Host-based IDS: a review and open issues of an anomaly detection system in IoT

Inês Martins[a], João S. Resende[a], Patrícia R. Sousa[a], Simão Silva[a], Luís Antunes[a], João Gama[a]

[a]*Universidade do Porto*

Abstract

The Internet of Things (IoT) envisions a smart environment powered by connectivity and heterogeneity where ensuring reliable services and communications across multiple industries, from financial fields to healthcare and fault detection systems, is a top priority. In such fields, data is being collected and broadcast at high speed on a continuous and real-time scale, including IoT in the streaming processing paradigm. Intrusion Detection Systems (IDS) rely on manually defined security policies and signatures that fail to design a real-time solution or prevent zero-day attacks. Therefore, anomaly detection appears as a prominent solution capable of recognizing patterns, learning from experience, and detecting abnormal behavior. However, most approaches do not fit the urged requirements, often evaluated on deprecated datasets not representative of the working environment. As a result, our contributions address an overview of cybersecurity threats in IoT, important recommendations for a real-time IDS, and a real-time dataset setting to evaluate a security system covering multiple cyber threats. The dataset used to evaluate current host-based IDS approaches is publicly available and can be used as a benchmark by the community.

Keywords: Internet of Things, Security, Intrusion Detection Systems, Real-time Streaming Processing, Anomaly Detection, Host-based Intrusion Detection System

1. Introduction

In a digital world growing at an incredible rate, the IoT plays a prominent role in our everyday lives by empowering interconnection and integration in physical and cyberspaces. Nowadays, the IoT paradigm embodies a dynamic global network infrastructure, offering high levels of accessibility, integrity, availability, and interoperability among heterogeneous smart devices [1].

Although the signs of progression in IoT are undeniable, they also represent a major challenge in privacy and security as the number of smart devices and dependencies increase. The pervasive connectivity to the internet poses numerous hidden security risks, such as eavesdropping on the wireless communication channel, unauthorized access to devices, or tampering with devices [2]. According to the statistics of AV-TEST[1] institute in Germany, there are more than a billion malicious executable scripts known to the security community. In fact, the digital transformation also meant the increase of cybercrime, often associated with significant financial losses for both individuals and organizations.

In particular, cyber threats continue to evolve and target IoT devices and communications that have been enabled by a weak network security posture and obsolete devices. As an example, it has been reported that 72% of healthcare systems mix IoT and IT assets, allowing malware to spread from users' computers to vulnerable IoT devices on the same network, and 41% of attacks scan through network-connected devices in an attempt to exploit known vulnerabilities[2]. Accordingly, to ensure reliable services, the network should not allow unauthorized access by verifying reliable communication methods to send and receive authentic information and perform sensor operations, transmissions, and treatments safely in real-time [3].

Cybersecurity analysis relies on vast data to predict, identify, characterize, and deal with threats. As the volume of data and complexity increases, all human efforts are not enough to deal with the cyber defense urgency. Recent developments in computer data acquisition, storage, and processing fueled the application of Machine Learning (ML) to step in and help to detect complex patterns and trends more efficiently and faster than humans [4]. Although smart devices can provide intelligent assistance and reduce manual work, most ML solutions count numerous incompatibilities with the IoT paradigm. From the prerequisite of training with huge amounts of data to the security risks involved with sharing raw data in complex and computational expensive frameworks, current solutions prove to be unsuitable for a continuous and real-time infrastructure [5].

The intertwined topics demand online security solutions and applications to build robust tools to defend systems against security threats. One of the most prominent methods - IDS - is built to monitor systems and detect anomalies or privacy violations [6]. These systems, devices, or software are responsible for preventing breaches of security incidents, monitoring, and reacting to any unauthorized access that causes damage to the

⋆Fully documented templates are available in the elsarticle package on CTAN.

Email address: inesmartins@fc.up.pt (Inês Martins)

[1]https://www.av-test.org/en/statistics/malware/(January 2021)

[2]https://iotbusinessnews.com/download/white-papers/UNIT42-IoT-Threat-Report.pdf)

stored information, the information system, or the network [7].

Although the necessity to build robust tools to defend networks and systems has been highly documented, current ongoing solutions prove ineffective against new threats and zero-days. The most common detection technique consists of a prebuilt intrusion pattern database to classify events. Despite obtaining low false alarm rates for well-known attacks, these systems cannot detect unknown threats as their signatures have not yet been created. On the other hand, in anomaly detection, decisions are made based on normal behavior or features. In this sense, a stream that significantly deviates from the expected pattern will be considered an intrusion. Although this solution offers the possibility to identify zero-day attacks, it still obtains high false alarm rates, rendering the intrusion system ineffective [8].

Current intrusion detection solutions that have been presented fail to design a system that matches all the requirements inherent in a real-world environment. Particularly, anomaly detection in cybersecurity is associated with an imbalanced and dynamic domain where the least-expected outcome is highly relevant, potentially undermining its performance and compromising the security system [9]. Additionally, real-time constraints of security IoT environments require efficient computational approaches to enable efficient solutions. Current solutions often focus on accuracy without considering the time delay between training, prediction, and the active response [10].

As the amount of data generated in real-world applications increases, deep human expertise is required to validate the entries and ensure reliable services [11]. Furthermore, current tools are often based on outdated and static datasets that cannot reflect the reality of modern threats or the ability to understand the evolution of concepts [6]. These shortcomings lead to a preference for signature-based systems over anomaly detection and, consequently, the IDS's vulnerability to new and adversarial attacks.

Accordingly, motivated by the incompleteness of currently available solutions in one of the most critical topics in the research field, the urgency to build an anomaly detection system that offers a real-time effective and efficient security system in IoT is the core issue of this work. Furthermore, with the increasing number of IoT applications, this work offers an overview of the most relevant concepts of an IDS, from the background concepts to a representative real-time dataset. Therefore, this paper focuses mainly on three research questions outlining our contributions:

RQ1: What are the main requirements of intrusion detection in IoT, from its paradigm to security challenges?

RQ2: How to design a complete intrusion solution suitable for an IoT cyber domain?

RQ3: How can a HIDS dataset represent a continuously evolving IoT ecosystem?

In summary, the main contributions offered by this work can be stated as:

- An overview of the most essential concepts to help identify points of failure and facilitate a risk assessment for a particular infrastructure. This contribution shapes the ecosystem and formulates RQ1 by defining the environment, from its challenges, main requirements, and data processing paradigm to the IDS taxonomy and classification of prominent cyber threats, according to the IoT layer, class vector, or compromised security principle.
- A set of guidelines and considerations dictated by the environment and fundamental requirements to design a real-time IDS in IoT, followed by a review of recent IDS proposals, delineates RQ2 and attests that there is not yet an efficient solution that meets all the directives.
- An evaluation setup configuration designed to test real-time and streaming systems using reported vulnerabilities as anomalous traffic. RQ3 is fulfilled, and a publicly available dataset containing cyber IoT attacks, ready to be exploited on multiple platforms, is proposed.
- Additionally, an open-source HIDS evaluation, using our proposed setting and attesting performance and detection methodology as a flexible and robust approach to different threat vectors and zero-day.

