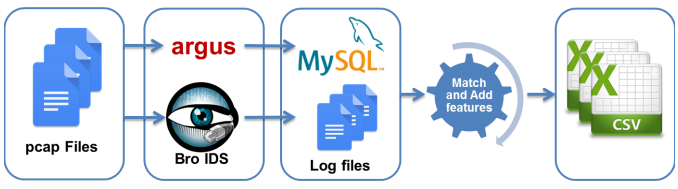

Fig. 5: How to generate UNSW-NB15 dataset [77]

- Sivanathan et al. IoT dataset [88], [92]: addresses IoT device classification based on network traffic characteristics. Authors instrument a smart environment for 28 IoT devices like spanning cameras, lights, plugs, motion sensors, appliances and health-monitors. Furthermore, they synthesized network traffic traces from their infrastructure for a period of six months released for research community. Sivanathan et al. present valuable insights about the network traffic patterns via statistical analysis using attributes such as activity cycles, port numbers, signaling patterns and cipher suites.
- CICIDS database [89]: is one of the recent Intrusion Detection/Intrusion prevention database released by Canadian Institute for Cyber-security, University of New Brunswick to reflect latest threats resembling the real-world data. It was built on the abstract behavior of 25 users based on the HTTP, HTTPS, FTP, SSH, and email protocols. The dataset is analyzed with CICFlowMeter [93] with labeled flows based on timestamp, initial and final IP, ports, protocols and attacks. To generate the realistic traffic, authors proposed B-Profile [94] approach to outline the behavior on HTTP, HTTPS, FTP, SSH and e-mail protocols. Authors implemented Brute force FTP, SSH Heartbleed and DDos attacks while capturing the data. The evaluation framework [95] identified eleven important features necessary to build a reliable benchmark dataset, unlike the existing traditional IDS datasets.
- CSE-CIC-IDS2018 [90] database: is a unique IDS dataset which has evolved to replace the existing suboptimal datasets that limits IDS/NIDS experimental evaluations. To overcome the use of static and one-time datasets, CSE-CIC-IDS2018 is an anomaly based dynamically generated dataset consisting intrusion in network traffic. Authors included seven attack scenarios including i) Brute-force; ii) Heartbleed; iii) Botnet; iv) DoS; v) DDoS; vi) Web attacks; and vii) Local network infiltration attacks. Attack infrastructure has 50 nodes and victim organization has 5 departments with 30 servers and 420 hosts. Authors extracted 80 features from network traffic and machine logs captured via CICFlowMeter-V3.

In the following, the presented free network datasets are discussed. As shown in the comparison table I, KDD99 is the most popular network dataset. It has been used since 1999. Unfortunately, it is out of date. To overcome KDD99 limitations, NSL-KDD was created. It has balanced data with no duplicate records. Since NSL-KDD lacks modern attacks, UNSW-NB15 was proposed. It is a well reputed dataset with recent attacks. Meanwhile, it is more complex than KDD99 in terms of similarity between the new attacks and the normal behaviors. As more recent network datasets, there are i) Sivanathan et al. dataset; ii) CICIDS and iii) CSE-CIC-IDS2018. Sivanathan et al. work is the only IoT network traffic dataset compared to the other presented ones. However, it is designed for IoT devices proliferation and not for intrusion detection. CICIDS and CSE-CIC-IDS2018 have labeled records but are not targeting IoT systems security despite their up-to-date attack list.

TABLE I: Comparison between free datasets

Datasets	Advantages	Drawbacks
KDD99 [51]	• KDD99 is popular and the most used. • Labeled data. • It is based on 41 features for each connection along with the class label. • Implements Denial of Service, Remote to User, User to Root and Probing attacks. • Provides network traffic (PCAP).	• KDD99 suffers from unbalanced classification methods. • The dataset is out of date. • Not for IoT systems.
NSL-KDD [87]	• It is a better version of KDD99. • It overcomes KDD99 limitations. • No duplicated records in the training and test sets.	• Lack of modern low foot print attack scenarios. • Not for IoT systems.
UNSW-NB15 [77]	• It provides hybrid real modern normal activities and synthetics contemporary attack behaviors. • Provides network traffic (PCAP) and CSV files. • It has nine types of attacks, namely, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms.	• It is more complex than the KDD99 dataset due to the similar behaviors of the modern attack and normal network traffic.
Sivanathan et al. Dataset [88]	• Network traffic IoT dataset. • It reflects real world IoT systems. • Provides network traffic (PCAP) and CSV files.	• Unlabeled data. • For IoT devices proliferation and traffic characterizing. • No attack data.
CICIDS [89]	• Labeled network flows. • For machine and deep learning purpose. • Provides network traffic (PCAP) and CSV files. • Implements attacks such as Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet and DDoS.	• Not public. • Not for IoT systems.
CSE-CIC-IDS2018 [90]	• Labeled network flows. • For machine and deep learning purpose. • Provides network traffic (PCAP), CSV and log files. • Implements Brute-force, Heartbleed, Botnet, DoS, DDoS, Web attacks and Local network infiltration attacks. • Dynamically generated dataset. • It is modifiable, extensible, and reproducible.	• Not public. • Not for IoT systems.

B. Free, Open Source Network Sniffers

In the following, the paper presents most free popular network sniffers which are free, and open-source software. A sniffing tool [96] aims to monitor the network transit traffic from source to destination. It can be used to capture, examine, analyze and visualize packets or frames.

- Tcpdump [91], [96] is the most popular, powerful and widely used packet analyzer. It a TCP/IP command-line tools which enables capturing, analyzing, saving and viewing packet data. Van Jacobson, Craig Leres, and Steven McCanneIt develop tcpdump at the Lawrence Berkeley Laboratory, UC, Berkeley. Tcpdump captures live packet data from a network interface. An interesting feature in tcpdump is the possibility to save the captured packets in a pcap file for a further analysis. Tcpdump uses the libpcap library to capture packets. Libpcap, is frequently used by other capture programs. Tcpdump tool is available for most of the Linux/Unix based operating systems. The most popular open source GUI (Graphical User Interface) based on tcpdump is Wireshark (third-party software) which reads tcpdump pcap files enabling an easy-to-use, user friendly interface.
- Wireshark [102], [96], [98] is a popular, free and open source packet analyzer, under the GNU license. It is used for network sniffing and network analysis. It captures live packet data from a network interface. Due to trademark issues, Wireshark was renamed Ethereal in May 2006. Wireshark runs on Unix-like operating systems, Solaris, and Microsoft Windows. It uses libpcap as a library to capture and filter packets; then displays records with its GUI. This tool enables reading tcpdump outputs. Wireshark decodes a large panel of protocols (¿ 400). It supports preliminary inspection of attacks in the network. Its command line version is “tshark”.
- Ettercap [99], [96] is a multi-platform network sniffer. It is “a multipurpose sniffer/interceptor/logger for switched LANs” [99] written by Alberto Ornaghi and Marco Valleri. Ettercap is known for its powerful ability in launching several different types of man-in-the-middle attacks. In addition, it provides users with many separate classic attacks and reconnaissance techniques within its interface. It sniffs live connections and filters packets as well as many other features in both active or passive way.
- Argus [100], [96] is tool to capture and analyze network flow data. It runs on several OS like like Linux and Windows. It focuses on developing network activity audit strategies. Moreover, argus treats live and captured traffic data to generate status reports / audits on detected flows with a semantic analysis. It processes libpcap and Endaces’s ERF packet data to enable the user having an

TABLE II: Comparison between free, open-source network sniffers

Network sniffers	**Advantages**	**Drawbacks**
Tcpdump [97], [98], [96]	• Long product life with different updates and plenty of features. • Well documented with a good community support. • Easy remote access with Telnet connection. • Cross-platform (has even been ported to Windows too). • Less intrusive compared to Ethereal. • Captures live packet data from a network interface. • Saves captured packet data. • Lightweight in terms of installation.	• Lacks critical analysis. • Discards invalid packets (Not helpful for detecting broken packets) . • No real GUI or administrative console.
Wireshark [97], [98], [96]	• Well documented with a good community support. • Cross-platform. • Supports large number of protocols. • Graphical tool. • Captures live packet data from a network interface. • Saves and open packet data files. • Provides detailed protocol information.	• No abnormal behavior notifications (Not an IDS). • Gathers information but cannot manipulate the network. • Resource consuming in terms of installation.
Ettercap [99], [96]	• Cross-platform. • Can be used for LAN hacking techniques. • Decodes several protocols. • Collects passwords for multiple applications • Manipulates the network by killing connections, by injecting packets and commands into active connection(s). • Extensible with additional plug-ins.	• Sniffing is a secondary feature. • Can be used as a hacker tool. • Can be detected by other network tools (for example by ettercap itself).
Argus [100], [96]	• Cross-platform. • Decodes several protocols. • Generates reports and audits about the network. • Native file system as well as MySQL support. • Efficient in large amount of network traffic analyzing.	• Not too obvious to master.
EtherApe [101], [96]	• Displays graphics for network activity with a color coded protocols mode. • Hosts and links change in size with traffic. • Can filter packets. • Supports multiple frames and packet types. • Supports file and real-time network traffic. • Good reputation among the system administrator community.	• Supports only Unix OS. • No command line version. • Captures only packet headers.

idea about what is going on a network. This tool provides information on almost all packet parameters like duration, rate, load, retransmission, delays, etc.

- EtherApe [101], [96] authored by Juan Toledo and Riccardo Ghetta in 2000 is a graphical packet sniffer and network monitoring tool. It supports only Unix platforms. EtherApe aims to represent packets, connections and data-flows visually with color coded hosts and links for the protocols. The tool also facilitates network troubleshooting. Furthermore, it supports real-time display of network packets via standard formats. Traffic may be consulted on one's own network, end-to-end (IP) or port-to-port (TCP).

In the following, the different free and open-source network sniffers are compared as represented in II. As it can be noticed, tcpdump is the most popular network sniffer (all the other sniffers try to support its outputs). It is a long life product often updated and extended with multiple features. It is well documented and enjoys community support. However, tcpdump is primarily developed for data capture unlike other tools equipped for network analysis. It is a command line tool with no real GUI. While the strength of wireshark and etherApe is in their graphical features,both can display real-time and captured network files. Wireshark is better known than etherApe. Besides, wireshark is a cross platform sniffer which is not the case for etherApe supporting only Unix platform. Moreover, unlike etherApe, wireshark takes into account both header and payload details. Regarding Argus, it is more a tool to audit network activities. It decodes several protocols for reports and audits. Concerning ettercap, apart from network data in sniffer, interceptor and logger mode, it can manipulates the network and launch different MITM attacks. It is able to collect passwords, kill connections, inject packets and commands in active connections. Hence, it can be considered more of a hacker tool than a network sniffer. Since network sniffers have been treated, the next part will review and discuss open-source NIDS.

C. Open-source NIDS

There are many free open-source NIDS tools that are exploited in sniffing, analyzing and detecting malicious events in network traffic like Snort, Suricata, Bro-IDS, etc. This section presents and compares the most popular free, open source NIDS.

- Snort [111], [104] is a lightweight intrusion prevention system capable of real-time traffic analysis and packet logging. It was first released in 1998. It supports Fedora, Centos, FreeBSD and Windows operating systems. Snort is a single threaded signature-based NIDS. It uses Talos, the most updated and popular open source rule list. It can run in three modes using different options in snort command line:
  - Sniffer mode where packets are simply read off of the network and displayed on the console.
  - Packet Logger mode logs packets to disk.
  - Network Intrusion Detection System (NIDS) mode detects and analyzes network traffic using a configure file containing detection rules such as rules to detect buffer overflows, stealth port scans, etc. Snort can detect application layer attacks such as SQL injection attack and cross-site scripting attack.
- Suricata [112] is a multi-threaded signature-based NIDS. It has several features mainly real time intrusion detection (IDS), inline intrusion prevention (IPS), network security monitoring (NSM) and offline pcap processing. The project is owned by the Open Information Security Foundation (OISF). The first beta version was released in December 2009. It supports Linux, FreeBSD, OpenBSD, macOS/Mac OS X and Windows as operating systems.
- Bro-IDS tool [113] is a network analysis framework for inspection of network traffic against malicious activities. Bro is both a signature and anomaly-based IDS. It supports many application layer protocols including DNS, FTP, HTTP, SMTP, etc. Bro system was designed and developed by Vern Paxson of ICSI's Center for Internet Research (ICIR). Bro-IDS supports Linux, FreeBSD and Mac OS X operating systems.
- Kismet [107] is a wireless network detector, sniffer and IDS. It runs on multiple platforms such as Linux, BSD, Android, Windows (with restricted hardware support), etc. Kismet works with Wi-Fi (IEEE 802.11[2]) cards, and Bluetooth powered devices for scanning discoverable BT and BTLE devices, the RTL-SDR radio for detecting wireless sensors, thermometers, and switches, and a growing collection of other capturing hardware.
- OpenWIPS-ng [108] is a modular Wireless IDS/IPS which monitors wireless traffic to detect and identify signature based attacks. It is developed by Thomas d'Otreppe de Bouvette, the creator of Aircrack software and basically runs on commodity hardware. OpenWIPS-ng is composed of sensors to capture wireless traffic and send it to the server; a server to aggregate data from all sensors, analyze, detect intrusions and send responses (attacks are logged and alerts are reported to the administrator) and finally a GUI to manage the server and display threat information. OpenWIPS-ng has extension plugins for more flexibility but does support only Linux systems.
- Security Onion [109] is a Linux distribution for intrusion detection, Network Security Monitoring (NSM) and log management. It contains a set of specific security tools including Snort, Suricata, Sguil, Bro, Elasticsearch, Logstash, etc. that work independently or together to detect malicious activity in vLANs and visualized networks. Security onion main features are : i) full packet capture; ii) NIDS and HIDS; and iii) powerful analysis tools.
- Sagan [110] is a real-time log analysis and correlation engine developed by Quadrant Information Security. It runs on Unix OS and it is written in C with a multi-threaded architecture to detect malicious activities at both log and network levels with a high performance. Sagan is

[2]IEEE 802.11 is the most used standard for Wi-Fi protocol and media access control (MAC) and physical layer (PHY) in wireless local area networks (WLAN).