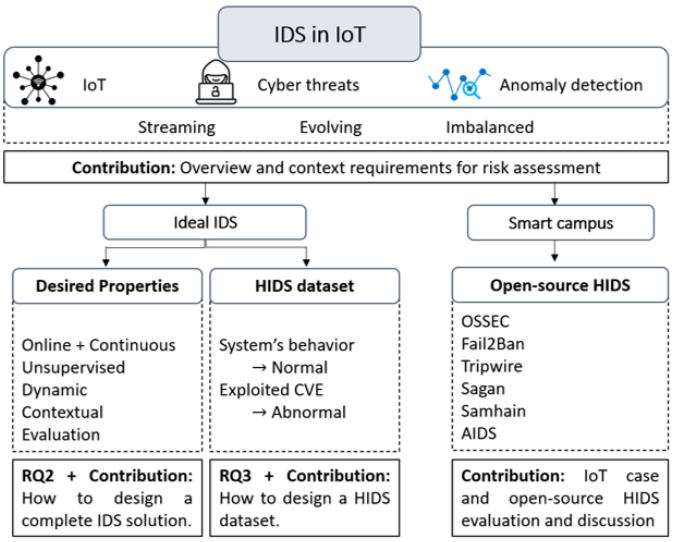

Figure 1: Scope and organization of this work.

Following the stated motivation and contributions, Figure 1 illustrates the organization, scope, and interactions of this work, starting by pointing out the inherent characteristics imposed by an intrusion system in IoT. As the problem dimensions and the operating conditions define the working environment, this first topic formulates RQ1 and our first contribution. After analyzing identical reviews on IDS systems in IoT, this review defines an ideal IDS in IoT by delineating desired properties to implement a practical solution and to build a HIDS dataset honoring real applications processes and meeting RQ2 and second and third contributions. Finally, using an IoT use case to continue reviewing open-source HIDS, RQ2 and RQ3 are completed by testing and discussing available intrusion systems. Therefore, this work distinguishes from similar reviews that neglect streaming characteristics or cyber threats and adversarial attacks by stating and describing the problem dimension considering both anomaly detection and cybersecurity fields, de-

tailing an ideal IDS, and suggesting a HIDS dataset using IoT vulnerabilities for a reliable evaluation scheme.

Concisely, this paper is structured as follows: Section 2 contains the main requirements, concepts, IDS taxonomy, and IoT threats taxonomy according to their procedures and compromised security principles leading to our first research question; Section 3 reviews similar studies comparing them to our work; Section 4 follows some crucial directions to build a reliable solution and discusses current intrusion systems that have been proposed; Section 5 details an open-source and real-time dataset and evaluation scheme for an IoT use case scenario, reporting the architecture and management services to attest our convictions; Section 6 and Section 7 close this paper by advancing future research challenges, significant discussions, and important topics to consider for a future research direction towards a real-time and robust security system.

2. Contextual Information

Discussing the background by describing the main concepts and requirements of an intrusion detection task in IoT, considering its characteristics and goals, is crucial to building a realistic solution that fits the environment's specifications and outlines the specifications to achieve. Figure 2 depicts the buzzwords for each concept as IoT domain, streaming processing, and intrusion detection task as vital subjects to consider for a realistic and practical cybersecurity system.

The first security concept specifies the IoT environment, from industry, finance, manufacturing, healthcare, our cities, and homes, enabling automation, cutting waste, and improving monitoring abilities. This environment incorporates heterogeneous devices, producing highly imbalanced and imprecise vast amounts of data portrayed as ambiguous and imperfect [12]. Due to the IoT characteristics, the data collected have a distributed and real-time nature, high volume, fast velocity, and variety. It demands cost-effective processing in order to be able to deliver enhanced insights and decision-making. Its high speed and possible low quality and interpretation have been proving to be the challenges posed by IoT since they compromise the models' consistency and reliability [13].

The second topic to address is the task of monitoring systems, where the data flow can be seen as a continuous stream of inputs, denoting data flowing in and out continuously. The characteristics of a real-time and continuously evolving paradigm introduce the second security concept presented in Figure 2. Streaming data processing is beneficial in most everyday scenarios where new and dynamic data is continually generated, allowing continuous monitoring of real-time data. In this sense, the streaming resource limitations meet the IoT directives [14]. Moreover, a streaming environment like the one found in IoT applications must consider the possibility of concept drift where data characteristics change over time. In this sense, security solutions need to be updated to detect dynamic changes in a fast and accurate way as the underlying assumptions used to validate the system can be affected, reducing its relevance with time [15].

Finally, an intrusion system depicted as a detection problem belongs to a dynamic and real-time anomaly detection scenario designed to identify possible attacks and proper responses [16], the number of threats can be considerably lower than the ordinary events, resulting in an imbalanced setting. Additionally, when dealing with cyber threats, an attacker could target the intrusion system, manipulating its predictions and responses, which adds the adversarial prospect to the security contextualization presented in Figure 2.

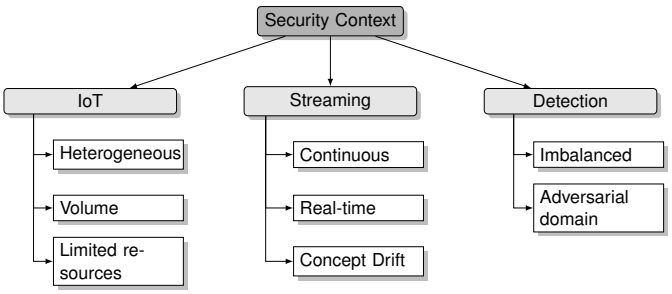

Figure 2: Security domain contextualization

2.1. Intrusion Detection Systems

Intrusion detection systems are common cybersecurity mechanisms designed to gather, process, and analyze the information derived from computer hosts or networks to identify malicious activities such as security breaches, including attacks arising inside or outside the infrastructure [17].

According to the information source and examined activity, the most common solutions are host-based intrusion detection systems (HIDS) and network-based intrusion detection systems (NIDS). Based on the types of analyzed data, another classification appears by employing multiple specialized detectors at different layers (network, kernel, and application) for a hybrid-based intrusion detection system that combines previous methods and other security mechanisms for a more practical detection of cyber attacks [18].

The host-based systems monitor system activity, such as files modifications or memory usage. Internal monitoring, which relies heavily on audit trails and system logs, determines if a system has been compromised. This approach runs on individual hosts monitoring the device and detecting improper use of the available resources. Popular examples of HIDS are *OSSEC* [19] or *Tripwire* [20] responsible for log analysis and file checking, respectively. On the other hand, network-based systems focus on monitoring network activity, communications, and auditing packet information to protect a system from network-based threats by searching inbound packets for suspicious behavior. This approach often operates under promiscuous mode by intersecting and reading packets without exposing them to potential threats. Two of the most popular NIDS applied in cybersecurity are *Snort* [21] and *Suricata* [22]. The first option provides real-time intrusion detection and prevention, as well as monitoring network security. *Suricata* is a modern alternative to *Snort*

with multi-threading capabilities and multiple model statistical anomaly detection [23].

On a comparative view, whereas a NIDS is mostly centrally managed in infrastructures with different devices, HIDS is a machine-oriented and distributed protection system. Although HIDS can exhaust many resources, increase time and space complexities, and have a hard time spreading cross-platform interoperability, this approach can still be practical if the host is connected to different networks. Despite NIDS being operating-system independent, this solution fails to handle high-speed networks efficiently, and it cannot scan the content of the network traffic if it is encrypted [24]. Furthermore, the NIDS solution is specific to network threats based on packet information, neglecting the interactions with the rest of the network, making it vulnerable to adversarial threats.

Although hybrid-based solutions can add the advantages of both methods for a more effective solution, their efficiency in handling the summarization of real-time data streams is compromised as the infrastructure evolves complex.

As these security mechanisms aim to spot cyber threats, the attack detection method adopted by an IDS is a widely covered topic in the literature. Depending on their approach, signature, anomaly, or hybrid detection methods are the most common techniques.

In signature-based detection, network traffic or system-level actions are compared against a well-known collection of previous attack signatures such as bit patterns, keywords, known malicious instruction sequences, and system vulnerabilities [25]. This approach efficiently identifies old and analyzed attacks but fails to detect zero-day attacks. For example, if the attack goal is to exploit a particular buffer overflow, the IDS can use pattern matching to look for particular strings. Due to its architecture, this method implies collecting and maintaining rules on a database. As a result, these methods only score low false alarm rates if the signature for a particular attack is already available. Some researchers classify this methodology as misuse-based and, depending on if the rules reflect the abnormal or expected behavior, separate into two more concepts - signature or specification - respectively [26]. Specification-based appeared as a solution to detect zero-day while keeping a low false alarm rate [27]. However, this alternative shares the same disadvantages of maintaining the updated standard pattern and being vulnerable to adversarial attacks that impersonate the expected behavior.

In anomaly-based, a model is built from a previous network or system activity data based on the assumption that any attack, interpreted as an anomaly, will have a different dynamic, flagging any discrepancies as suspicious. Although this technique excels at detecting new attacks, it requires periodic updates as the ongoing distribution is not static. Although anomaly-based detection, augmented with data mining techniques, can help to identify new exploits with powerful insights, such methods and their training data are vulnerable to a variety of security threats [28]. Although the accuracy of these methods against zero-days is better when compared to signature-based, the false alarm rate is often higher as the boundary between the expected and anomaly behavior can be difficult to define [6, 8, 29].

As a result, a new approach has been documented as an alternative to improve misuse-based by combining the attributes of both strategies and accumulating knowledge about specific attacks or system vulnerabilities. In hybrid-based methods, often reinforced with domain experts, the disadvantages of one method are mitigated by the strengths, increasing performance and facilitating preventive or corrective actions while also introducing a delay in the validation. These topics may compromise the real-time response required by an effective and efficient IDS in IoT [26].

According to the designed IDS architecture deployment structure, standalone and collaborative schemes are commonly discussed topics where the former depends on traffic patterns and enables more continuous and straightforward tracking of data within the infrastructure, without depending on additional domain or user information [30]. On the other hand, the latter enables an IDS node to exchange required information with other IDS nodes providing context and keeping a broader look at the network. While collaborative approaches can reduce the unnecessary network usage of equally assigned tasks, these solutions assume that all peers are reliable. Therefore, this alternative is vulnerable to insider attacks when a node in the network is corrupted [31] or to privacy policy violations when the user data is shared [5]. In this sense, it is often seen collaborative options adopting trust mechanisms to help mitigate insider attacks by assigning node reputations and robustness to the security system [32]. According to its communication architecture, collaborative IDS can be categorized into centralized, decentralized, and distributed. In the first setting, there is a central analysis unit applying alert correlation algorithms on received alerts or detection algorithms on traffic data. On a decentralized approach, the preprocessing and correlation of the monitored data is employed through the hierarchy until it converges to a central unit, reducing unnecessary resource usage. Finally, in distributed solutions, all monitors are analysis units where all tasks are equally assigned [33].

Table 1 summarizes the taxonomy of an IDS based on the information source, detection method, and primary architecture schemes, as well as the main advantages and disadvantages discussed early on.

2.2. Anomaly Detection

In 1980, *Hawkins* [34] described an anomaly as an observation which "deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism." The importance of anomaly detection arises as many anomalies represent essential, prominent, and often critical information in a wide variety of applications. As computer systems and networks become more complex and exposed to vulnerabilities and sophisticated attacks, anomaly detection emerges as a fundamental measure of security [35].

Thereby, intrusion detection can be considered a subproblem of anomaly detection, which endeavors to determine and control abnormal incoming events [36]. This domain can provide meaningful solutions by tracking relevant hidden patterns imperative in identifying intrusions. Given the ultimate goal of detection process automation, the main objectives of monitoring

Table 1: IDS taxonomy and main advantages and disadvantages

		Advantages	Disadvantages
Information source	Network	Packet rejection Independent environment	Protection off LAN Limited visibility inside the system
	Host	Detect insider threats System monitorization	Host resource requirements Compromised with the host
	Hybrid	Interoperability	Complexity
Detection method	Signature	Low false alarms Simple design	Frequent updates Unable to detect zero-days
	Anomaly	Detects zero-days Creates new signatures	High false alarms Hard to define normal behavior
	Hybrid	Detects zero-days	Efficiency
Architecture	Standalone	Simplicity Continuous flow	Lack of context
	Collaborative	Sharing data Robustness	Complexity Insider threats

events are to ensure data protection and implement lightweight, deployable, scalable, and resilient systems to unknown threats without requiring hardware updates or prior knowledge of the anomaly detection method [37].

Anomaly approaches rely on behavior analysis by observing a sequence of events to define standard patterns. However, it is a misconception to consider anomaly detection as behavior analysis since signature-based methods also apply behavior evaluation [38]. Due to its ability to recognize any deviation from the usual activities, this approach can detect novel attacks and provide a customized model for the typical operations, lowering the possibility of an attacker disguising its movements.

Given the requirements imposed by a security system applied in IoT, the data collected, from network flow data and sensors data to host performance metrics or system logs, have a distributed and real-time nature. This paradigm, which portrays real-world data as a continuous stream, drives current approaches to be dynamic and to handle massive data analysis without consuming substantial resources of computational power and memory [39].

In real applications, the underlying dynamic is never static. Our opinions and reactions evolve, introducing concept drift as new knowledge of real-time applications. Hence, the security system needs to be updated in order to detect dynamic changes in a fast and accurate way [40]. Concept drift can affect the decision boundaries decreasing performance with time. Therefore, concept drift methods should be integrated into every anomaly detection solution to cope with possible changes that can incorrectly flag anomalies and increase the false alarm rate. The first challenge of drift detection is its multiple types, including gradual, sudden, and recurring drifts. The second challenge includes IoT systems factors such as system updates, IoT device replacement, and abnormal network events [41]. Learning-based methods work under a closed and static assumption, failing to include organic behavior and malicious mutations introduced by attackers [42]. Although many research works primarily focus on static models that require periodic training based on its performance, some solutions propose time window-based methods to monitor statistical differences between windows [43, 44], implement forgetting factors to fade old samples' importance [45], and build online streaming models to readjust as a new instance appears [46].