TABLE III: Comparison between open-source NIDS

IDS	Advantages	Drawbacks
Snort [103]	• Lightweight intrusion detection [104]. • Long product life with different updates, new features and plenty of administrative front-ends. • Well documented with a good community support. • Well tested. • Easy to deploy.	• No real GUI or easy to use administrative console. • Problems of packet loss when the process rates 100-200 megabytes per second before reaching the processing limit of a single CPU.
Suricata [105], [106]	• Multi-threaded architecture for fast network traffic analysis. • Network traffic inspection can be built using graphic cards (hardware acceleration). • Detects file downloads. • Can use LuaJIT scripting to detect complex threats easier and more efficient. • Logs more than packets like TSL/SSL certs, HTTP requests and DNS requests.	• It requires more memory and CPU resources than Snort.
Bro-IDS [105], [106]	• Has a signature and anomaly-based detection methods. • Sophisticated signatures. • Analyzes network traffic at a much higher level of abstraction. • Stores information about past activity and incorporates them for analysis of new activity. • Supports high speed network.	• Bro is a UNIX platform. • Bro-IDS is based on log files without any GUI, maintained by Bro Project. • Needs expertise for set up.
Kismet [107]	• It can extend its functionality to networks of other types via plugins. • It allows channel hopping to find as many network as possible. • Undetectable while sniffing packets (passively monitors wireless networks). • Widely used and up to date open source wireless monitoring tool. • Live streaming of real-time captures over HTTP.	• Can not directly obtain the IP addresses. • Only for wireless networks.
OpenWIPS-ng [108]	• Modular and plugin-based (additional features can be integrated). • Software and hardware required can be built by non-professionals. • Higher detection accuracy since it supports multiple sensors.	• Only wireless networks are considered. • The traffic between sensor and server is not encrypted. • Not famous and not very developed. • No detailed documentation and no big community support.
Onion Security [109]	• Flexible system. • Easy Network Security Monitoring and event driven analysis because of the real-time graphical interface Sguil. • Provides so many pre-installed and easily configurable functions and software (Wizard installation). • Has regular updates to improve security levels.	• Inherits the drawbacks of each constituent tool. • Does not work as an IPS after installation, but only as an IDS.
Sagan [110]	• Fast (multi-threading) and real-time log processing. • Supports multiple output formats and log normalization. • Allows the geographical location of the IP addresses. • Can distribute its processing over several devices. • Lightweight CPU and memory resources. • Actively developed and easy to install.	• Is primarily developed for log analysis rather than for intrusion detection.

essentially a HIDS and was extended to be considered as a signature-based NIDS also. It is compatible with data gathered by Snort, Bro, Suricata and other tools.

In the following, advantages and disadvantages of the presented NIDSs are discussed as illustrated in Table III. On one hand, Snort is an old and an up to date IDS. It is well documented and well tested. Unfortunately, it suffers from some problems related to packet loss. On the other hand, Suricata offers a multi-threaded architecture but, requires more memory and CPU resources compared to Snort which could represent problems especially in IoT context. Bro-IDS analyzes network traffic at a higher level of abstraction but requires UNIX platform and does not have a GUI supported by Bro Project. However, Kismet supports multiple platforms but works only for wireless networks [114]. The fact that Kismet is undetectable when it sniffs networks is an important advantage. Concerning OpenWIPS-ng, it is also specialized in wireless networks. Despite it's modularity and scalability, this NIDS not famous and does not have community support. Finally, Onion security and Sagan are more big tools. They bring many tools in one solution like Snort and Bro-ids in the same tool. They are considered more as Security Information and Event Management (SIEM). So they offer a rich panel of tools. They guarantee the ease of deployment and use of the integrated systems. Both can handle real-time monitoring, but Sagan announces its advantage in terms of light CPU and memory consumption which is relevant for IoT systems.

In this Section, tools that can be used to build a NIDS have been presented and discussed; from free network datasets to free network sniffers, to free, open source NIDSs. The rest of the paper will introduce and compare NIDS proposed from researchers specifically for IoT systems security.

IV. Network Intrusion Detection Systems for IoT (NIDS)

To get a better idea about IoT NIDS architectures and deployments, the survey discusses in the following, relevant

works from the state-of-art of NIDS IoT security. Only solutions that are not based on machine learning will be treated since this special case will be the main concern in Section V.

Researches about NIDS for IoT demonstrate that there are mainly two axes of works: i) NIDS for Wireless Sensor Networks (WSN)[3] [115] and ii) NIDS for IoT systems in general. The first axe is out of the scope of our paper for two reasons; first because many surveys have treated it in details like [116], [117], [118] and [28]; second, because our survey concentrates on IoT systems with its heterogeneity and all the previously presented challenges and not in NIDS that do not take into account Internet enabled attacks (WSN NIDS) [8]. The second axe treats researchers ideas and implementations as a continuity and an improvement of aforementioned IoT NIDS surveys [28] and [8]. This part describes works about NIDS for IoT systems with a special focus on NIDS detection mechanisms, architectures and validation strategies. The reviewed IoT NIDS papers are important in the state-of-art. They enable to follow the evolution of the domain from the first proposed solution [9] to the present day. This subsection starts with detailed description of the authors' solutions. Then it summarizes them in a comparative Table IV with the advantages and disadvantages of each solution.

A. State-of-art of NIDS for IoT

This Section details each IoT NIDS proposal with a special focus on architectures, detection methodologies and treated threats.

Raza et al. [9], [119] designed and implemented SVELTE, the first IoT IDS. It is a real time intrusion detection system based on a hybrid signature accompanied by an anomaly based detection technique. The work meets the requirements of IPv6-connected IoT and concentrates on routing attacks such as spoofing and sinkhole. SVELTE considers IoT challenges and deploys lightweight IDS modules in resource constrained nodes and resource-intensive IDS modules at the Border Router (BR). It integrates three main modules: i) 6Mapper (6LoWPAN[4] Mapper) which gathers information about the RPL[5] network and reconstructs the network in the 6BR, ii) Intrusion Detection component that analyzes mapped data for intrusion detection; and iii) a mini-firewall (whitelist firewall for the IP-connected IoT that uses RPL as a routing protocol in 6LoWPAN networks) which is distributed and designed to offload nodes by filtering unwanted traffic before it enters the resource constrained network. SVELTE was implemented in the Contiki OS. It detects sinkhole attacks with 90% true positive rate (TPR) in a small lossy network and almost 100% TPR for a lossless network configuration. Unfortunately, DoS attacks can affect the solution [120] as well. Since IDS nodes use the network to transmit attack information, once DoS affects the network, it fails to detect denial of service attack.

Kasinathan et al. studied [12], [120] DoS detection for 6LoWPAN developed as a part of ebbits, a EU FP7 project[6]. The supported architecture presented in Fig. 6 is based on Suricata IDS (Section III-C). The proposed architecture is considered centralized despite the distributed IDS probes. In fact, IDS probes which are external modules, sniff the network in promiscuous mode than send data to the main NIDS (based on Suricata) via wired connection. When the latter matches traffic with an attack signature, an alert is launched to the DoS protection manager. The protection manager analyzes the attempt with additional data collected from other ebbits managers, reduce the false alarm (incorrect detection genuine data: false positives plus false negatives) rate. This solution overcomes SVELTE limitations since the IDS mechanism does not depend on the network architecture so it cannot be affected by DoS attacks against the IoT network. The suggested framework DEMO in [120], is scalable and real-word applicable for most IoT systems. This work was evaluated using a penetration testing (PenTest) system Scapy [120] which is more light-weight then Metasploit [121]. The IDS adapts existing open source technologies. It starts with Suricata which is an open-source IDS and modifies it with IEEE 802.15.4 and 6LoWPAN decoders. Further, an additional detection module; FAM (consists on frequency agility manager which analyzes channel occupancy states in real time to allow the network to become aware of the interference level) and monitors attacks on Prelude (which is a security incident and event management system (SIEM) to monitor the attack events or alerts). The Suricata engine triggers alerts based on the rules programmed; therefore this solution could detect different attacks depends on developed rules.

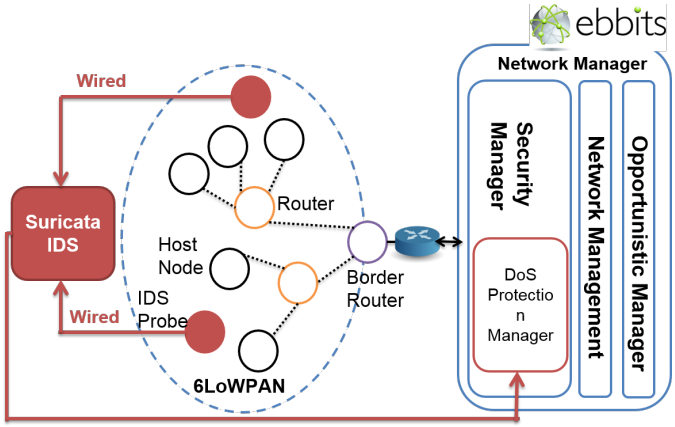

Fig. 6: DEMO architecture [12], [120]

Jun and Chi [122] proposed an IDS for IoT systems based on Complex Event Processing (CEP) technology which is an emerging and efficient technology to filter and process real-

[3]A WSN is wireless geographically distributed network of sensors to monitor physical or environmental conditions.

[4]IPv6 over Low-power Wireless Personal Area Network

[5]Routing Protocol for Low-Power

[6]The ebbits project is a European research project which deals with architecture, technologies and processes to enable mainstream enterprise incorporate IoT eco-system.

time events[7]. It is a good solution for large volumes of messages with low latency. So such a technology can be adapted to IoT needs. Jun and Chi evaluated on-line performance rather than off-line. The architecture of this solution is schematized in Fig. 7. The system starts with collecting data (network traffic and event usage) from IoT devices, extracts events from sensed data, then perform security events detection using Event Processing Repository EPR[8] and CEP engine[9]. Finally, actions are performed by the action engine. Jun and Chin implemented their event-processing IDS architecture using Esper (CEP engine for complex event processing and event series analysis). Their approach is CPU intensive, but consumes less memory. Effectively it proved better real-time performance. For example, for 800k of data, CEP-based IDS consumes 62% of CPU, 730MB of memory and 422 millisecond processing time. However, traditional IDS utilizes 57% of CPU, 1064MB of memory and 8688ms for processing 800k of data. It is interesting to note that, the framework is designed but not evaluated for any kind of attack detection.

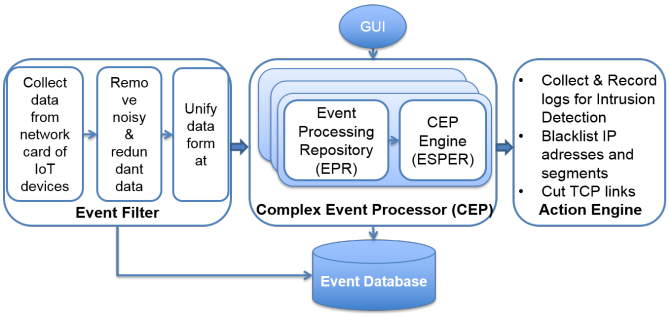

Fig. 7: CEP-based IDS architecture for IoT [122]

Cervantes et al. [123] detected the dangerous sinkhole attack on the routing services in IoT. They proposed Intrusion detection of SiNkhole attacks on 6LoWPAN for InterneT of ThIngs (INTI). It combines watchdog, reputation and trust strategies for detection of attackers. First, as a hierarchical structure, the nodes (grouped or separated) are classified as leaders. Then, the nodes can change role over the time based on network requirements. Each node monitors a number of transmissions performed by a superior node. If an attack is detected, an alert message is broadcasted and a cooperative isolation of the malicious node is performed. Cervantes et al. gave importance for node mobility and network self-repair, which are limitations with Raza et al. [9] approach. Their simulation results show sinkhole detection rate of 92% on 50 fixed nodes scenario and of 75% for 50 mobile nodes. Also, authors reported low false positives and false negatives compared to SVELTE.

Surendar and Umamakeswari [124] set up an Intrusion Detection and Response IoT System (InDReS) with 6LowPAN. InDReS uses constraint based specification technique to detect sinkhole attacks in RPL networks. In InDReS, sensor nodes are grouped into clusters under the supervision of an observer node. Observer nodes count the dropped packets of their adjacent nodes and assign a score to each of them using Dempster Shafer theory to detect malicious node. The latter will be announced so that all the nodes co-operate to isolate it. Finally, the network reconstruct itself. Authors' strategy improves efficiency of some critical QoS metrics over the existing INTI scheme which is limited by average energy consumption and packed drop ratio. Authors simulated their proposal on NS2 simulator.

Fu et al. [125] presented uniform intrusion detection system while considering the following two important points: i) varied heterogeneity of IoT networks; ii) IoT sensor and smart device resource constraint. Authors claimed that, their solution is the first one that benefits from automata theory to model and detect the intrusions of IoT networks. Based on an extension of Labelled Transition Systems, they provided a uniform description of the IoT systems network traffic flows then compared the real-time action flows with Standard Protocol Libraries to detect and report: jam-attack, false-attack and reply-attack. The proposed approach is composed of following four components:

- Event Monitor collects the network traffic and transmits data into digital files to the IDS Event Analyzer. This component should be implemented on the PAN (Personal Area Network) co-ordinator or other IoT gateways to monitor the network traffic.
- Event Database implements three databases stored on the cloud: i) Standard Protocol Library (the description of the standard protocols through Glued-IOLTS [126]); ii) Abnormal Action Library (recognized anomaly actions flows for the system); and iii) Normal Action Library (possible action flows created from the Standard Protocol Libraries using the techniques of Fuzzing [127] and Robustness Testing [128]).
- IDS Event Analyzer is composed of the following three basic models:
  1) Network Structure Learning Model: considers packet data as input, builds a general view of the network topologies, distinguishes IoT devices IDs and sends them to the Action Flows Abstraction Model,
  2) Action Flows Abstraction Model: classifies the collected real-time packets from IoT into message sequences then translates these messages to abstract action flows with the help of Standard Protocol Library,
  3) Intrusion Detection Model: compares the result of Action Flows Abstraction Model with the Abnormal Action Library. If it matches, the action flow is marked intrusive; otherwise, an anomaly detection method will be applied. For the latter, if the input transition sequence does not match entries of the Normal Action Library, an expert manual verifica-

[7]CEP technologies merge multiple sources data to interpret real time actions from complicated events or patterns.

[8]EPR is a repository of Event Processing Model statement EPM which is a collection of events correlation.

[9]CEP engine analyzes a mass of events, identifies the most important ones, and produces actions.

tion is needed (to avoid the false positive). If it is finally marked as safe, then the record is added to the Normal Library, otherwise, it is added to the Abnormal Action Library.

- Response Unit: reports three types of attacks (jam-attack, false-attack and reply-attack) to a management station.

Fu et al. experimented their solution on IoT experiment environment but unfortunately, they did not present detection rates.

Midi et al. [129] proposed Kalis which is "the first approach to intrusion detection for IoT that does not target an individual protocol or application, and adapts the detection strategy to the specific network features". Kalis is a "network-based, hybrid signature/anomaly-based, hybrid centralized/distributed, online IDS that adapts to different environments". It can be deployed as a standalone tool on a separate external device (to overcome the fact that most IoT devices does not support software changes). It is an automatic knowledge-driven IDS which means that it chooses automatically detection techniques depending on collected network's features. Precisely, each attack could be done only in some IoT systems and not in others depending on the system features for example it is not possible to have replication attack in a single hop system.

Kalis identifies even the presence, or not, of prevention techniques such as the use of cryptographic functions. Hence, Kalis is effective and efficient regarding resource consumption. Kalis was implemented using Java on an Odroid xu3 development board and evaluated with real-world IoT devices. The system includes "small WSN of six TelosB nodes, a Nest Thermostat, an August SmartLock, a Lifx smart lightbulb, an Arlo security system, and an Amazon Dash Button". To sniff intermediate hops of data packets, Kalis was located near the middle portion of the WSN. Midi et al. replayed actual traces of network traffic of the prototype and added additional packets with 50 different symptom instances for each attack. The detection rate of Kalis is 91% with 100% accuracy, 0.19% CPU usage and 13978.62 KB memory consumption. Here, its is important to note that the traditional IDS use have around 48% detection rate with 75% accuracy, 0.22% CPU usage and 23961.06 KB RAM.

B. Comparison and Discussion

In the following, a comparison of previously reviewed proposals for IoT NIDS is given. A summary and a visual comparison are provided in Table IV.

As it can be noticed, most of the works deploy distributed architectures [9], [123], [124], [125]. This **type of deployment** is more suitable for IoT systems than centralized strategies [120], [122] since the distribution of devices is an important IoT characteristic. However, centralized IDS detect better security attacks that involve a group of devices operating silently (without directly shutting down the network) than the distributed ones. For example, DDoS attacks are difficult to detect in a distributed deployment. For such attacks, a hybrid architecture such as [129] is more appropriate. Hence, a distributed network analysis with a centralized general inspection is guaranteed (called also hierarchical strategy). Moreover, deployment of NIDS on the IoT system itself or in a separate external device is an important. Kasinathan et al. [120] and Midi et al. [129] are the only ones who proposed their NIDS as a standalone tool. The adoption of such a strategy is considered constrained since resource constraints no longer pose a challenge. This overcomes the problem of IoT devices without software changes. This enables the protection of the initial IoT system from network and device overloading. Thus, employing additional infrastructure adds complexity in case of network maintenance and system protection.

Concerning **detection methodology**, both signature and anomaly detection are deployed. Each method has its advantages and its drawbacks. Signature-based detection is efficient for known attacks; however, it cannot detect unknown attacks since the signature database must be updated and time consuming. When the size of signature database increases, NIDS is required to compare the input with all the existing signatures. Anomaly detection detects unknown / unseen attacks; however, it suffers from high false alarms. Consequently, hybrid detection such as [125] and [129] have been deployed as practical solutions.

Regarding the **validation strategy**, two important parameters are identified: simulation and emulation. Simulation models the behavior of the target system in a different environment. It provides the basic behavior of a system; it may not necessarily adhere to the rules of the original system. Emulation duplicates the exact same target behavior of the original system operating in a different environment. Therefore, emulation is more close to real life situation when compared with simulation [120], [125] and [129]. Simulation is acceptable in IoT since the implementation of an IoT system requires a large number of physical devices to get closer to reality which is not an easy task for experimental research. The second point to discuss about validation is the evaluation metrics. Findings of the review show that researchers does not always provide the same metrics [130], [131] in their works evaluation which does not allow a true fair comparison;

- Detection rate (DR) is the ratio of true intrusion detections to the total number of intrusions. DR is different from precision rate (positive predictive rate) which represents a fraction of data instances predicted as positive that are actually positive. Cervantes et al. [123] reported 92% detection rate on a network composed of 50 fix nodes and 75% for 50 mobile nodes. Midi et al. [129] achieved 91% detection rate against 48% with traditional IDS and 89% with snort tool.
- Accuracy is the ability to differentiate intrusions and normal behaviors correctly. It represents the ratio of correctly classified intrusions to the total number of inputs. Midi et al. [129] achieved 100% accuracy, whereas traditional IDS had 75% and snort reported 76%.
- False positive rate (FP) represents normal traffic misclassified as intrusive. Fu et al. [125] considered FP metric

TABLE IV: Comparison of NIDS for IoT

References	IDS deployment	Detection Methodology	Validation Strategy	Treated Threats	Advantages	Disadvantages
Raza et al. [9], [119]	Distributed	Hybrid (signature and anomaly based)	Simulation	Routing attacks like spoofing and sinkhole, selective forwarding and information alteration	• Resource constraints challenge is taken into consideration • Distributed mini-firewall for the IP-connected IoT devices is integrated • Flexible and can be extended to detect more attacks	DoS attack can affect SVELTE
Kasinathan et al. [12], [120]	Centralized	Signature-based	Emulation	DoS attack	• False alarms reduction • IDS is deployed on additional infrastructure • Scalable and real-word applicable	Detected attacks depend on declared rules
Jun and Chi [122]	Centralized	Signature-based	—	—	• Real-time detection • Better real-time performance • Low memory consumption • IoT Massive data are taken into consideration	• CPU intensive • Detected attacks depend on declared rules
Cervantes et al. [123]	Distributed	Hybrid (trust and reputation strategy)	Simulation	Sinkhole attack	• INTI takes into consideration node mobility and network self-repair • Less false positive and false negative rate than SVELTE	• IDS placements change over the time which can consume more resources.
Surendar and Umamakeswari [124]	Distributed	Specification-based	Simulation	Sinkhole attack	• Resource constraints challenge is taken into consideration • Low average energy consumption • Low packet drop ratio • Instant network response against detected attacks	• Cannot detect unknown attacks
Fu et al. [125]	Distributed	Hybrid (signature and anomaly based)	Emulation	Jam-attack, false attack and reply attack	• Heterogeneity of IoT networks is taken into consideration • Resource constraints challenge is taken into consideration • Low false positive rate	• The state based algorithm may cause "state space explosion" • Human intervention is needed for false positive alarms • DoS attack can affect the solution
Midi et al. [129]	Hybrid (centralized and distributed)	Hybrid (signature and anomaly based)	Emulation	DoS, routing and conventional network attacks	• Real-time detection • Lightweight in terms of CPU and RAM requirements • Dynamic self-adapting IDS • Automatic knowledge-driven IDS • Different IoT communication protocols and applications are taken into consideration • Deployable on border router or as a standalone tool	• High level perspective may not suitable for constrained compute objects. • Kalis proposes compile time deployment which may not be feasible for resource constrained sensors which may even be resource constrained in comparison to WSN nodes.

while discussing their solution without concrete value.

- True positive rate (TP) successfully identifies attack which refers to the number of intrusions that are detected as intrusions. Raza et al. [9] reported 90% TP in a small lossy network and 100% in a lossless one.
- Energy consumption and packet drop and packet delivery rate are the metrics used by Surendar et al. [124]. They achieved better results compared to Cervantes et al.
- CPU usage and memory consumption are taken into consideration into Midi et al. experiments [129]. They consumed 0.19% of CPU and 13978.62Kb of memory vs 0.22% and 23961.06Kb in traditional IDS and 6.3% and 101978.24Kb with snort.

Some of works did not provide experimental results like in [120] or did not even experiment their solution like in [122]. Evaluation metrics need to be fixed and processed in each work to have a reliable comparison even if the used metrics depend on the objectives and aspects on which each study focuses.

About the **treated attacks** in reviewed papers, as in Table IV,

there is no work that takes into consideration all the threats at the same time. Normally NIDS based on anomaly or hybrid detection methodology should be able to detect all types of attacks but no one of the reviewed works concentrate on detecting the maximum attack types. [9] is the only work which mentions that the solution could be expanded to detect more than the experimented attacks.

IoT is an environment of coexisting protocols and technologies. Despite the heterogeneity aspect of IoT, [122], [125] and [129] have the capability to detect multiple protocols. [9], [120] and [123] focus on intrusions in 6LoWPAN and RPL which are important techniques for IoT networks. Furthermore, complexity is high due to heterogeneity. Moreover, resource constraints challenge is considered by [9], [120], [124], [125] and [129]. Scalability, on the other hand, has been a subject of study in [120] and [122]. For [122], it was more concerning data scalability. Finally, [123] is the only proposal which considers mobility and connectivity.
Strengths and weaknesses for each solution have been identified as illustrated in Table IV.

This Section have discussed traditional NIDS for IoT systems. Details on their architecture, detection methodologies and experimental results have been provided. Furthermore, we have performed a thorough comparison of NIDS for IoT, their strengthens & weaknesses while evaluating their pros and cons. The next Section discusses, evaluates and compares IoT NIDS based on learning techniques.

V. NIDSs for IoT systems based on Learning techniques

Before moving from NIDSs for IoT to the ones based on learning techniques, the paper discusses learning techniques briefly, defines them and discusses their classification. On the other hand, IoT NIDSs powered by learning techniques are surveyed. State-of-art works are compared and discussed to extract strenghts and weeknesses of each one.

A. Learning techniques

IoT powered technology advances are supplemented via cloud storage, data mining and big data analytics. In 2015, Ben Walker at vouchercloud [132] reported that current web applications generate 2.5 Quintillion bytes each day. The massive data explosion is due to the relevance of social media and IoT in daily lives. Further, more than ninety percentage data is unstructured and/or incomprehensible. Here comes the role of Big Data [133] and learning techniques. Big Data offers technologies and architectures that aims to treat, organize and take profit from huge data. It enables distributed parallel treatment. According to Gartner, Big Data helps dealing with 3V rules: Volume (explosion of data volume), Velocity (frequency of data creation is varied and can attend fractions of seconds) and Variety (data are generated from different sources and in different structured/unstructured format).

After preparing and treating collected data, the role of analytics and intelligent algorithms begins. **Big data analytics** enables discovering market trends, customer preferences, predict customer behavior via associating dependencies between input variables or with Data Mining (DM) tools and techniques. **Data mining** is about exploring the data for hidden pattern, relationships and previously unknown correlation to predict and react. With data mining comes Machine Learning (ML) concept. **Machine learning** is an artificial intelligence technique that is widely used in data mining. In year 1959 Arthur Samuel, the pioneer of Machine Learning (ML), defined ML as "field of study that gives computers the ability to learn without being explicitly programmed" [134] . It consists in the deployment of algorithms in order to obtain a predictive analysis from data (Learning from examples). There are mainly two types of ML algorithms:

- Supervised learning is based on learning from labeled training data which means that training data includes both the input and the desired results.
- Unsupervised learning is based on clustering the input data in classes on the basis of their statistical properties only. In other words, it describes hidden structure from "unlabeled" data (no predefined classification or categorization in the observations).

ML is based on a set of features that identify a state of an object. Features should be chosen carefully and accurately to avoid inaccuracy and unnecessary time consumption. To achieve good data learning with accurate and efficient results, unrelated or irrelevant features must be removed from the set. Such a process is known as features reduction or dimensionality reduction. It is necessary tool for analyzing high dimensional [135], noisy data. Applying a dimension reduction on an original high-dimensional data in terms of features would preserve (feature selection [136]) or generate only important features.

Deep Learning (DL) [137] is part of a broader family of ML methods based on learning high-level abstract representations. DL groups generic algorithms mimicking the biological functioning of a brain without being intended for a specific task. Technically, DL is the application of artificial neural networks (ANNs) which contain multiple hidden layers.

Researchers start to put more energy in exploiting machine learning algorithms in NIDS for many reasons:

- Unknown / zero day attacks evade the traditional signature based NIDSs; whereas supervised machine learning algorithms have an interesting potential in novel attacks detection.
- Traditional NIDSs suffer from high false recognition rate which can be reduced with machine learning techniques. Compared to Signature/Non-Signature IDS, an ML equipped IDS/NIDS employs statistical, genetic and heuristics or a combination of them to disseminate complex attack pattern to improve the detection rate with reduced False Negatives.
- Traditional solutions suffer from attack complex proper-

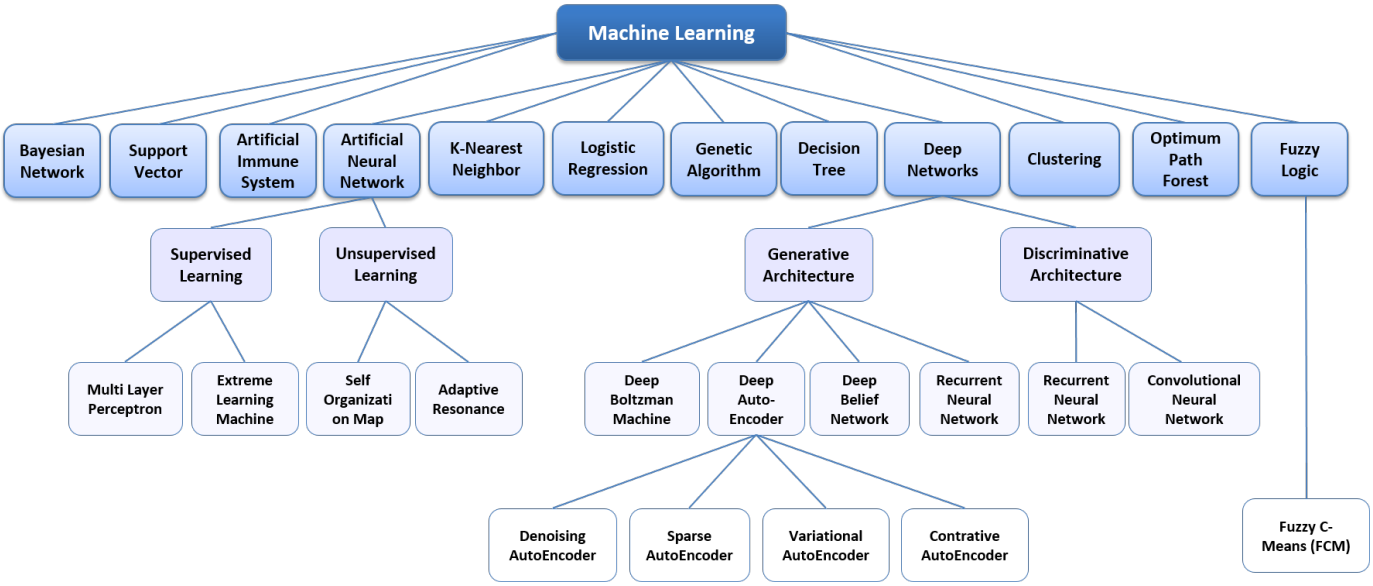

Fig. 8: ML algorithms

ties capture, whilst, machine learning can improve the detection accuracy and speed.

- Slight variations in attacks can not be effectively detected with traditional NIDSs. Even a Heuristic detector can be evaded by inverting the attack pattern. However, ML equipped IDSs learn recent traffic pattern continuously; hence, they effectively identify minor variation in traffic pattern. In other words, ML algorithms are efficient against variant detection [27].
- Cyber criminals deploy evolving attack patterns to evade the detectors. Traditional NIDSs, especially signature-based NIDSs require continuous updates. However, ML NIDSs based on clustering and outlier detection do not necessitate regular updates.
- Traditional solutions and more precisely signature based IDSs match each signature with IDS database. The process is CPU consuming (large signature database which grows exponentially); whereas, ML based IDS consume low to medium processing. Hence, the processing element can be used effectively.

Consequently, learning techniques seem to be a suitable solution especially with the good results that they achieve in the different domains.

A classification for the most popular ML algorithms is established in Fig. 8. The ML taxonomy is presented in this paper to help the reader identify the type of the used algorithms in the NIDSs reviewed later. However, detailed ML algorithm description is out of the scope of our article. Multiple surveys such as [24], [25], [13], [15], [26] and [27] have treated ML for intrusion detection. A description about each survey is detailed.

B. IoT NIDSs based on machine learning techniques

With the evolution, complexity and diversity of security attacks, researchers are focusing on the use of artificial intelligence and machine learning for security threats detection. To do that, IDS must embed the machine intelligence and improve decision making capabilities [138]. Many studies apply ML in IDSs and prove promising results. Agrawal et al., Buczak and Guvan, Fadlullah et al., Hodo et al., Wang and Jones and Mishra et al. present different learning IDSs in their surveys [25], [15], [26] where ML algorithms such as decision trees (DT), support vector machines (SVM), naive bayes (NB), artificial neural networks (ANN), k-means clustering, fuzzy logic, genetic algorithms, stacked auto encoder (SAE), were deployed separately and combined [139], [140] for improved outcome in general systems. As evoked in [19], science started with applying each machine leaning algorithm separately than the trade on combining the algorithms in the same system takes place.
Since our main focus is towards the deployment of intelligent IDS in IoT, we now singularly discuss NIDSs for IoTs employing learning techniques. The rest of the paper gives detailed description of each proposal than in the next subsection, researchers choices and results will be discussed.