Given the distribution of stream events in security applications, anomaly detection adds the imbalanced data problem as a core concept. The class-imbalance problem arises when the recurrence of an event is much less frequent than often occurring events. Rare events are difficult to detect because of their infrequency and casualness. However, misclassifying rare events can result in high costs. In cybersecurity, an anomalous entry can be infrequent compared to the thousands of network traffic events a system is processing. However, failing to identify a cyber attack could compromise the infrastructure and resources, resulting in incalculable damages [47]. Recent research works focus on evaluating their proposed methods on datasets that offer some attack coverage and an attack ratio on the order of dozens to tackle the imbalanced proportion [48, 49, 50]. Nevertheless, especially in a real-world IoT environment, where the percentage of anomalous events can be even lower, such approaches are not realistic enough.

ML has been a prevailing research field in various intrusion or fraud detection applications. In an imbalanced scenario, the use of any potential biased measures to average the system's performance is discouraged. The conventional way of maximizing overall performance will often fail to learn useful insights from rare events due to the dominating effect of the majority class. A solution to balance the number of samples is to remove some of the majority class or add anomaly samples by replicating or artificially generating more cyber attacks. In fact, network traffic targeting computational resources, known as honeypots, designed to lure hackers, can contribute with anomalous data similar to the one that is being monitored [51]. As the dynamic generated in honeypots can be different from an IoT environment-designed device, a more realistic solution would be to self-inflict cyber threats to generate more samples in a controlled environment. In IoT, such procedures might lead to unrepresentative and unrealistic models [52].

ML has been formulated as an effective solution to many IDS problems by extracting useful patterns and improving performance. However, most of the proposed ML methods discussed in the literature and throughout this work do not implement computationally light approaches to enable fast prediction. Despite acknowledging velocity and variety, most solutions still focus on developing parallel implementations [53], big data processing paradigms [54], and interpolate between nodes and cloud servers updated regularly [44]. Furthermore, when gathering data from sensors with different sample frequencies and spatial and temporal contexts, IoT data presents high dimensionality, increasing data complexity and compromising processing times and performance [55, 56]. In this sense, relevant

topics must be considered when designing an anomaly-based model. From concept drift and imbalanced settings to real-time, statistical summaries, and online solutions, the computational load and impact on the feasibility in real-world systems should be a top priority [10].

2.3. Cybersecurity threats in IoT

The IoT ecosystem is considered an attractive target due to its interdependence and interactions between devices, making it possible to attack surrounding components to access the target. The diversity and constrained characteristics, accommodating different applications and scenarios, almost unique for each task, also affect the system's security. These factors require very specific security designs, which become less stable as the devices become more lightweight and small [57]. In summary, IoT systems function in more dangerous and heterogeneous environments with limited resources, fewer security guards and a large attack surface.

Network and systems' protection requires understanding the common causes, procedures, and protection methods of a data breach where the interaction with domain experts can indicate and interpret potential points of failure and hidden relations between components induced by cyber threats [58]. Therefore, with a thorough study on the cybersecurity field and risk assessment focusing on information assets, security solutions become more proactive and robust against subsequent attacks [59].

Accordingly, security architecture must honor principles such as confidentiality, integrity, and availability to ensure reliable services. These fundamental measures protect data from being exposed from unauthorized access, protect information's accuracy and completeness from unauthorized alteration, and ensure that the system is available to all users, respectively [60]. In this sense, the network should ensure reliable communication to send and receive authentic information and perform sensor operations, transmissions, and treatments safely in real-time [61].

Cyber attacks correspond to a type of offensive action that intends to intercept, steal, alter, or destroy data or information systems, compromising computer information systems, infrastructures, computer networks, or devices. According to the attacker's intentions, it is possible to distinguish two types of attacks - active and passive. The first type represents the attacks that intentionally disrupt the system, alter system resources, or affect its operations. It involves masquerading when an entity pretends to be another, modifying messages, or overloading the system. Passive attacks attempt to learn or use information from the system but do not affect system resources. These attacks appear like eavesdropping on or monitoring of transmission aiming to obtain information transmitted in the network [62]. In this sense, active attacks compromise integrity and availability, while passive attacks endanger confidentiality.

Passive attacks like eavesdropping or monitoring, which do not have a break or evade feature, will not have a particular impact on the system's standard expression, suggesting a difficulty to host-based systems. However, some attacks, like SQL-injection, where its ultimate goal is to expose or alter information in a database, cannot be considered active or passive until their true purpose is exposed [63]. Under these circumstances, it is expected that the basic requirements, such as secure communications, are guaranteed to diminish the efficiency of passive attacks.

Table 2 classifies the IoT attacks based on the class vector, compromised security principles, and hardware vs. software procedures that describe the scope of the most popular threats. The IoT architecture is organized based on the scoped abstraction level and the performed duties [64]. Each row reporting common IoT cyber threats in smart applications is described according to the designed target layer procedure, the methodology attack vector, and the compromised principle of common practices.

Table 2: IoT threats taxonomy

	Procedure		**Vector**					**Principles**		
	Hardware	Software	DoS	Injection	Spoofing	Eavesdropping	Engineering	Confidentiality	Availability	Integrity
Fault Injection [65]	✓			✓				✓		
Sinkhole [66]	✓				✓					✓
Jamming [67]	✓		✓						✓	
Sleep deprivation [68]	✓		✓						✓	
MitM [69]	✓					✓		✓		✓
Phising [70]		✓					✓		✓	
Sybil [71]		✓			✓				✓	
XSS [72]		✓		✓				✓	✓	
XXE [72]		✓		✓				✓	✓	
RCE [73]		✓		✓				✓	✓	
SQL-injection [63]		✓		✓				✓		
PE [73]		✓						✓		
Crypanalysis [74]		✓					✓			✓

Initially, based on the chosen procedure to attack the IoT architecture, cyber threats can be categorized into physical or software attacks depending on if the target the hardware structure or the preferred strategy is to perform an attack by using a virus, worms, spyware, or adware [75]. Therefore, physical attacks target the perception layer responsible for collecting information and performing different measurements such as temperature or humidity through sensors and actuators. This layer is particularly affected due to its physical device exposure, resource-constrained devices, technological heterogeneity, and distributed architecture that hinders the authentication process [76]. Among its threats, the common approaches can be distinguished as injection and spoofing attacks where vulnerabilities are exploited by injecting input into the infrastructure, such as fault injection, or by impersonating and falsifying data and nodes, such as sinkhole attacks [77]. These alternatives are a physical attack on the data and behavior of the circuit, stealing

private data and compromising the infrastructure [65] or guiding the network traffic towards malicious nodes, compromising the integrity principle [66], respectively. In this category, depending on if the availability of the system is compromised by making it inaccessible to others, the attack vector is classified as a jamming attack, which tries to disrupt the communication by decreasing performance, ending up in sleep deprivation or Denial of Service (DoS), jeopardizing availability in the infrastructure [67]. DoS is a very common vector in network attacks in which the perpetrator tries to make the resources unavailable, typically by flooding and overloading the system [68]. On the other hand, another strategy is to intercept information by eavesdropping, such as Man in the Middle (MitM), where the attacker secretly relays and possibly alters the communications between two parties. In this sense, this attack compromises both confidentiality and integrity principles [69].