Hodo et al. [17] used Multi-Layer Perceptron (MLP) which is a type of supervised Artificial Neural Network (ANN) in an off-line IoT IDS. Their analysis is built on internet packet traces and tends to detect DoS and DDoS attacks in IoT network. Their MLP characteristics are:

- three-layer feed-forward and backward Neural Network.
- unipolar sigmoid transfer function in each of the hidden and output layers' neurons.
- stochastic learning algorithm with mean square error function.

NIDS was tested on a simulation composed of four clients nodes and a server relay node. DOS/DDoS attacks were performed on the server node with 10 million UDP packets sent from a single host for DoS attack and with three hosts at wire speed for DDoS. The training dataset was composed of 2313 samples, out of which 496 samples were deployed

for validation and 496 samples were used for testing. Overall attack detection accuracy was 99.4% with 0.6% false positive. Such results guarantee a good stability of the network.

Nobakht et al. [141] proposed a host-based IDS framework IoT-IDM for user-chosen smart devices in smart homes environment. IoT-IDM monitors traffic going through the devices to identify threats. The framework takes benefit of Software Defined Networking (SDN) architecture with machine learning techniques to detect compromised hosts and mitigate these attacks by pushing the appropriate actions (like blocking the intruder or redirecting the malicious traffic) to underlying routers/switches. Nobakht et al. deployed OpenFlow protocol which is a network standard to deploy SDN implementation. SDN abstracts network services. It separates control plane (the decision maker about data forwarding) from data plane (the responsible of sending the data). This technology offers the opportunity of remotely managing the security which leads to provide the user of IoT-IDM with a Security as a Service (SaaS). Nobakht et al.'s solution is characterized with modularity in design: it is composed of five separate modules (Device Manager, Sensor Element, Feature Extractor, detection Unit and Mitigation Unit). Consequently, there is a flexibility to choose machine learning algorithm from a set of given techniques. ML algorithms use learned signature patterns of known attacks to train the model. Besides the wide range of detected attacks, one of the drawbacks of IoT-IDM is that technically it cannot survey all home IoT devices due to high volume of network traffic with the detail that sensor elements are positioned on top of SDN controller. Consequently, IoT-IDM can only inspect chosen IoT devices that do not overload the SDN controller. Nobakht et al. tested IoT-IDM on a real IoT device which is the smart light bulb (Hue lights) and compare logistic regression and SVM (support vector machines) machine learning techniques. In unauthorized detection, the first one gives 94.25% of accuracy rate and 85.05% of recall rate against 98.53% and 95.94% for SVM.

Hosseinpour et al. [18] proposed a novel real-time, distributed and lightweight IDS based on Artificial Immune System (AIS), an effective combination of edge, fog and cloud computing. Cisco introduced Fog computing concept to extend cloud computing at the network layer. The fog layer is between the IoT sensors and the cloud. The layer is equipped with fog computing capacity (intelligent data processing at an intermediate level) for efficiency and reduced data transport to the cloud. Consequently, the processing takes place in hubs, routers or gateways. Such a technology enables distributed attack detection; efficient in terms of scalability, autonomy in local attack detection, acceleration on data training near sources and neighbor' parameters sharing. Authors evaluated detectors in the edge layer, intrusion alerts treated with smart data concept at the fog layer. A clustering of primary network traffic and detector training are performed at the cloud. Following are the important work strengthens: i) Fog computing enabled quality of service with low latency in data analysis; ii) Combination of lightweight analysis in fog layer with an advanced analysis in the cloud; iii) Detection of silent attacks such as botnet attacks using smart data strategy ("an active and intelligent data structure which facilitates the management of Big Data in IoT" [18]); and iv) detection of unknown and zero-day attacks via AIS based on an online self training method with unsupervised machine learning. The AIS algorithm of the IDS is composed of following three parts:

1) A training engine: learns from an initial learning dataset and trains detectors (initialization phase of the AIS). This step is treated in the cloud layer since it needs complex and powerful processing units.
2) An analyzer engine: analyzes anomalies reported by the detectors to alert and reject the false positive signals. The authors use memory cell detectors and genetic algorithms as presented in previous works [142] and [143] to improve precision. This step requires more communication between the infected edge nodes and the main engine, hence the analyzer engine is deployed at the fog layer.
3) Detector sensors: detection logic is inserted in each node monitoring the network. The proposed IDS is doted with an intelligent and distributed detection where each type of attack, could be detected by a number of different detectors. If a threshold is reached, the anomaly will be reported to the analyzer engine, thus a deep intrusion alert is generated.

Two datasets have been used to evaluate the lightweight IDS efficiency which are KDD-Cup99 and SSH Brute Force from ISCX dataset [144]. According to experimental results, the three-layered proposed solution achieve 3.51% of false positive rate with 98.35% of accuracy and 97.83% of precision.

Bostani and Sheikhan [37] suggested a real-time hybrid of anomaly-based and specification-based Internet of Things IDS. It enables the detection of sinkhole and selective-forwarding attacks in 6LowPAN networks and can be extended to detect blackhole, rank and wormhole attacks. This IDS works mainly in two steps: specification detection in router level and anomaly detection in the root level. For the first one, the routers analyze features locally from network traffic and host nodes. The results on the first step are sent to the root node for the second step and removed from routers to ensure lower consumption of memory and CPU cycles. The second step is the global intrusion detection where anomaly-based analysis is performed on incoming data packets at the root node. This step employs the unsupervised optimum-path forest algorithm (OPF) to create clustering models for each source node router. With a MapReduce architecture platform, a parallel, distributed execution of anomaly detection according to clustering models is ensured. The final decision about tagging a suspicious behavior as an attack is done with a voting mechanism. The proposed system neither uses additional control messages, nor makes use of additional infrastructure. Consequently, it saves on communication and setting cost compared to other IDS. Authors evaluated the proposed

technique on their own simulation tool. They prove appropriate real-time detection results with three main experiments each one done with ten simulations: the first experiment deals with values of evaluation criteria, the second experiment tackles the scale of the networks (small and medium size) to confirm independent scale-network IDS and the third one proves the possibility of extending the detected attacks such as wormhole. The experimental results of simulated scenarios showed that when both sinkhole and selective-forwarding attacks were launched simultaneously, the proposed hybrid method can achieve true positive rate of 76.19% and false positive rate of 5.92%. However, for wormhole attack the rates are 96.02% and 2.08%, respectively.

Bostani and Sheikhan resumed in [145], [146] the same architecture as given in [37] (i.e., based on distributed MapReduce Model). They proposed an anomaly and misuse agents with supervised and unsupervised optimum-path forest model instead of anomaly and specification based detection. They also reduced dataset features with a hybrid feature selection algorithm which is built on mutual information and binary gravitational search algorithm.

Pajouh et al. [147] presented an anomaly IDS built with Two-layer Dimension Reduction and Two-tier Classification (TDTC) for IoT Backbone. They concentrated mainly on low-frequency, common attacks: User to Root and Remote to Local attacks while their experiments were based on NSL-KDD dataset. Pajouh et al. deployed a two-layer dimension reduction to limit dataset's high dimensionality:

- The first layer benefits from an unsupervised technique which is Principal Component Analysis (PCA) for feature dimension reduction (combine dataset features to construct new ones). So for NSL-KDD, the overhead complexity was reduced in TDTC since only 35 out of 41 data set features were used.
- The second layer uses a supervised technique: Linear Discriminant Analysis (LDA) to make PCA reduced features better for classification and to improve the speed of intrusion detection. After analyzing the dataset classes, LDA finishes with two dimension dataset for NSL-KDD.

This dimension reduction decreases the false positive detection rate and the computational complexity. The second step is the multilayer classification where TDTC uses Naive Bayes (NB) and Certainty Factor version of K-Nearest Neighbor (KNN) to classify inputs. Pajouh et al. started with NB for anomaly detection then results are refined with CF-KNN. Their work proved computation reduction of about ten times with faster detection and less resource requirements. They achieved a detection rate of about 84.86% for binary classification with 4.86% of false alarm.

Lopez-Martin et al. [148] proposed an unsupervised anomaly NIDS for IoT based on Conditional Variational AutoEncoder (CVAE). Their method is unique due to its ability to carry out feature reconstruction i.e., it can retrieve missing features from incomplete training datasets. Authors claimed to achieve 99% accuracy in retrieving categorical features. In addition to that, this feature makes the proposed system relevant to IoT networks since they are more sensitive in terms of connection problems and sensing errors which affect sent/received data. As inputs to the Intrusion Detection CVAE (ID-CVAE), they used not only intrusion features (like in Variational AutoEncoder VAE) but also intrusion class labels. Despite the unsupervised aspect in their NIDS, they benefited from the class labels in the training phase to deploy a deviation-based NIDS employing discriminative framework (rather than threshold) that associate the low-reconstruction-error label to an input sample. Another key strength in their study is that ID-CVAE performs only a single training step to generate only one model from multiple trainings depending on the number of different labels like in VAE. This characteristic makes the ID-CVAE a suitable option for IoT systems due to the efficiency in computation time, flexibility and accuracy results. The selected dataset for ID-CVAE training and testing was a refined version of NSL-KDD. It ended with 116 features and 23 possible labels. They proved experimentally that their work is less complex compared to other unsupervised NIDS, with better classification accuracy than well-known algorithms like linear support vector machine and multi-layer perceptron, etc. The authors got an accuracy of 99%, 92%, and 71% when model recovers missing categorical features with respectively three, 11 and 70 values.

Thing [149] analyzed IEEE 802.11 network threats and proposed an anomaly network IDS to detect and classify attacks in IEEE 802.11 networks. This work is considered as the first work that employ deep learning algorithms for IEEE 802.11 standard. Thing experimented Stacked Auto-encoder (SAE) architecture with both two and three hidden layers. The author experienced different activation functions for the hidden neurons. To test his strategy, he used a dataset generated from a lab emulated Small Office Home Office (SOHO) infrastructure. He achieved an overall accuracy of 98.66% in a 4-class classification (legitimate traffic, flooding type attacks, injection type attacks and impersonation attacks).

Diro et al. [150] recommended the use of the fog computing in IoT systems to detect intrusions. Fog computing is about equipping the fog layer (hubs, routers or gateways) with an intelligent data processing at an intermediate level in the aim to improve efficiency and reduce the data transported to the cloud. Such a technology enables distributed attack detection which is more efficient in terms of scalability, autonomy in local attack detection, acceleration on data training near sources and neighbor's parameters sharing. Authors proposed a deep learning approach to detect known and unseen intrusion attacks. Known attacks represent 99% which leads to affirm that zero-day attacks are crafted with small mutations in the old ones. Therefore, multi-layer deep networks enhance small changes awareness (in a self taught algorithm with compression capabilities) compared to shallow learning classifiers. Distributed deep learning approach is based on distributing the dataset to train each sub-dataset locally and rapidly than share and coordinate the learning parameters with neighbors. So the architecture ends with a

master IDS which updates the parameters values of the down distributed IDSs and keeps synchronization. The studies show that the distributed parallel deep learning approach realize better results in accuracy than centralized deep learning NIDS and also than shallow machine learning algorithms. To train the models and evaluate the IDS, Diro et al. used NSL-KDD dataset after adding some modification on it to finish with 123 input features and 1 label. As results, they obtained multi-class detection consisting 4 labels (normal, DoS, Probe, R2L.U2R) to achieve 96.5% detection rate and 2.57% of false alarms for deep model in comparison to shallow classifier achieving 93.66% detection and 4.97% false detection rate. They also noted an increase in the overall detection accuracy while adding the number of fog nodes from 96% to 99%. The proposed approach taked longer training time; however, real detection was fast and accurate.

Prabavathy et al. [151] proposed a novel fog computing based intrusion detection technique using Online Sequential Extreme Learning Machine (OS-ELM). The distributed security mechanism (guaranteed by the fog computing idea) respects interoperability, flexibility, scalability and heterogeneity aspects of IoT systems. The proposed system is composed of the following two major parts:

1) Attack detection at fog nodes: Prabavathy et al. use OS-ELM algorithm to detect intrusions in fog nodes. The IoT network is divided into virtual clusters where each cluster corresponds to a group of IoT devices under a single fog node. The OS-ELM classifies the incoming packets as normal or an attack. ELM is a single hidden layer feedforward neural network characterized by its fast learning phase. The input layer weights and hidden layer bias values are randomly selected to analytically deduce the output weights using simple matrix computations. However the online nature of OS-ELM favors a streaming detection of IoT attacks.
2) Summarization at cloud server: to have a general idea about the global security state of the IoT system, detected intrusions are sent from the fog node to the cloud server. After the analysis and the visualization of the current state, Prabavathy et al. propose two actions; i) predict next attacker action using the attacker plan recognition approach; or ii) identify fog node geographical position based multistage, and DDoS attacks. Hence, an intrusion response can be activated.

Prabavathy et al. proposed a proof of concept to evaluate their proposition on DUALCORE processor, 1 GB RAM and 200 GB HDD as fog nodes. Authors deployed Azure cloud service (4 X Dual-Core AMD Opteron 2218 @2.6 GHz, 8 core, 32 GB RAM, 6146 GB HDD) for experimental setup. They implemented OS-ELM using MATLAB and NSL-KDD as benchmark dataset. Authors claimed high accuracy and response time. They achieve 97.36% accuracy with reduced false alarm rate 0.37%. The detection rate with the fog node strategy was 25% faster when compared with cloud based implementation. An important advantage is that new online data can be incorporated in the learning process, which is not the case for ANN and NB.

Rathore et Park [16] deployed a novel fog detector using ELM-based Semi-supervised Fuzzy C-Means (ESFCM) NIDS. This distributed IDS handles geographically distributed and low-latency IoT detection for limited resource and network via fog computing. Supervised ML does not detect unknown attacks despite its good accuracy. Unsupervised ML has lower accuracy but, has the capability to detect unknown / zero-day attacks. Hence, Rathore et Park proposed a semi-supervised approach using the supervised and unsupervised ML for labeled and unlabeled inputs. For the unsupervised learning, Fuzzy C-Means (FCM) was the chosen algorithm (one of the widely used in clustering). FCM selects unlabeled data and assigns each input to one or more clusters with several degrees of membership. While the supervised part deploys Extreme Learning Machine (ELM) for effective and efficient detection. Hence, the authors proposed an ESFCM classification where Semi-supervised Fuzzy C-Means (SFCM) works with ELM classifier for a faster detection of known and unknown attacks. The IDS starts by generating a model (M) after having trained the ELM classifier on labeled dataset. Then, SFCM algorithm learns from both, labeled and unlabeled data to assign a degree of membership to the unlabeled inputs. The unlabeled instances that have better opportunity to belong to one class, are then classified using the trained model M and added to labeled data according to defined threshold. Moreover, the remaining unlabeled data are re-clustered with SFCM and retrained with ELM until all instances are assigned. Finally, a trained model is generated for labeled and unlabeled data.
Two types of evaluation for the proposed algorithm were established using the NSL-KDD dataset after scaling and preprocessing; i) a comparison between the authors' distributed solution and a centralized cloud-based framework; and ii) the effectiveness of the ESFCM was compared with traditional machine learning methods in terms of standard measures. Results show better performance of 11 ms in terms of detection time and 86.53% accuracy.