The network and application layers are responsible for the communication between devices and processing information and different tasks, respectively. Given the heterogeneity of the network and large users accessibility and potential critical applications it has access to, these layers face routing, transit, and data leakage attacks and vulnerabilities [76]. In these layers, which belong to a more software vector, the information or the communications between devices are the most valuable asset. Therefore, the common attack vector seeks to obtain access to a system, analyze side-channel interactions using a passive strategy, or disrupt the system [75]. In this category, it is possible to encounter spoofing attack impersonating reliable sources or social engineering attacks that aims to steal confidential data through human interactions, compromising confidentiality fundamentals. In this sense, Phishing and Sybil attacks [70, 71] are common threats that impersonate a reliable source or node to obtain information from a target, decode, and manipulate unencrypted packets, targeting trust and secure connections as described in Bluetooth impersonation attacks [78]. Similarly to physical attacks, in the hardware layer, we can find injection vectors, malicious inputs executed as a program in websites or queries, such as cross-site scripting (XSS) or SQL-injection, respectively. Moreover, XML external entity injection (XXE) is a web security vulnerability that interferes with the XML data, allowing an attacker to view files on the server and connect with external systems interacting with it. This vector can lead to a confidentiality leak by accessing unauthorized files, proving that most approaches are not isolated, and their consequences can escalate and compromise different security principles [72]. Another popular method exploits vulnerabilities in systems or networks to get privileged access or bypass access control, compromising the confidentiality principle. In this category, Privileged Escalation (PE) and Remote Code Execution (RCE) are application-layer threats that aim to control the system by exploiting design flaws to access resources that should be unavailable or to execute malicious code on a remote machine and take complete control of an affected system [73]. On the other hand, in this setting, despite not compromising the availability of the infrastructure, it is still important to mention encryption attacks that aim to uncover information about the encryption technique used and private keys. As a result, attacks such as MitM and cryptanalysis are regular approaches [74].

As evidence, all threats, even with different assumptions and principles, have the same target and purpose. They either aim to control the devices or steal the information collected in the perception layer. Looking at the infrastructure, the attacks influence the system itself. Hence, regardless of the assumption or the target layer, the system is expected to reflect the imprint caused by the attack, internal or external, as a direct or indirect consequence.

In order to facilitate the communications between users and manufacturers, the report of updated vulnerabilities using public lists maintained by institutions and financed by governments can be used to disclose computer security flaws publicly. An example of these lists is the Common Vulnerabilities and Exposures (CVE), a database that provides an identification number, a description, and at least one public reference to a security breach.

3. Comparative review on Intrusion Detection Systems in IoT

As IoT embodies a dynamic global network infrastructure where the number of vulnerabilities and attack vectors grow day by day, threat detection has been a highly documented topic over the past few years as it is a shared concern among different areas [1].

From information sources, architecture debate, detection techniques, and data collection challenges, it is difficult to monitor real-time attacks as they can embody different infrastructures, locations, and tasks. Although most literature surveys define the IoT environment, security issues and attack taxonomy, when anomaly-based solutions are taken into consideration, some inbred background concepts regarding real-time nature, concept drift, and properties are still neglected, only focusing on increasing performance. Considering the background concepts, from the domain characteristics to the anomaly and cybersecurity taxonomies, as well as the validation approaches, often disregarded, Table 3 summarizes the essential topics an intrusion defense approach must cover to design an effective and practical system in real applications. As such, similar reviews were analyzed according to the streaming characteristics, anomaly detection, IoT and cybersecurity taxonomies, analyzed datasets, and validation criteria, distinguishing our work from the equivalent literature.

Sicato et al. [79] touches on relevant aspects of security issues, vulnerabilities, and attack surfaces in IoT. It provides an overview of IDS in IoT, mentioning deployment strategies, detection techniques, and data source methods. Similar to other surveys on intrusion detection in IoT, the taxonomy attacks have been discussed according to the 3-layer IoT architecture, detailing specific attacks and security principles that come with IoT issues. Furthermore, this work proposed a distributed software-based IDS, which allows a dynamic, evolving concept in the design and management of optimized network resources. Despite showing special concerns about complexity in lightweight IoT devices and listing security challenges in discovering realistic attack models, poor protection methods,

Table 3: Related work comparisons

	Streaming characteristics	**Anomaly Detection taxonomy**	**IoT and Cybersecurity taxonomy**	**Analyzed datasets**	**Validation criteria**
Sicato et al. [79]	Real-world Dynamic heterogeneity	-	IDS taxonomy (detection method, architecture, validation) IoT 3-layer taxonomy Privacy concerns Security issues Security threats	Experimented on NLS KDD dataset	Effectiveness
Aravamudhan et al. [30]	Real-time Dynamic Limited resources Heterogeneity	-	IDS taxonomy (information source, detection method, validation)	-	Effectiveness Efficiency
Liu et al. [80]	Real-time Scalable	Feature selection Rule learning Classification Clustering Hidden Markov model Neural networks	IDS taxonomy (information source, detection method), Collaborative	Overview HIDS datasets Dataset customization	Effectiveness Efficiency
Khraisat et al. [81]	Real-time Dynamic	Supervised Unsupervised Reinforcement learning Deep learning	IDS taxonomy (detection method, architecture, validation) IoT 3-layer taxonomy Privacy concerns	Data representative of the working environment	Effectiveness
Adnan et al. [10]	Real-time Dynamic Concept drift	Concept drift High dimensional aware ML Computational efficient ML	IDS taxonomy (detection method) Anomaly-based IDS (knowledge, statistical, ML)	Review well-known datasets	Effectiveness Efficiency
This work	Real-time Dynamic Concept drift Heterogeneity Continuous Imbalanced	Concept drift Computational efficient ML	IDS taxonomy (information source, detection method, architecture) IoT 3-layer security issues and threats Privacy	Customizable real-time dataset	Effectiveness Efficiency

trust challenges, privacy, and malicious adversarial threats, this work fails to mention anomaly detection methods to cope with intrusion threats, imbalanced nature, concept drift properties or validation concerns about real-time demands other than performance.

Aravamudhan et al. [30] focuses on describing IDS taxonomy characterized based on deployment architecture, information source, and detection method. It gathers various literature evidence on demand for intrusion detection systems, analyzing the drawbacks in current solutions. This work discusses the IoT ecosystem, reporting important inherent challenges, such as limited resources, multi-level attacks, difficult device protection, and heterogeneity data collection. Furthermore, this survey points out the dynamic and real-time nature, bearing highly efficient and scalability approaches with real-time, fast, and accurate responses. Despite the obvious concern regarding the requirements imposed by the IoT network, this work misses an attack taxonomy according to security challenges and principles compromised by a network exposed to different attack vectors and vulnerabilities.