Moustafa et al. [19] proposed an ensemble network intrusion detection technique based on established statistical flow features to mitigate malicious events, particularly botnet attacks against DNS, HTTP and MQTT protocols utilized in IoT networks. Their solution can be divided into:

1) A set of features are extracted from the network traffic protocols MQTT, HTTP and DNS protocols via deep analysis of the TCP/IP model. Authors employ Bro-IDS tool for the basic features and develop a novel extractor module (which works simultaneously with Bro-IDS) to generate additional statistical features of the transactional flows.
2) A feature selection step where correlation coefficient is applied on result features to extract the most important ones. This step enables the reduction of computational cost of NIDS.
3) An ensemble method where the network data is distributed with AdaBoost algorithm. Then, Decision Tree (DT), Naive Bayes (NB) and Artificial Neural Network

(ANN) ML algorithms are deployed to detect attacks. The choice of the classification techniques is justified by calculating the correntropy measure. The AdaBoost (Adaptive Boosting) method enhances the performance of the detection compared to separate machine learning algorithms. It can deal with the small differences of the feature vectors via computing an error function. The error function is assigned to each instance of the distributed input data to learn and decide which learners can correctly classify each instance.

To extract the best features and evaluate the proposed ensemble technique, Moustafa et al. used the UNSW-NB15 and NIMS botnet datasets with simulated IoT sensor data. Experiments results have high detection rate (DR) and a low false positive rate(FPR) compared to existing state-of-the-art techniques. The ensemble strategy achieved between 95.25% and 99.86% DR and 0.01% to 0.72% FPR.

C. Comparison and Discussion

As presented in the previous Section, many researchers give a special interest for IoT powered NIDS via machine learning algorithms. A comparison between the previously detailed proposals is illustrated in Table V where we focus mainly on IDS deployment, detection methodology, used dataset, treated threats and ML used algorithms.

While reviewing [17], [141], [147], [148] and [149], we observe a lack of details on the **architecture deployments**. The above proposals concentrate on ML mechanisms for intrusion detection without discussing the architecture designs. Solutions of Hosseinpour et al. [18], Bostani and Sheikhan [37], [145], [146], Diro et al. [150], Prabavathy et al [151], Rathore and Park [16] and Moustafa et al. [19] deployed distributed architecture for intrusion detection, most suitable for IoT needs. In fact, IoT systems are distributed since they are composed of geographically distributed nodes. Such a criterion plays an important role when choosing the machine learning algorithm. Depending on where and how we want to deploy our NIDS, researchers have to pay attention and identify the algorithm for resource constrained smart objects. Hence, training an intensive algorithm on a limited node may not be feasible. However, the intensive task can be transferred to the cloud. The best strategy is to process resource consuming tasks in cloud/server part, and execute lightweight parts in the IoT edge. It is the case in fog computing based NIDS such as in [18], [37], [146], [150] and [151]. The proposed solutions take advantage of cloud layer for machine learning model training and use fog nodes for intrusion detection. Fog based intrusion detection enables co-ordination for better, low latency detection (near to the source of data). It reduces network bandwidth consumption since partial data is sent at cloud. Only some details are reported to the centralized, source-intensive part of the IoT system to summarize and detect distributed attacks. Fog concept enables autonomic and parallel distributed attack detection [150]. [16] and [18] claimed that, their proposals can be deployed in distributed IoT systems. Authors concentrated more on the distribution of traffic network data.

Concerning the **datasets**, recent information is necessary to train and evaluate IoT NIDS. Proposals such as [147], [148] and [150] are based on NSL-KDD dataset; non IoT dataset. The proposals neither support IoT protocols like 6LowPAN, Zigbee, CoAP, nor IoT architecture and principles like mobility and heterogeneity. Unfortunately, **no IoT-dedicated NIDS dataset** exist which explains the use of NSL-KDD. [17], [141] and [149] evaluate their proposal with their own data. However, [18], [37], [151], [16] and [19] used a combination of real and synthetic data or simulate the environment.

Moreover, most of the studied researches are made to protect IoT systems from precise **types of attacks** mainly DoS, U2R, R2L and Probe since they get inspired from NSL-KDD dataset. Thing is the only proposal which concentrate specially on IEEE 802.11 attacks. However, unsupervised and semi-supervised machine learning solutions are evaluated against specific attacks. However, they are able to detect unknown attacks such as in [18], [146], [148], [16].
The last important point to discuss is about **ML used algorithms**. Hodo et al. [17] use Multi-Layer Perceptron (MLP), a part of Artificial Neural Networks (ANN) family for off-line detection. From ANN, we move towards the solutions with deep machine learning (DL) proposed by [148], [150] and [149]. DL is deployed on multi-layer neural network. The good detection results of the proposals using DL as represented in Fig. 9 are due to i) training stability and generalization of DL; ii) its ability to reach a high accuracy rate if there is enough data and time [23]; iii) DL is a self-learning algorithm which means that it does not need manual feature engineering [16]; iv) DL extracts complex and non linear hierarchical features from training data of high dimension [150]. Lopez-Martin et al. [148] used conditional variational autoencoder algorithm (CVAE) which is a generative model based on variational autoencoder (VAE) concepts. CVAE relies on two inputs: i) the intrusion features and ii) the intrusion class labels, instead of using only the intrusion features as input as in VAE. CVAE is better in flexibility and performance. The authors chose CVAE for its ability in feature reconstruction, its ability to retrieve missing features from incomplete dataset. Despite their use for an unsupervised DL algorithm, they benefit from labeled data in training phase for a deviation-based NIDS. Martin-Lopez et al. ensured a good computational time with a good flexibility though generating one model from multiple trainings in only one single training step. They proved better accuracy 80% than linear SVM (75%), MLP (78%) and Random Forest (73%) algorithms. Meanwhile, Diro et al. used Multi-layer DL algorithm in a distributed strategy which gives better results compared to centralized DL (99% vs 96% of accuracy). It is true that training phase takes longer time, but real-time detection is faster and more accurate. Diro et al. [150] chose multi-layer DL since it is the most prevalent form of DL. It shows training stability with a significant scalability on big data concept. Moreover, Diro et al. compared DL with ML

TABLE V: Summary of NIDS for IoT based on Learning Techniques

References	IDS deployment	Detection Methodology	Used Dataset	Treated Threats	ML algorithms
Hodo et al. [17]	—	Anomaly-based	Simulation	DoS / DDoS	Multi-Layer Perceptron (MLP)
Nobakht et al. [141]	—	Anomaly-Host based	Real IoT devices (Hue lights)	Unauthorized access	Logistic Regression vs SVM
Hosseinpour et al. [18]	Distributed	Anomaly-based	KDD99 and SSH Brute Force from ISCX	Botnet attack	Artificial Immune System (AIS)
Bostani and Sheikhan [37], [145], [146]	Centralized / Distributed (Big Data architecture MapReduce)	Hybrid: Anomaly-based for the centralized part and specification-based for the distributed part [37]	Proprietary Simulator + NSL-KDD	Sinkhole / Selective Forwarding in 6LoWPAN and can be extended to Blackhole rank and Wormhole	Unsupervised Optimum Path Forest (OPF) in [37]
		Hybrid: Anomaly-based for the centralized part and misuse-based for the distributed part [145], [146]			Supervised & Unsupervised Optimum Path Forest (OPF) in [145], [146]
Pajouh et al. [147]	—	Anomaly-based	NSL-KDD	Low frequency attacks (such as U2R, R2L)	{Unsupervised Principal Component Analysis (PCA) + Supervised Linear Discriminant Analysis (LDA)} for Feature Reduction, & {Naive Bayes (NB) + Certainty Factor version of K Nearest Neighbors (CF-KNN)} for Classification
Lopez-Martin et al. [148]	—	Anomaly-based	NSL-KDD	DoS / R2L / U2R / Probe	Conditional Variational AutoEncoder (CVAE)
Thing [149]	—	Anomaly-based	Generated Dataset from lab SOHO	IEEE 802.11 attacks (flooding, injection and impersonation)	Stacked AutoEncoder (SAE)
Diro et al. [150]	Distributed	Anomaly-based	NSL-KDD	DoS / R2L.U2R / Probe	Multi-Layer Deep Learning
Prabavathy et al. [151]	Distributed	Anomaly-based	Emulation + NSL-KDD	Probe / R2L / U2R / DoS	Online Sequential Extreme Learning Machine (OS-ELM)
Rathore and Park [16]	Distributed	Anomaly-based	Simulation + NSL-KDD	Probe / R2L / U2R / DoS	ELM-based Semi-supervised Fuzzy C-Means (ESFCM)
Moustafa et al. [19]	Distributed	Anomaly-based	UNSW-NB15 + NIMS + Simulation	Botnet attack	AdaBoost ensemble method using three techniques of DT, NB and ANN

in the distributed context and proved that accuracy of the deep model is greater than that of shallow model (multi-class detection accuracy increase from 96.75% to 98.27%) and false alarms rate are lower for DL (from 4.97% for ML to 2.57% with DL in multi-class detection). Regarding the DL used version of Thing [149], he experimented Stacked Auto-Encoder (SAE) with two and three hidden layers but does not provide any primary choice arguments. SAE is a neural network built by stacking multiple layers of sparse auto-encoders. The output of each layer forms the input to the successive layer. Its hidden layers reduce the feature dimensionality and produce new set of features [152]. These new features are learned in cascade depths to improve precision. Nodes in the input and the output layer of SAE are the same [27]. The proposed solution achieved good accuracy results (98.66%) compared to J48 (an implementation of DT). The 2-hidden-layer model had a better performance over the 3-hidden-layer model.

This was about DL proposals. Another strategy in the use of ML algorithms that is taking more and more attention is the combination of different algorithms in the same system [147], [19], [16] and [145], [146]. Pajouh et al. [147] used two simple ML techniques which are Naive Bayes (NB) and K-nearest networks (KNN) for more exact class labels. NB is applied in the first place to identify anomalies. Then normal behaviors will be analyzed with KNN to refine normal instances. NB assumes the independence of all the characteristics of each sample in the given class label. It has the ability to measure good similarities of rare instances in the aim to handle imbalanced data. KNN uses a bucketing technique [153] to accelerate the classification task. On the other side, Pajouh et al. applied dimension reduction before running the classification. To do so, they deployed both Linear Discriminant Analysis (LDA) (a supervised

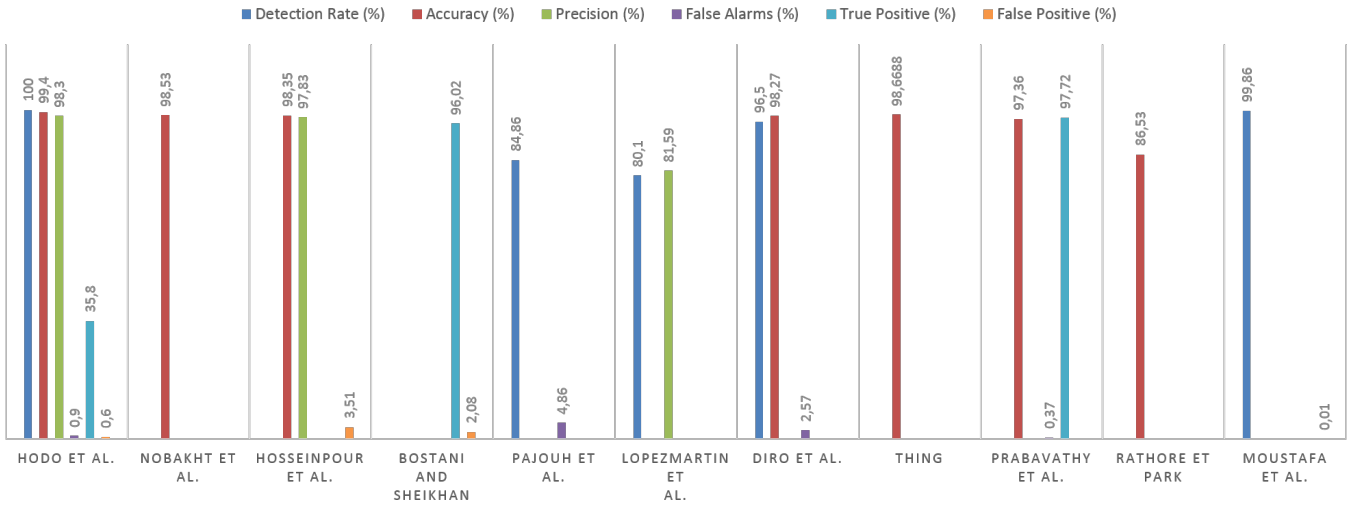

Fig. 9: State-of-art intrusion detection results

dimension reduction technique) and Principal Component Analysis (PCA) (an unsupervised dimension reduction technique). PCA provides a lower feature space by generating uncorrelated features from the initial correlated ones. LDA reduces the dimension of large working datasets by examining class labels. Hence, these two dimension reduction techniques represent a good strategy to i) reduce computational needs which is perfect for IoT systems and ii) fast the detection with less errors which is perfect for intrusion detection. Pajouh et al. achieved 84.86% of detection rate on NSL-KDD however Moustafa et al. [19] succeeded to have 99.86% which is an impressive value. Their idea is based on an AdaBoost ensemble learning method which uses three ML techniques, namely Decision Tree (DT), Naive Bayes (NB) and Artificial Neural Network (ANN). The combination of these algorithms is done in a distributed parallel way. Data is divided into N sets (according to an error function) and each data subset will be treated with a chosen algorithm to finally update the distribution. Such a logic is guaranteed thanks to AdaBoost flow. Moustafa et al. applied also feature selection before starting the classification. By analyzing the treated attacks using correntropy, authors noticed that there are small variations between legitimate and suspicious vectors. Thus the ML algorithms to be used, should classify these small differences. That's how DT, NB and ANN were chosen. DT [19] has multiple advantages while classifying network data. It selects important feature, prepare learning data points easily and manipulates directly the values of features. Even if a non-linear relations exist between parameters, DT performance is not affected. NB [19] is known for its good detection of abnormal inputs. It needs less training data and scales linearly predictors and features values. It is simple in parameters optimization. ANN [19] has many merits. It demands less formal statistical training and defines complex non-linear correlations between dependent and independent variables. Furthermore, it enables the detection of all possible interactions between predictors and variables. The third type of ML algorithms combination is presented in [16]. Rathore and Park integrated a Semi-supervised Fuzzy C-Means algorithm (SFCM) with the Extreme Learning Machine (ELM) classifier to compose ELM-based Semi-Supervised Fuzzy C-Means (ESFCM) method. SFCM is based on the unsupervised Fuzzy C-Means (FCM) algorithm which clusters inputs data. It is one of the widely used techniques in unsupervised learning. It captures hidden and visible data structures. However, ELM algorithm [154] is originally created to train single hidden-layer feedforward neural networks (SLFNs). ELM is efficient and doted of fast learning capacity in highly dynamic environment like IoT systems. Consequently, Rathore and Park achieved faster detection (11ms) with a better accuracy rate 86.53% comparing to traditional ML in their framework with the advantage of labeled and unlabeled data classification. Prabavathy et al. [151] took advantage of ELM algorithm in their intrusion detection proposal. They operated an online version of ELM (OS-ELM) for a real time analysis. Compared to ANN and NB, authors achieved better accuracy (97.36%) with lower false positive rate (0.37%) in a lower period of time (25% faster). A major advantage of OS-ELM is that it can incorporate new data online for learning which is not possible with the other compared algorithms.

About Sheikhan and Bostani solutions, they used in their works [37], [145] and [146] mainly Optimum-Path Forest algorithm (OPF) which is an efficient graph-based ML. They used two variants of OPF; i) OPFC (OPF Clustering) which is an unsupervised ML and ii) MOPF (Modified OPF) which is a supervised algorithm. OPF main strength [155] is that it does not make any assumption about the shape of classes. The authors use OPFC to project clustering models on a MapReduce architecture. MOPF is used in a misuse-based detection engine with a feature selection module. The proposed solutions are simple and fast classifiers, they are parameter independent and originally support multi-class problems [155].

Moreover, Nobakht et al. [141] executed a feature reduction heuristically and experimented two ML algorithms; Logistic

Regression (LR) and SVM for intrusion detection. LR is gradient descent which aims to find out the optimal parameters of a LR model. The accuracy of the obtained linear model with LR was less interesting than the non linear model of SVM (96.2% for LR whereas SVM achieves 100%).
Finally, Hosseinpour et al. [18] used Artificial Immune System (AIS) which is an unsupervised ML algorithm that is inspired from human immune system. It is characterized by a multi-layered protection structure. First line of defense responses immediately to previously seen problems then a non specific protection for unknown attacks is processed. It does not need prior knowledge of specific outsiders. Another important point for AIS is the memory aspect; AIS is efficient in unknown attacks detection. Authors achieve 98.35% of accuracy and 97.83% of precision which are remarkable results as noticed in Fig. 9. However, AIS training needs resource which is why Hosseinpour et al. proceed it in the cloud layer.

As seen previously, many ML works for IoT networks intrusion detection was developed. Each state-of-art proposal has its arguments, its advantages and its drawbacks depending on the IDS chosen architecture; if it is centralized or distributed or the two combined. The detection strategy in terms of using only anomaly detection or combine it with signature based detection plays also a role. Moreover, each researcher study is based on a dataset since ML techniques are constructed on the base of data. Datasets can be labeled or unlabeled, from an on-line system or a pre-existing repository. Furthermore, different ML algorithms were experimented; from supervised to unsupervised, to semi-supervised strategies. Algorithms were deployed in a standalone or combined. And even combination is either in a parallel or in cascade. Many combinations are possible and each one give different results. Comparing the NIDS of the state-of-art is hard since each one treat special attacks, in a special architecture, with different dataset, using various ML algorithms in different strategies. To have a graphical representation of the detection efficiency in IoT NIDS based on ML, we present an histogram in Fig. 9 where we tend towards summarizing the performances and not really compare one-to-one results because of differences in deployed strategies. It is true that the IoT environment is conditioned by many challenges as presented in Section II e.g. resource constraints, hence the application of ML algorithms in IoT is not always obvious. For example, training ML models may be a very computationally-intensive task for small IoT devices. On the one hand, training the model on a server will help with lower-power devices problems. On the other hand, this solution will require transferring all the data collected on the local device to the external server for processing which puts us in face of the limited connectivity of low-power devices. However, as shown in Section V, ML strategies overcome such IoT/ML problems and achieve interesting results in IoT NIDS. Consequently, while putting in place a NIDS for IoT, a good strategy to overcome IoT, ML and security limits should be well studied. Intensive processes must be executed on powerful devices, and the opposite type have to be placed in small devices. Sent data between the different layers of IoT (Section II) should be well chosen. Data need to be pre-processed to accelerate results, and increase the precision of detection rates while decreasing false alarms. Unneeded treatments must be removed. Features dependence/independence as well as data imbalance have to be deeply studied and IoT characteristics should be exploited; e.g. the IoT distributed aspect. As shown in this section, many ML solutions are possible, researchers have just to make the right choices in terms of deployment, detection methodology, used datasets and developed ML algorithms. In the next section, we will point out possible future research directions.

VI. Future research directions

With the explosion of IoT, two new paradigms come out: **edge computing and fog computing**. Both of them tend to push intelligence and processing logic employment down near to data sources (which means as close as possible to sensors and actuators) to reduce the network bandwidth needed to communicate data from the perception layer to data-centers where analytics are usually processed. The main difference between edge and fog architecture lies in the place where the intelligent processing and the computing power are located. Edge computing pushes them to the extremes of the network such as edge gateways and devices (e.g. Programmable Automation Controllers PACs). However, fog computing tends to place them in the local area network level of the network architecture which means in hubs, routers or gateways (fog nodes). These two concepts should be deeply explored and exploited for future IoT IDS architecture. They enable the intrusion detection process to be distributed. Consequently, this strategy should enable intrusion detection with less resource needs which is suitable for IoT. As an example, Al-Turjman proposes in [156] a low cache replacement approach based on fog computing logic. He retains the cache value of active sensor nodes in SDN for a longer duration and improves network efficacy. In the same vein, **Big Data** [133] is a solution to remedy problems related to the big volume of network traffic generated by IoT networks. So as future works in IDS architecture deployment, edge and fog computing as well as Big Data methods should be deeply explored for IoT NIDS with paying more attention on protecting IDSs themselves in case of IoT system fall.

Furthermore, IoT NIDS needs a **real-world IoT-dedicated dataset**. A common real-world dedicated dataset would help with a real, efficient comparison between the different researches. A dataset benchmark enables training, validating and evaluating studies with different ML algorithms.

Besides, according to Sommer and Paxson [14], IDS based on learning techniques suffer from “a semantic gap between results and their operational interpretation”. Unfortunately, IDS based on learning techniques are usually evaluated with rates such as accuracy, false positive and false negative. We believe that presenting just these metrics is not sufficient. Researchers should interpret the results and understand semantics of the

features choice and the detection process. Semantic would also help to differentiate between abnormal and malicious behaviors. Therefore, **semantic relation between detection and learning process** seems to be an interesting track to explore.

Moreover, **features choices** as well as **features reconstruction and features dimension reduction** can be more inspected for IoT NIDS based on learning techniques. Such techniques can help overcoming IoT resource constraints challenges. **Deep learning techniques** used alone or combined should be also more experienced since algorithms like auto-encoders are efficient in features reconstruction and dimension reduction.

In addition, techniques like **software accelerator**, for low-powering the learning algorithms on tiny devices, could be experienced in IoT security environment like in [157]. Nicholas D. Lane en al. designed and implemented DeepX, a software accelerator for deep learning execution DeepX that lowers significantly the device resources (viz. memory, computation, energy) required by deep learning. Security researchers can get inspired from works such as Ravi et al. in [158] where they presented an optimization approach to enable the use of real-time deep learning in low-power devices. The authors used a spectrogram representation of the inertial input data to provide invariance against changes in sensor placement, amplitude, or sampling rate, thus allowing a more compact method design. The same authors proposed in [159] a combination of shallow learned features from a deep learning approach to enable accurate and real-time activity classification. Such a proposal overcomes some limitations for deep learning when on-node computation is needed.

Last but not least, for the future, more efforts need to be made to detect **unknown and zero-day attacks** in IoT networks and develop IDSs that can automatically update the list of the considered attacks when new ones appear. IoT NIDS need to be experienced with ML algorithms and big data [160], [133] strategies to update their training model in **real-time**, in a **streaming detection**. For example, **Incremental ML** field in the intrusion detection should be experienced. Incremental learning is about retraining the model on both previously seen and unseen data to construct new models. It aims to ensure continuity in the learning process through regular model update based only on the new available batch of data. This idea joins the approach of making IDSs more intelligent and human-independent in decision making.

Finally, IoT is being deployed increasingly in industrial systems, military operations, health-care environment and many other sensitive areas that are cognitive based human-centric IoT. Sensitive data and private information are exchanged between the travelling objects in a context that puts people's lives at risk on the one side and where human behaviors affect the IoT systems in the other side. Hence, more security attention needs to be paid to these IoT human-based systems.

VII. Conclusion

Connected Things (IoTs) have become pervasive for every individual. In fact, the IoT benefits has human life evolving with the Things. IoT is in smart cities (e.g. smart parking), smart environment (e.g. for air pollution), in smart metering (e.g. smart grid), in industrial control (e.g. for vehicle auto-diagnosis), etc. They are in every domain even in critical ones like military, health care and buildings security. Unfortunately, industries are focusing on innovating and developing more connected products without verifying too much their quality and security. At this stage, we remark that IoT is a double-edged weapon. This army of connected devices can be hacked and used against humanity. One compromised node can affect the whole IoT network. A malicious user becomes able to break down home automation systems and so steals them or can remotely control vehicles to hit innocent people in roads.

One of the powerful mechanisms to ensure IoT network security are NIDSs. They help detecting intrusions in systems. To enhance their efficiency, they are becoming provided with learning techniques.

To the best of our knowledge, our survey is the first proposal with comprehensive discussion of learning based NIDSs for IoT systems. In this paper the field of IoT security has been introduced with a comparison between previous surveys. Moreover, IoT threats and detection techniques over traditional defense mechanisms have been classified. Then, a comprehensive evaluation of NIDS implementation tools have been presented; starting with free network datasets, to free and open-source network sniffers, to open-source NIDS that can be used by researchers and industrials to implement and evaluate their own sophisticated NIDS solution. Furthermore, an overview about NIDS in IoT systems has been given with a focus on their architecture, deployments, detection methodologies and treated threats. The pros and cons of each proposal is thoroughly evaluated. Last but not least, we continued with learning NIDSs for IoT eco-system where, learning terminologies have been introduced and the state-of-art of IoT learning NIDS has been detailed. Each work has been summarized separately; then, adopted strategies have been compared to come up with strengths and tactics ideal for ML and non-ML NIDS. The State-of-art shows interesting results; up to 99% detection accuracy and 0.01% false positive. Finally, top IoT NIDS proposals have been compared with a focus on ML algorithms and future research directions have been detailed.

In the coming time, IoT based solutions will explode. We believe that one of the most important needs to deal with is the validation strategy improvement; more specifically, the development of a public benchmark dataset for network exchanges of IoT systems. It should include different IoT protocols with the different IoT threats. This dataset would enable a clear, practical and convenient comparison of the different developed NIDS. Furthermore, it is also important to concentrate on developing IoT NIDS that detect known and unknown attacks without being protocol-dependent. To conclude, a combination of edge and fog computing approaches could be more and more explored for IoT NIDS architectures. These approaches enable IoT intrusion detection with less resource consumption, thus, with respect to IoT challenges.

REFERENCES

[1] J. Rifkin, "The Zero Marginal Cost Society: The Internet of Things, the Collaborative Commons, and the Eclipse of Capitalism: Book," Apr. 2014.

[2] A. Grau, "The Internet of Secure Things What is Really Needed to Secure the Internet of Things? | Icon Labs," Mar. 2014. [Online]. Available: http://www.iconlabs.com/prod/internet-secure-things-%E2%80%93-what-really-needed-secure-internet-things

[3] U. N. IDC, Intel, "A Guide to the Internet of Things Infographic," Feb. 2015. [Online]. Available: https://www.intel.com/content/www/us/en/internet-of-things/infographics/guide-to-iot.html

[4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X13000241

[5] D. Singh, G. Tripathi, and A. J. Jara, "A survey of Internet-of-Things: Future vision, architecture, challenges and services," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, Mar. 2014, pp. 287–292.

[6] O. Vermesan and P. Friess, "Internet of Things Applications - From Research and Innovation to Market Deployment Book," *River Publishers*, Jun. 2014. [Online]. Available: http://www.internet-of-things-research.eu/pdf/IERC_Cluster_Book_2014_Ch.3_SRIA_WEB.pdf

[7] R. Mitchell and I. R. Chen, "A Survey of Intrusion Detection Techniques for Cyber-physical Systems," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 55:1–55:29, Mar. 2014. [Online]. Available: http://doi.acm.org/10.1145/2542049

[8] E. Benkhelifa, T. Welsh, and W. Hamouda, "A Critical Review of Practices and Challenges in Intrusion Detection Systems for IoT: Towards Universal and Resilient Systems," *IEEE Communications Surveys Tutorials*, pp. 1–1, Jun. 2018.

[9] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2661–2674, Nov. 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870513001005

[10] E. Bertino and N. Islam, "Botnets and Internet of Things Security," *Computer*, vol. 50, no. 2, pp. 76–79, Feb. 2017.

[11] J. P. Anderson, "Computer security threat monitoring and surveillance," James P. Anderson Company, Technical Report, 1980.

[12] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits, "Denial-of-Service detection in 6lowpan based internet of things," in *IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications*, 2013, pp. pp. 600–607.

[13] Z. M. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "State-of-the-Art Deep Learning: Evolving Machine Intelligence Toward Tomorrow's Intelligent Network Traffic Control Systems," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2432–2455, 2017.

[14] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in *2010 IEEE Symposium on Security and Privacy*, May 2010, pp. 305–316.

[15] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey," *arXiv:1701.02145 [cs]*, Jan. 2017, arXiv: 1701.02145. [Online]. Available: http://arxiv.org/abs/1701.02145

[16] S. Rathore and J. H. Park, "Semi-supervised learning based distributed attack detection framework for IoT," *Applied Soft Computing*, vol. 72, pp. 79–89, Nov. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494618303508

[17] 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 *2016 International Symposium on Networks, Computers and Communications (ISNCC)*, May 2016, pp. 1–6.

[18] F. Hosseinpour, P. Vahdani Amoli, J. Plosila, T. Hmlinen, and H. Tenhunen, "An Intrusion Detection System for Fog Computing and IoT based Logistic Systems using a Smart Data Approach," *International Journal of Digital Content Technology and its Applications*, vol. 10, Dec. 2016. [Online]. Available: https://jyx.jyu.fi/handle/123456789/54088

[19] N. Moustafa, B. Turnbull, and K. R. Choo, "An Ensemble Intrusion Detection Technique based on proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things," *IEEE Internet of Things Journal*, pp. 1–1, Sep. 2018.

[20] F. Chen, P. Deng, J. Wan, D. Zhang, A. V. Vasilakos, and X. Rong, "Data Mining for the Internet of Things: Literature Review and Challenges," *International Journal of Distributed Sensor Networks*, vol. 11, no. 8, p. 431047, Aug. 2015. [Online]. Available: https://doi.org/10.1155/2015/431047

[21] C. W. Tsai, C. F. Lai, M. C. Chiang, and L. T. Yang, "Data Mining for Internet of Things: A Survey," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 77–97, Jan. 2014.

[22] L. Cui, S. Yang, F. Chen, Z. Ming, N. Lu, and J. Qin, "A survey on application of machine learning for Internet of Things," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 8, pp. 1399–1417, Aug. 2018. [Online]. Available: https://doi.org/10.1007/s13042-018-0834-5

[23] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine learning for internet of things data analysis: a survey," *Digital Communications and Networks*, vol. 4, no. 3, pp. 161–175, Aug. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S235286481730247X

[24] S. Agrawal and J. Agrawal, "Survey on Anomaly Detection using Data Mining Techniques," *Procedia Computer Science*, vol. 60, pp. 708–713, Jan. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050915023479

[25] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Communications Surveys Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[26] L. Wang and R. Jones, "Big Data Analytics for Network Intrusion Detection: A Survey," *International Journal of Networks and Communications*, vol. 7, no. 1, pp. 24–31, 2017.