Liu et al. [80] provides a thorough study, reviewing the development efforts for system-call-based HIDS and describing trends based on reduction of the false positive rate, improvement of detection efficiency, and enhancement of collaborative security. This work also offers an overview of current HIDS datasets, detection methods, and future research trends, pointing out detection efficiency, the importance of datasets representing real-world and contemporary systems, and processing techniques to handle known and unknown intrusions. Additionally, this survey supports dataset customization to represent intrusion methods, summarizing into tables for system call traces and dataset generation security tools, as well as discussing the application of system call and cloud-based HIDS on embedded systems. Although this work endorses enhancing collaborative security and constructing a real-time and scalable framework, it still fails to mention the imbalanced nature or concept drift that needs to be accounted for in a dynamic stream.

Khraisat et al. [81] discussed IoT techniques, deployment strategies, problems related to datasets, and validation of security systems, contextualized with a use case scenario in ICS (Industrial Control Systems). IDS solutions are also classified according to their placement (distributed, hybrid, or centered), detection method, and validation techniques. This work presents a list of major causes of IoT as a malware target and defines an IoT attack taxonomy based on the common 3-layer architecture. Moreover, security in IoT is a major concern where pri-

vacy, compromised communication protocols, and data collection processes representative of the working environment play an important role. However, despite providing a specific section about ML methodologies and briefly referring to IoT data as voluminous and varied, the streaming anomaly scenario, imbalanced and concept drift nature, as well as memory and time complexities were not an obvious concern.

Adnan et al. [10] contributes with a review article tackling three of the most IDS problems in a streaming IoT environment. This work emphasizes the evolving characteristics of a dynamic real-world system sustaining that an IDS must account for the concept drift nature of IoT data, high dimensionality particularly challenging in streamed data due to the inability to store or process continuous flows, and computational efficiency of ML methods interfering on the feasibility in IoT ecosystems. Moreover, this survey provides a taxonomy of IDS systems organizing the main concepts into anomaly, signature, and specification, only considering intrusion systems as computer network security monitoring and surveillance. Despite discussing important concepts such as the non-stationary nature of IoT data and briefly referring to the imbalanced distribution problem in anomaly detection, this work does not mention the outdated and unrealistic properties of well-known datasets lacking volume, variety, and veracity. In fact, this survey fails to provide a solid cyber background on IoT taxonomy and adversarial threats.

Concerning these recent surveys, our approach starts by defining the background concepts regarding anomaly detection in IoT, collecting data from different devices across a vast network, and listing IoT data in a streaming, dynamic, and real-time nature, connecting anomaly detection and cybersecurity fields. To the best of our knowledge, this is the first review that presents some guidelines to design an anomaly-based security system and a real-time dataset and validation set, which are used to evaluate familiar tools in a smart case application. As in other approaches, our work provides a security threat taxonomy based on the IoT architecture and the security principles and issues an IoT network is exposed to.

4. Building an Host-based Intrusion Detection System for IoT

In IoT, HIDS solutions distribute the load when monitoring available hosts on large networks, assigning scalability to the infrastructure. Although these solutions are highly dependent on the operating system and do not have access to the network traffic, host-based approaches should reflect the interactions between the network input and the responsive device, making these systems a mirror and the ultimate goal of the exploited threats. Given the location of a HIDS and the ability to detect indications of external and internal attacks on the system, this solution offers the possibility of being robust to broader vectors of cyber threats and of reducing evasion adversarial threats [82]. Furthermore, as supported by *Liu et al.* [80], research works have shown that modeling system call arguments and return values together provide an unsupervised system, enhance the detection performance, and decrease the false-alarm rate.

4.1. Desired Properties

This section will suggest some considerations to develop an effective, robust, and reliable anomaly detection security system. Based on the fundamental concepts introduced in the previous sections, the following topics are the ones that must be followed to achieve a representative solution in IoT and then lead to the important factors to answer the second research question (RQ2):

Online and Continuous

The IoT principles propose reliable communication to send and receive authentic information, safe and real-time sensor operations, transmissions, and treatments [3]. All these concerns connect the IoT environment with streaming analysis. In this setting, real-time data demands real-time processing, where each instance frequently arrives one instance at a time and needs to be processed at most once. Therefore, these demands constitute the strongest constraint for processing data streams when the allowed time and memory complexities should be limited and constant [83]. Streaming applications like the one proposed in IoT impose unique constraints and challenges for ML models. These applications involve analyzing a continuous, possibly infinite, data sequence occurring in real-time at high speed. This setting demands a fast learning phase as each data record arrives in sequential order where the whole dataset is not available, and storing the entire stream is unattainable [84].

Unsupervised

In cybersecurity, as well as in other domains, the process of generating labels or rules to distinguish normal from abnormal behavior that work as ground truth is expensive, leaving the system completely dependent on domain experts and vulnerable to zero-day attacks [85]. In this sense, a promising solution is to define an unsupervised environment that does not require labeled input and where the objective is to draw inferences about the underlying distribution. This method provides theoretical support and interpretation, building a transparent solution. As these methods rely on probabilistic distributions, distance, or density measures, they support continuous output scores achieved by the degree of such measures. On the other hand, if such assumptions are incorrect, it can mislead the model, making it vulnerable to adversarial attacks and high false alarm rates [86]. From an ML perspective, by combining the predictions of a set of base learners, or individual learners, ensemble methods have shown to be a prominent solution as they offer both flexibility and predictive power [87]. Moreover, this approach becomes more resilient to evasion attacks since the adversary would need to modify an attack pattern to mislead the majority of the classifiers [11].

Dynamic

By definition, streaming data processing incorporates sequential and various data, allowing a more realistic representation of real-world phenomena. As so, it belongs to a dynamic environment where the data distribution can change

over time. Data distributions are subjected to environmental and operational conditions changes, such as different traffic loading, implying a normal behavior may evolve with time, making the current definition of normal behavior no longer acceptable [88]. To account for normal behavior evolution, online anomaly detection in a non-stationary data stream is formulated as a concept drift adaptation problem either solved by data windowing techniques to limit the number of processed data points [89] or incremental learning methods training and updating the model as new events emerge [90].

Contextual Information

The available high-level information, such as contextual information, situational awareness, and cognitive information, combined with experts' judgment of the expected behavior and how it is supposed to react to certain inputs, should be included in the intrusion detection process [91]. While most of the proposed solutions concentrate on monitoring network flow, providing a high-level solution, the suggested detection strategy is to monitor the main asset or target each cyber threat is looking to achieve. In this sense, developing a host-based strategy as a second defense line, where the system performance on the endpoint should mirror what the attacks intend to compromise, designs a resilient approach independent from the source of the attack. For instance, system calls can offer great insights into the tasks a process is performing and the capabilities and accesses it is acquiring. In this sense, they reflect the potential points of failure and, most importantly, represent specific, real-time, and behavioral data to build a real-time resilient security system [43]. Therefore, it is advised to develop a behavioral and context-aware security system that distinguishes events interacting with the infrastructure.

Evaluation Criteria

In a highly imbalanced scenario, it is discouraged to use any potential biased measures to average the models' performance due to the dominating effect of the majority class. In this setting, the performance is fraught by the base-rate fallacy problem often unveiled by the false positive paradox, where the proportion of false positives outnumbers the true positives. Supposing the false-positive rate is higher than the proportion of anomalous samples, the supervisory experts will conclude, from experience, that a positive output indicates an anomaly when, in fact, it is more likely to be a false alarm. In order to fulfill the requirements of an IDS, concepts such as effectiveness, measured by the true positives and false alarm rates; efficiency, considering the time and memory complexities and delay of response actions; collaboration with domain experts so they could identify points of failure, improve the alert description, and guide the model; interoperability between other defense layers to control a broader range and resolve each other's disadvantages [92].

4.2. IDS comparison

Based on the advocated properties for a real-world security system, Table 4 summarizes the review on recent available IDS solutions to attest their ability as a defense line in an IoT environment. Each row identifies recent research solutions for a security system detailed based on the adopted setting featured in each column.

As evidenced, finding an ML-based solution for security systems that would match all requirements and desired properties is challenging. Most solutions design a batch processing in contrast to streaming, continuous, and online solutions, developing a static system that cannot cope with the IoT ecosystem or an adversarial scenario [93, 94, 95, 96, 97, 98, 37, 99]. In particular, solutions such as [93, 94, 100, 101] only focus on detecting a specific type of attack, DDoS, using available datasets that are either old and deprecated or do not represent genuine environments. Most solutions do not show a clear concern about the compatibility between the ML method and the IoT requirements, choosing based on the solution's effectiveness. Although NIDS approaches, like [95, 99], did not consider the imbalanced nature of anomaly detection in IoT, they proposed an ensemble method with the ability to combine different predictions and improve generalizability and robustness against insider attacks. *Milajerdi et al.* [102] presented a solution that aligns attack behavior with kernel audit records, combining both information in behavior graphs and attempting to map cyber threat intelligence indicators of compromise with the system's behavior. Although this procedure implements a score that prioritizes effortful produced flows by attackers to contain evasive threats, the memory and time comparisons reveal a bottleneck on the graph search expansion. This solution implicates a continuous update on the attacks signatures and system records mapped on the graphs, which have not yet been described, classifying this solution as static and vulnerable to zero-days. Moreover, solutions that introduced integration with *Snort* and signature-based collaboration approach [94], or proposed a real-time and lightweight environment to analyze the communications of the IoT service layer [97], validated their methods on both effectiveness, collaboration, and efficiency, respectively. Although most of the reviewed methods define a network system built on deprecated and not realistic datasets, *Lobato et al.* [46] synthesized packet information from honeypots as malicious activities and the incoming flow as the normal pattern, ensuring dynamic and adaptive behavior to detect new threats. While both [97] and [94] network solutions created real test environments and [37] validated on a smart home scenario with adaptive mechanisms to when the network configuration changes, *Chawla et al.* [96] provided a host-based approach evaluated on a predefined balanced system call dataset that does not reflect the working environment or provide any contextual information to resolve intrusion procedures.

Although most of the analyzed network solutions failed to provide a stream processing alternative, *Noble et al.* [45] developed a correlation-based streaming model based on a directed graph structure to capture the communication process of a network router-based protocol. Despite validating their results on an available cybersecurity events dataset, they designed a streaming solution with forgetting factors, which allows a temporally adaptive mechanism with dynamic and continuous properties. *Du et al.* [103] proposed an anomaly detection

Table 4: Reviewed IDS solutions based on the proposed suggestions.

	Online and Continuous	**ML technique**	**Dynamic**	**Contextual**	**Evaluation**
Wang et al. [93]	Batch	Unsupervised	Static	Flow-level features to compare against the overall traffic	Published available datasets to introduce malicious entries. Only validated effectiveness on DDoS.
Deshpande et al. [43]	Streaming Sliding window	Unsupervised	Static	Monitoring system calls and system metrics	Real-time data. Not clear how the anomalous events are being tested. Validated on effectiveness.
Saied et al. [94]	Batch	Supervised	Static	NIDS focused on packet headers.	Training on old and real data with different DDoS attacks. Validated on effectiveness and collaboration with SnortAI.
Noble et al. [45]	Streaming Forgetting factors	Unsupervised	Temporally adaptive	NIDS based on the Netflow router-based protocol.	Large published available dataset. Validated on effectiveness.
Besharati et al. [95]	Batch	Ensemble	Static	NIDS network packet information.	Published available dataset not representative of real-world networks. Validated on effectiveness.
Lobato et al. [46]	Streaming Online	Supervised	Adaptive behavior	NIDS with honeypot traffic packet information	Generated dataset from incoming flows and honeypot data anomalies. Distributed processing module. Validated on effectiveness and efficiency.
Chawla et al. [96]	Batch	Supervised	Static	HIDS based on system calls.	Available dataset of system calls with balanced classes. Validated on effectiveness and efficiency.
Chaabouni et al. [97]	Batch	Supervised	Static	NIDS analyzing the communication model of OneM2M servive layer.	Generated traffic dataset capturing the relations between source and destination. Real IoT test environment. Validated on effectiveness and efficiency.
Khraisat et al. [98]	Batch	Ensemble	Static	NIDS with packet information from an IoT environment	Available dataset with normal and botnet IoT activity. Validated on effectiveness and collaboration.
Eskandari et al. [37]	Batch	Semi-supervised Unsupervised	Adaptive behavior.	NIDS with packet information from an IoT environment	Real-world collected raw packets in an IoT smart home. Validated on effectiveness and efficiency.
Srivastava et al. [99]	Batch	Ensemble	Static	NIDS with raw network packets	Published available dataset with ground-truth anomaly instances. Comparative analysis with other solutions validated on effectiveness.
Milajerdi et al. [102]	Batch	Unsupervised	Static	HIDS audit logs and cyber threat intelligence indicators of compromise	Experiments were conducted on DARPA, simulating enterprise network adversarial scenarios, real public incidents, and benign events to stress-test on false positives. Validated on efficiency and effectiveness.
Du et al. [103]	Streaming	Supervised	Incremental learning	Unlearning anomaly detection through incremental learning for host and network data.	Experiments conducted on real application datasets from filesystem's log entries and network to credit card transactions. Validated on false positive and negative rates.
Wagner et al. [100]	Streaming	Unsupervised	Static	Collaborative NIDS from different network locations and exchanging information.	Experiments conducted on real datasets collected network and routing information from Internet Exchange Points (IXP). Validated on accuracy.
Otoum et al. [56]	Batch	Supervised	Static	Hybrid NIDS combining signature and anomaly-based for known and unknown threats.	Experiments conducted on filtered traffic from IoT gateways and available datasets for a limited attack vector. Validated on efficiency and effectiveness.
Mothukuri et al. [5]	Batch	Federated Ensemble	Static	Federated, decentralized and on-device NIDS, only sharing learned weights with central server.	Modbus-based dataset with traffic from physical devices that lack inbuilt communication protocols. Validated on efficiency and effectiveness.
Shen et al. [104]	Offline Continuous	Reinforcement Game-based	Dynamic	Reinforcement learning in a fog-cloud-based IoT network to detect malware diffusion and prevent privacy leakages.	Real fog-cloud IoT network in a continuous multistage. Numerical experiments on the optimal diffusion probability.
Liu et al. [101]	Offline Continuous	Reinforcement Game-based	Dynamic	Reinforcement learning in sensor edge cloud (SEC) devices to enhance dependable resource allocation against DDoS.	It evaluates the expected utility of the defenders and attackers according to different allocation schemes, attack states, and learning techniques.