[27] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, "A Detailed Investigation and Analysis of using Machine Learning Techniques for Intrusion Detection," *IEEE Communications Surveys Tutorials*, pp. 1–1, Jun. 2018.

[28] B. B. Zarpelo, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, Apr. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804517300802

[29] S. Krushang and H. Upadhyay, "A Survey: DDOS Attack on Internet of Things," *International Journal of Engineering Research and Development*, vol. Volume 10, no. Issue 11, pp. 58–63, Nov. 2014. [Online]. Available: www.ijerd.com

[30] I. B. Ida, A. Jemai, and A. Loukil, "A survey on security of IoT in the context of eHealth and clouds," in *2016 11th International Design Test Symposium (IDT)*, Hammamet, Tunisia, Dec. 2016, pp. 25–30.

[31] S. Babar, A. Stango, N. Prasad, J. Sen, and R. Prasad, "Proposed embedded security framework for Internet of Things (IoT)," in *2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (Wireless VITAE)*, Feb. 2011, pp. 1–5.

[32] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in Cloud," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 42–57, Jan. 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804512001178

[33] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, "Intrusion detection techniques in cloud environment: A survey," *Journal of Network and Computer Applications*, vol. 77, pp. 18–47, Jan. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804516302417

[34] A. W. Atamli and A. Martin, "Threat-Based Security Analysis for the Internet of Things," in *2014 International Workshop on Secure Internet of Things*, Sep. 2014, pp. 35–43.

[35] M. Conti, N. Dragoni, and V. Lesyk, "A Survey of Man In The Middle Attacks," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 2027–2051, 2016.

[36] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," in *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, 2003.*, May 2003, pp. 113–127.

[37] H. Bostani and M. Sheikhan, "Hybrid of anomaly-based and specification-based IDS for Internet of Things using unsupervised OPF based on MapReduce approach," *Computer Communications*, vol. 98, no. Supplement C, pp. 52–71, Jan. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366416306387

[38] L. Wallgren, S. Raza, and T. Voigt, "Routing Attacks and Countermeasures in the RPL-Based Internet of Things," *International*

Journal of Distributed Sensor Networks, vol. 9, no. 8, p. 794326, Aug. 2013. [Online]. Available: https://doi.org/10.1155/2013/794326

[39] Microsoft, "The STRIDE Threat Model," 2005. [Online]. Available: https://msdn.microsoft.com/fr-fr/en-en/enus/library/ee823878(v=cs.20).aspx

[40] N. Arajo, R. d. Oliveira, E. Ferreira, A. A. Shinoda, and B. Bhargava, "Identifying important characteristics in the KDD99 intrusion detection dataset by feature selection using a hybrid approach," in *2010 17th International Conference on Telecommunications*, Apr. 2010, pp. 552–558.

[41] E. Bou-Harb, M. Debbabi, and C. Assi, "Cyber Scanning: A Comprehensive Survey," *IEEE Communications Surveys Tutorials*, vol. 16, no. 3, pp. 1496–1519, 2014.

[42] S. Anwar, Z. Inayat, M. F. Zolkipli, J. M. Zain, A. Gani, N. B. Anuar, M. K. Khan, and V. Chang, "Cross-VM cache-based side channel attacks and proposed prevention mechanisms: A survey," *Journal of Network and Computer Applications*, vol. 93, pp. 259–279, Sep. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804517302205

[43] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, "Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 184–208, 2016.

[44] V. Zlomisli, K. Fertalj, and V. Sruk, "Denial of service attacks, defences and research challenges," *Cluster Computing*, vol. 20, no. 1, pp. 661–671, Mar. 2017. [Online]. Available: https://link.springer.com/article/10.1007/s10586-017-0730-x

[45] B. Prabadevi and N. Jeyanthi, "Distributed Denial of service attacks and its effects on Cloud environment- a survey," in *The 2014 International Symposium on Networks, Computers and Communications*, Jun. 2014, pp. 1–5.

[46] Cisco, "A Cisco Guide to Defending Against Distributed Denial of Service Attacks," Oct. 2012. [Online]. Available: http://www.cisco.com/c/en/us/about/security-center/guide-ddos-defense.html

[47] K. Hengst, "DDoS through the Internet of Things An analysis determining the potential power of a DDoS attack using IoT devices," Jul. 2016. [Online]. Available: http://referaat.cs.utwente.nl/

[48] C. Rossow, "Amplification hell: Revisiting network protocols for DDoS abuse," *Network and Distributed System Security Symposium*, Feb. 2014. [Online]. Available: https://dud.inf.tu-dresden.de/~strufe/rn_lit/rossow14amplification.pdf

[49] S. Liron, "Mirai: The IoT Bot that Took Down Krebs and Launched a Tbps Attack on OVH," Oct. 2016. [Online]. Available: https://f5.com/labs/articles/threat-intelligence/ddos/mirai-the-iot-bot-that-took-down-krebs-and-launched-a-tbps-attack-on-ovh-22422

[50] D. Holmes, "What's the Fix for IoT DDoS Attacks? | SecurityWeek.Com," Oct. 2016. [Online]. Available: http://www.securityweek.com/whats-fix-iot-ddos-attacks

[51] S. Hettich and S. Bay, "KDD Cup 1999 Data - The UCI KDD Archive. Irvine, CA: University of California, Department of Information and Computer Science." 1999. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[52] M. R. Asghar, G. Dn, D. Miorandi, and I. Chlamtac, "Smart Meter Data Privacy: A Survey," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2820–2835, 2017.

[53] S. Game and C. Raut, "Protocols for detection of node replication attack on wireless sensor network," *Journal of Computer Engineering*, vol. 16, no. 1, pp. 01–11, Jan. 2014. [Online]. Available: http://www.iosrjournals.org/iosr-jce/papers/Vol16-issue1/Version-2/A016120111.pdf?id=8539

[54] L. Sujihelen, C. Jayakumar, and C. S. Singh, "Detecting Node Replication Attacks in Wireless Sensor Networks: Survey," *Indian Journal of Science and Technology*, vol. 8, no. 16, Jul. 2015. [Online]. Available: http://www.indjst.org/index.php/indjst/article/view/54150

[55] W. Ben Jaballah, M. Conti, G. Fil, M. Mosbah, and A. Zemmari, "Whac-A-Mole: Smart node positioning in clone attack in wireless sensor networks," *Computer Communications*, vol. 119, pp. 66–82, Apr. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366416307381

[56] Z. Zhang, M. C. Y. Cho, C. Wang, C. Hsu, C. Chen, and S. Shieh, "IoT Security: Ongoing Challenges and Research Opportunities," in *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications*, Nov. 2014, pp. 230–234.

[57] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues," *IEEE Communications Surveys Tutorials*, vol. 17, no. 3, pp. 1294–1312, 2015.

[58] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[59] F. Al-Turjman, "QoSaware data delivery framework for safety-inspired multimedia in integrated vehicular-IoT," *Computer Communications*, vol. 121, pp. 33–43, May 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366417306060

[60] S. A. Alabady, F. Al-Turjman, and S. Din, "A Novel Security Model for Cooperative Virtual Networks in the IoT Era," *International Journal of Parallel Programming*, Jul. 2018. [Online]. Available: https://doi.org/10.1007/s10766-018-0580-z

[61] S. A. Alabady and F. Al-Turjman, "Low Complexity Parity Check Code for Futuristic Wireless Networks Applications," *IEEE Access*, vol. 6, pp. 18 398–18 407, Apr. 2018.

[62] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," *Computer Networks*, vol. 76, pp. 146–164, Jan. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128614003971

[63] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A Survey on Security and Privacy Issues in Internet-of-Things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1250–1258, Oct. 2017.

[64] A. K. Sikder, H. Aksu, and A. S. Uluagac, "6thsense: A Context-aware Sensor-based Attack Detector for Smart Devices," *26th USENIX Security Symposium (USENIX Security 17)*, p. 19, Aug. 2017.

[65] P. Faruki, V. Ganmoor, V. Laxmi, M. S. Gaur, and A. Bharmal, "AndroSimilar: Robust Statistical Feature Signature for Android Malware Detection," in *Proceedings of the 6th International Conference on Security of Information and Networks*, ser. SIN '13. New York, NY, USA: ACM, Nov. 2013, pp. 152–159. [Online]. Available: http://doi.acm.org/10.1145/2523514.2523539

[66] P. Faruki, A. Bharmal, V. Laxmi, V. Ganmoor, M. S. Gaur, M. Conti, and M. Rajarajan, "Android Security: A Survey of Issues, Malware Penetration, and Defenses," *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 998–1022, 2015.

[67] R. E. Crossler, F. Blanger, and D. Ormond, "The quest for complete security: An empirical analysis of users multi-layered protection from security threats," *Information Systems Frontiers*, Apr. 2017. [Online]. Available: https://doi.org/10.1007/s10796-017-9755-1

[68] M. Tanase, "IP Spoofing: An Introduction | Symantec Connect Community," Mar. 2003. [Online]. Available: https://www.symantec.com/connect/articles/ip-spoofing-introduction

[69] F. Al-Turjman and S. Alturjman, "Confidential smart-sensing framework in the IoT era," *The Journal of Supercomputing*, vol. 74, no. 10, pp. 5187–5198, Oct. 2018. [Online]. Available: https://doi.org/10.1007/s11227-018-2524-1

[70] F. Al-Turjman, Y. K. Ever, E. Ever, H. X. Nguyen, and D. B. David, "Seamless Key Agreement Framework for Mobile-Sink in IoT Based Cloud-Centric Secured Public Safety Sensor Networks," *IEEE Access*, vol. 5, pp. 24 617–24 631, Oct. 2017.

[71] F. Al-Turjman and S. Alturjman, "Context-Sensitive Access in Industrial Internet of Things (IIoT) Healthcare Applications," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2736–2744, Jun. 2018.

[72] S. Patil, P. Kulkarni, P. Rane, and B. Meshram, "IDS vs IPS," *International Journal of Computer Networks and Wireless Communications*, vol. V 2, no. Issue 1, 2012. [Online]. Available: http://www.ijcnwc.org/papers/vol2no12012/16vol2no1.pdf

[73] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, Jan. 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804515002891

[74] W. Haider, J. Hu, J. Slay, B. P. Turnbull, and Y. Xie, "Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling," *Journal of Network and Computer Applications*, vol. 87, pp. 185–192, Jun. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804517301273

[75] P. Casas, J. Mazel, and P. Owezarski, "Unsupervised Network Intrusion Detection Systems: Detecting the Unknown without Knowledge," *Computer Communications*, vol. 35, no. 7, pp. 772–783, Apr. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366412000266

[76] S. Sharma and M. Dixit, "A Review on Network Intrusion Detection System Using Open Source Snort," *International Journal of Database Theory and Application*, vol. 9, no. 4, pp. 61–70, Apr. 2016. [Online]. Available: http://www.earticle.net/article.aspx?sn=272711

[77] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Nov. 2015, pp. 1–6.

[78] C. Xiang and S. M. Lim, "Design of Multiple-Level Hybrid Classifier for Intrusion Detection System," in *2005 IEEE Workshop on Machine Learning for Signal Processing*, Sep. 2005, pp. 117–122.

[79] N. Chandolikar and V. Nandavadekar, "Selection of Relevant Feature for Intrusion Attack Classification by Analyzing KDD Cup 99," *MIT International Journal of Computer Science & Information Technology*, vol. 2, no. 2, pp. 85–90, Aug. 2012.

[80] N. Kayacik and M. Heywood, "Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets," in *The 3rd Annual Conference on Privacy, Security and Trust (PST)*, 2005.

[81] S. K. Sahu, S. Sarangi, and S. K. Jena, "A detail analysis on intrusion detection datasets," in *2014 IEEE International Advance Computing Conference (IACC)*, Feb. 2014, pp. 1348–1353.

[82] H. Nguyen, K. Franke, and S. Petrovic, "Improving Effectiveness of Intrusion Detection by Correlation Feature Selection," in *2010 International Conference on Availability, Reliability and Security*, Feb. 2010, pp. 17–24.

[83] A. O. Adetunmbi, S. O. Adeola, and O. A. Daramola, "Analysis of KDD 99 Intrusion Detection Dataset for Selection of Relevance Features," in *Proceedings of the World Congress on Engineering and Computer Science*, vol. 1, San Francisco, USA, Oct. 2010.

[84] P. Gogoi, M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Packet and Flow Based Network Intrusion Dataset," in *Contemporary Computing*, ser. Communications in Computer and Information Science. Springer, Berlin, Heidelberg, Aug. 2012, pp. 322–334. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-32129-0_34

[85] A. R. Vasudevan, E. Harshini, and S. Selvakumar, "SSENet-2011: A Network Intrusion Detection System dataset and its comparison with KDD CUP 99 dataset," in *2011 Second Asian Himalayas International Conference on Internet (AH-ICI)*, Nov. 2011, pp. 1–5.

[86] M. V. Mahoney and P. K. Chan, "An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection," in *Recent Advances in Intrusion Detection*, ser. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Sep. 2003, pp. 220–237. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-45248-5_13

[87] "NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB," 2016. [Online]. Available: http://www.unb.ca/cic/research/datasets/nsl.html

[88] 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 *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, May 2017, pp. 559–564.

[89] C. I. for Cybersecurity (CIC), "IDS 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB," 2017. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html

[90] "CSE-CIC-IDS2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB," 2018. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2018.html

[91] tcpdump, "Tcpdump/Libpcap public repository," 2017. [Online]. Available: http://www.tcpdump.org

[92] A. Sivanathan, A. Hamza, H. Habibi, and V. Sivaraman, "UNSW Proliferation Dataset." [Online]. Available: https://iotanalytics.unsw.edu.au/index

[93] A. Habibi Lashkari, G. Draper Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of Tor Traffic using Time based Features:," in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy*. Porto, Portugal: SCITEPRESS - Science and Technology Publications, 2017, pp. 253–262. [Online]. Available: http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0006105602530262

[94] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization:," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*. Funchal, Madeira, Portugal: SCITEPRESS - Science and Technology Publications, 2018, pp. 108–116. [Online]. Available: http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0006639801080116

[95] A. Gharib, I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "An Evaluation Framework for Intrusion Detection Dataset," in *2016 International Conference on Information Science and Security (ICISS)*, Dec. 2016, pp. 1–6.

[96] N. Hoque, M. H. Bhuyan, R. C. Baishya, D. K. Bhattacharyya, and J. K. Kalita, "Network attacks: Taxonomy, tools and systems," *Journal of Network and Computer Applications*, vol. 40, pp. 307–324, Apr. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804513001756

[97] F. Fuentes and D. C. Kar, "Ethereal vs. Tcpdump: A Comparative Study on Packet Sniffing Tools for Educational Purpose," *J. Comput. Sci. Coll.*, vol. 20, no. 4, pp. 169–176, Apr. 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1047846.1047873

[98] P. Asrodia and H. Patel, "Analysis of various packet sniffing tools for network monitoring and analysis," *International Journal of Electrical, Electronics and Computer Engineering*, vol. 1, no. 1, pp. 55–58, 2012.

[99] "Ettercap Home Page." [Online]. Available: https://www.ettercap-project.org/

[100] "ARGUS- Auditing Network Activity," 2017. [Online]. Available: https://qosient.com/argus/

[101] "EtherApe, a graphical network monitor." [Online]. Available: https://etherape.sourceforge.io/

[102] "Wireshark Go Deep." [Online]. Available: https://www.wireshark.org/

[103] J. Schreiber, "Open Source Intrusion Detection Tools: A Quick Overview," Jan. 2014. [Online]. Available: https://www.alienvault.com/blogs/security-essentials/open-source-intrusion-detection-tools-a-quick-overview

[104] M. Roesch, "Snort Lightweight Intrusion Detection for Networks," p. 11, 1999.

[105] B. Cusack and M. Alqahtani, "Acquisition Of Evidence From Network Intrusion Detection Systems," in *Australian Digital Forensics Conference*, Dec. 2013. [Online]. Available: http://ro.ecu.edu.au/adf/118

[106] K. Thongkanchorn, S. Ngamsuriyaroj, and V. Visoottiviseth, "Evaluation studies of three intrusion detection systems under various attacks and rule sets," in *2013 IEEE International Conference of IEEE Region 10 (TENCON 2013)*, Oct. 2013, pp. 1–4.

[107] "Kismet Wireless," 2018. [Online]. Available: https://www.kismetwireless.net/index.shtml

[108] "OpenWIPS-ng," 2012. [Online]. Available: http://openwips-ng.org/index.html

[109] "Security Onion," 2018. [Online]. Available: https://securityonion.net/

[110] "Sagan Solution, Managed SIEM, Log Analysis, Network Analysis, Monitoring, Alerting, MSSP | Quadrant Information Security," 2018. [Online]. Available: https://quadrantsec.com/sagan_solution/

[111] S. Team, "Snort - Network Intrusion Detection & Prevention System," 2017. [Online]. Available: https://www.snort.org/

[112] T. O. I. S. Foundation, "Suricata," 2017. [Online]. Available: https://suricata-ids.org/

[113] P. Vern, "Bro: A System for Detecting Network Intruders in Real-Time," *Computer Networks*, vol. 31, no. 23-24, pp. 2435–2463, 1999. [Online]. Available: http://www.icir.org/vern/papers/bro-CN99.pdf

[114] C. Kolias, V. Kolias, and G. Kambourakis, "TermID: a distributed swarm intelligence-based approach for wireless intrusion detection," *International Journal of Information Security*, vol. 16, no. 4, pp. 401–416, Aug. 2017. [Online]. Available: https://doi.org/10.1007/s10207-016-0335-z

[115] Y. Zhang, W. Lee, and Y. A. Huang, "Intrusion detection techniques for mobile wireless networks," *Wireless Networks*, vol. 9, no. 5, pp. 545–556, Jan. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=942545.942556

[116] A. Abduvaliyev, A. K. Pathan, J. Zhou, R. Roman, and W. Wong, "On the Vital Areas of Intrusion Detection Systems in Wireless Sensor Networks," *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1223–1237, 2013.

[117] I. Butun, S. D. Morgera, and R. Sankar, "A Survey of Intrusion Detection Systems in Wireless Sensor Networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 266–282, 2014.

[118] O. Can and O. K. Sahingoz, "A survey of intrusion detection systems in wireless sensor networks," in *2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*, May 2015, pp. 1–6.

[119] A. Aris and S. F. Oktug, "Poster: State of the Art IDS Design for IoT," in *Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks*, ser. EWSN 17. USA: Junction Publishing, Feb. 2017, pp. 196–197. [Online]. Available: http://dl.acm.org/citation.cfm?id=3108009.3108037

[120] P. Kasinathan, G. Costamagna, H. Khaleel, C. Pastrone, and M. A. Spirito, "DEMO: An IDS Framework for Internet of Things Empowered by 6lowpan," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, ser. CCS '13. New York, NY, USA: ACM, Nov. 2013, pp. 1337–1340. [Online]. Available: http://doi.acm.org/10.1145/2508859.2512494

[121] "Metasploit | Penetration Testing Software, Pen Testing Security," 2017. [Online]. Available: https://www.metasploit.com/

[122] C. Jun and C. Chi, "Design of Complex Event-Processing IDS in Internet of Things," in *2014 Sixth International Conference on Measuring Technology and Mechatronics Automation*, Jan. 2014, pp. 226–229.

[123] C. Cervantes, D. Poplade, M. Nogueira, and A. Santos, "Detection of sinkhole attacks for supporting secure routing on 6lowpan for Internet of Things," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, May 2015, pp. 606–611.

[124] M. Surendar and A. Umamakeswari, "InDReS: An Intrusion Detection and response system for Internet of Things with 6lowpan," in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Mar. 2016, pp. 1903–1908.

[125] Y. Fu, Z. Yan, J. Cao, O. Kon, and X. Cao, "An Automata Based Intrusion Detection Method for Internet of Things," May 2017. [Online]. Available: https://www.hindawi.com/journals/misy/2017/1750637/abs/

[126] Y. Fu and O. Kon, "Security and Robustness by Protocol Testing," *IEEE Systems Journal*, vol. 8, no. 3, pp. 699–707, Sep. 2014.

[127] P. Tsankov, M. T. Dashti, and D. Basin, "SecFuzz: Fuzz-testing Security Protocols," in *Proceedings of the 7th International Workshop on Automation of Software Test*, ser. AST '12. Piscataway, NJ, USA: IEEE Press, Jun. 2012, pp. 1–7. [Online]. Available: http://dl.acm.org/citation.cfm?id=2663608.2663610

[128] B. Lei, X. Li, Z. Liu, C. Morisset, and V. Stolz, "Robustness testing for software components," *Science of Computer Programming*, vol. 75, no. 10, pp. 879–897, Oct. 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167642310000328

[129] D. Midi, A. Rullo, A. Mudgerikar, and E. Bertino, "Kalis A System for Knowledge Driven Adaptable Intrusion Detection for the Internet of Things," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, Jun. 2017, pp. 656–666.

[130] G. Kumar, "Evaluation Metrics for Intrusion Detection Systems - A Study," no. 11, p. 7, Nov. 2014.

[131] G. Francia, L. Ertaul, L. H. Encinas, E. El-Sheikh, and K. Daimi, *Computer and Network Security Essentials*. Springer Publishing Company, Incorporated, 2018.

[132] VCloudNews, "Every Day Big Data Statistics 2.5 Quintillion Bytes of Data Created Daily (http://www.vcloudnews.com/every-day-big-data-statistics-2-5-quintillion-bytes-of-data-created-daily/)," Apr. 2015. [Online]. Available: http://www.vcloudnews.com/every-day-big-data-statistics-2-5-quintillion-bytes-of-data-created-daily/

[133] E. Ahmed, I. Yaqoob, I. A. T. Hashem, I. Khan, A. I. A. Ahmed, M. Imran, and A. V. Vasilakos, "The role of big data analytics in Internet of Things," *Computer Networks*, vol. 129, pp. 459–471, Dec. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128617302591

[134] J. F. Puget, "What Is Machine Learning? (IT Best Kept Secret Is Optimization)," May 2016. [Online]. Available: https://www.ibm.com/developerworks/community/blogs/jfp/entry/What_Is_Machine_Learning

[135] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, "Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm," *IEEE Transactions on Computers*, vol. 65, no. 10, pp. 2986–2998, Oct. 2016.

[136] K. K. Gupta, B. Nath, and R. Kotagiri, "Layered Approach Using Conditional Random Fields for Intrusion Detection," *IEEE Transactions on Dependable and Secure Computing*, vol. 7, no. 1, pp. 35–49, Jan. 2010.

[137] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016. [Online]. Available: https://www.nature.com/articles/nature16961

[138] L. Deng, D. Li, X. Yao, D. Cox, and H. Wang, "Mobile network intrusion detection for IoT system based on transfer learning algorithm," *Cluster Computing*, Jan. 2018. [Online]. Available: https://doi.org/10.1007/s10586-018-1847-2

[139] W. Feng, Q. Zhang, G. Hu, and J. X. Huang, "Mining network data for intrusion detection through combining SVMs with ant colony networks," *Future Generation Computer Systems*, vol. 37, pp. 127–140, Jul. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X13001416

[140] W. C. Lin, S. W. Ke, and C. F. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-Based Systems*, vol. 78, pp. 13–21, Apr. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705115000167

[141] M. Nobakht, V. Sivaraman, and R. Boreli, "A Host-Based Intrusion Detection and Mitigation Framework for Smart Home IoT Using OpenFlow," in *2016 11th International Conference on Availability, Reliability and Security (ARES)*, Aug. 2016, pp. 147–156.

[142] F. Hosseinpour, A. Meulenberg, S. Ramadass, P. V. Vahdani Amoli, and Z. Moghaddasi, "Distributed Agent Based Model for Intrusion Detection System Based on Artificial Immune System," *JDCTA Int. J. Digit. Content Technol. its Appl*, vol. 7, pp. 206–214, May 2013.

[143] F. Hosseinpour, P. V. Amoli, F. Farahnakian, and J. Plosila, "Artificial Immune System Based Intrusion Detection : Innate Immunity using an Unsupervised Learning Approach," *JDCTA Int. J. Digit. Content Technol. its Appl.*, vol. 8, no. 5, pp. 1–12, Oct. 2014.

[144] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers & Security*, vol. 31, no. 3, pp. 357–374, May 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404811001672

[145] M. Sheikhan and H. Bostani, "A hybrid intrusion detection architecture for Internet of things," in *2016 8th International Symposium on Telecommunications (IST)*, Sep. 2016, pp. 601–606.

[146] ——, "A Security Mechanism for Detecting Intrusions in Internet of Things Using Selected Features Based on MI-BGSA," *International Journal of Information & Communication Technology Research*, vol. 9, no. 2, pp. 53–62, Oct. 2017. [Online]. Available: http://journal.itrc.ac.ir/index.php/ijictr/article/view/261

[147] H. H. Pajouh, R. Javidan, R. Khayami, D. Ali, 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 Transactions on Emerging Topics in Computing*, vol. PP, no. 99, pp. 1–1, Nov. 2016.

[148] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Conditional Variational Autoencoder for Prediction and Feature Recovery Applied to Intrusion Detection in IoT," *Sensors*, vol. 17, no. 9, p. 1967, Aug. 2017. [Online]. Available: http://www.mdpi.com/1424-8220/17/9/1967

[149] V. L. L. Thing, "IEEE 802.11 Network Anomaly Detection and Attack Classification: A Deep Learning Approach," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, Mar. 2017, pp. 1–6.

[150] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," *Future Generation Computer Systems*, Sep. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17308488

[151] S. Prabavathy, K. Sundarakantham, and S. M. Shalinie, "Design of cognitive fog computing for intrusion detection in Internet of Things," *Journal of Communications and Networks*, vol. 20, no. 3, pp. 291–298, Jun. 2018.

[152] M. E. Aminantoa and K. Kimb, "Deep Learning in Intrusion Detection System : An Overview," in *International Research Conference on Engineering and Technology (2016 IRCET). Higher Education Forum, 2016.*, 2016. [Online]. Available: /paper/Deep-Learning-in-Intrusion-Detection-System-%3A-An-Aminantoa-Kimb/c0fa578c1fae002e02834806a576d811002cb4a4

[153] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An Algorithm for Finding Best Matches in Logarithmic Expected Time," *ACM Trans. Math. Softw.*, vol. 3, no. 3, pp. 209–226, Sep. 1977. [Online]. Available: http://doi.acm.org/10.1145/355744.355745

[154] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, Dec. 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231206000385

[155] J. P. Papa and A. X. Falco, "A Learning Algorithm for the Optimum-Path Forest Classifier," in *Graph-Based Representations in Pattern Recognition*, ser. Lecture Notes in Computer Science, A. Torsello, F. Escolano, and L. Brun, Eds. Springer Berlin Heidelberg, May 2009, pp. 195–204.

[156] F. Al-Turjman, "Fog-based caching in software-defined information-centric networks," *Computers & Electrical Engineering*, vol. 69, pp.

54–67, Jul. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045790618311856

[157] 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 *Proceedings of the 15th International Conference on Information Processing in Sensor Networks*, ser. IPSN '16. Piscataway, NJ, USA: IEEE Press, Apr. 2016, pp. 23:1–23:12. [Online]. Available: http://dl.acm.org/citation.cfm?id=2959355.2959378

[158] D. Ravi, C. Wong, B. Lo, and G. Yang, "Deep learning for human activity recognition: A resource efficient implementation on low-power devices," in *2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, Jun. 2016, pp. 71–76.

[159] 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 Journal of Biomedical and Health Informatics*, vol. 21, pp. 56–64, Jan. 2017. [Online]. Available: http://doi.org/10.1109/JBHI.2016.2633287

[160] H. Hromic, D. L. Phuoc, M. Serrano, A. Antoni, I. P. arko, C. Hayes, and S. Decker, "Real time analysis of sensor data for the Internet of Things by means of clustering and event processing," in *2015 IEEE International Conference on Communications (ICC)*, Jun. 2015, pp. 685–691.

Nadia Chaabouni received the Master's degree in computer science from the University of Bordeaux, in 2016. She is currently pursuing Ph.D. in the University of Bordeaux and in AtoS Innovation Aquitaine Lab under the supervision of Mohamed Mosbah and Akka Zemmari from the University of Bordeaux, France, and Cyrille Sauvignac from Atos innovation Aquitaine Lab, France. Her research interests include the area of Internet of Things Systems Security and machine learning techniques.

Mohamed Mosbah is a full Professor in computer science at the Polytechnic Institute of Bordeaux, France. He obtained his Ph.D. from the University of Bordeaux, in 1993. He carries his research in LaBRI, a research Lab in computer science common with the University of Bordeaux and CNRS, where he is currently the Deputy Director. His research interests include distributed algorithms and systems, formal models, security, and ad hoc and sensor networks. He participated to several national and European research projects, including collaborations with industry. He wrote more than 60 research papers published in international journals and conference proceedings and he is involved in various technical program committees and organizations of many international conferences.

Akka Zemmari has received his Ph.D. degree from the University of Bordeaux, France, in 2000. He is an associate professor in computer science since 2001 at University of Bordeaux, France. His research interests include distributed algorithms and systems, graphs, randomized algorithms, machine learning and security.

Cyrille Sauvignac is a Project Manager Professional Certified and an Innovation manager (ITEA2 Usenet, ITEA2 A2Nets, Smart Services for connected vehiCles ITS project), with strong background in real time and avionics systems, enterprise portals, M2M interoperability standardization and embedded devices. Now, Cyrille is in charge of Atos innovation Aquitaine Lab to develop ITS & IoT complex systems.

Parvez Faruki received M.Tech and PhD in Computer Science and Engineering from Malaviya National Institute of Technology Jaipur India in July 2012 and March 2016, respectively. In 2012, he was awarded CFAIT Common wealth fellowship for further research. He was a part of research team on behalf of NIT Jaipur for a joint DST-RFBR project with South-Russian Educational and Research Center for IT-Security at SFU Russia. He visited LaBRI-Laboratoire Bordelais de Recherche en Informatique Bordeaux, France to pursue research in 2015. He published 19+ papers, few among the topmost international peer-reviewed journals and good security conferences including IEEE Communication Surveys & Tutorials and ACM Computing Surveys.