solution validated on host, network, and credit card transaction datasets. Although this approach designs a deep-learning model, they use the idea of unlearning in an incremental and continuous setting that allows the model to update and still remember important past events. Despite not specifying a streaming or continuous setting, *Wagner et al.* [100] uses Internet Protocol Flow Information Export (IPFIX) to aggregate the information per flow, without storing the payload of network traffic, and implements a collaborative solution by limiting the amount of information received from the same source and comparing with well-know DDoS attacks to avoid false positives. *Otoum et al.* [56] proposes an anomaly and signature-based IDS to detect both unknown and known threats accurately and efficiently. This hybrid solution debates the challenges imposed by a detection system in an IoT environment, from the detection rate to the processing time, by considering data preprocessing and algorithms and data structures to accelerate signature searching and matching. However, this method introduces an overhead by preprocessing and selecting real-time traffic and instances from a public dataset with limited attack vectors. This method does not envision a nonstationary and unsupervised setting despite combining signature and anomaly solutions by prefiltering known attacks and integrating real-time and real-world traffic. Following a current trend in the literature, *Mothukuri et al.* [5] presents a federated learning-based anomaly detection for IoT security attacks that argues that in many ML-assisted approaches, data is generated at the edge and is transferable to a central server without compromising user privacy. This model trains and keeps the data in local devices, only sharing learned weights to the central server. Although this strategy emphasizes user privacy and evaluates precision, accuracy, and training time, this deep learning ensemble proposes an IoT intrusion system running on an operating system and specialized processing units that are not common to find in lightweight and simple architectural devices. As the urgency of privacy protection in IoT networks has been catching the attention, *Shen et al.* [104] suggests a reinforcement learning method to detect malware diffusion and prevent privacy leakages in fog-cloud-based IoT networks. This IDS and response system employs a signaling game to disclose interactions between smart devices and processing nodes. Despite proposing a real-time evaluation framework by experimenting on an IoT network using fog-cloud-based infrastructure, this method first observes and records data and then analyzes and computes statistical points and optimal probabilities. Furthermore, this work does not validate the efficiency or effectiveness of the detection system, only focusing on which parameters influence the convergence of diffusion and infected probabilities. These latest techniques explored in recent literature manage to incorporate privacy concerns and investigate interactions between devices and the network with cyberattacks, modeling the purpose of cyber threats and the IoT system in a game-theoretic framework. As an example, *Liu et al.* [101] renders the interaction of the IoT network and DDoS attacks as a game where the former seeks to allocate computational resources and the latter tries to hinder the defender from achieving the tasks' demands. Despite examining the interaction and effects of a distributed attack on the host resource consumption, this work only investigates distributed DoS attacks. It attests to a high computational and storage capacity architecture that contradicts the IoT paradigm.

To sum up, based on the analyzed considerations, *Deshpande et al.* [43] presented an unsupervised streaming solution with relevant contextual information regarding the system's interactions that matches the highest number of recommendations, only dismissing dynamic properties of this environment, without considering concept drift, and evaluating on more than effectiveness.

Finally, based on the reviewed proposals and their shortcomings, finding supervised, batch, and static settings in cyber security threat hunting are more recurrent, reflecting the lack of collaboration between intrusion and anomaly detection fields to team up and apply their best techniques. Moreover, this could be one of the reasons these solutions fall short in terms of real-time and appropriate evaluation criteria that leave experts reluctant to deploy such methods.

4.3. How to build a real-time host-based dataset

Scientific advances rely on the reproducibility of results to be independently validated and compared. Many intrusion detection approaches have been evaluated based on proprietary data, and results are generally not reproducible [105]. Due to the rapidly changing nature of cybersecurity, the standardization of evaluation settings and metrics faces new challenges, which have been addressed based on standard datasets with generated traffic of previously known and studied attacks [106].

Currently, proposed security systems fail to design a complete solution that resembles the environment it will be integrated. In fact, they do not provide a real environment training and testing to validate their methods according to effectiveness, efficiency, or collaboration. In essence, some of the biggest limitations include the limited attack coverage, as most researchers focus on specific threats or the lack of real-world simulations since the datasets do not provide contextual information of the expected workload and applications.

Data collection is one of the most important parts of a data-driven solution. This process should be a priority since building a solution on top of a biased, skewed, or not representative work setting produces unfit models. Data is also a mirror of a set of interactions always connected to a certain time and context. As so, in order to fully understand the environment of our task, domain knowledge and data analysis cannot be separated [107].

The following discussion outlines a data collection strategy for a security system. In this sense, a precise definition of the task to accomplish, the data used to reproduce the implicit relations, and the context and concerns to consider are primary steps to detail in a data-driven security solution. Obtaining valid, representative, and accurate data that reflects the context and environment could be the key to building an IDS fit for exploitation.

Considering that a realistic detection dataset should represent the problem at hand, some core requirements must be followed. Primarily, since anomaly detection datasets assemble normal and abnormal instances, both cases should be drawn

from a real-world generating process. Additionally, a benchmark dataset should also obey meaningful problem dimensions, such as relative frequencies between incoming normal and abnormal points, variations so that different concepts and attack vectors are considered, and feature relevance to better describe the observed behavior [108].

Based on a detailed risk assessment and collaboration with domain experts, the data collection framework should analyze patterns to spot potential threats and points of failure. Furthermore, the normal behavior is the product of several underlying events from different stages that could potentially increase the false positive rate. By monitoring the system and considering the system's current state, a broader cyber threat coverage is available, and the IDS becomes more resilient to adversarial attacks [37].

Given that system calls are the virtual interface between an application and the kernel, allowing actions like opening files, creating network connections, reading and writing from and to files, monitoring these calls and processing the information based on customized metrics are good solutions to record critical operations. System calls can offer great insights into the tasks a process is performing and the capabilities and accesses it is acquiring. These perceptions can be invaluable for troubleshooting, monitoring, and bottleneck identification [43]. However, analyzing the entire system call trace results in a slow or late response against the intrusion, compromising the efficiency of the security system [80]. The most important step to achieving stability and security is to separate the operating system core and application programs or user processes. In this sense, it is crucial to be specific and filter processes handling delicate jobs.

Therefore, we suggest filtering system interactions based on system calls that imply tender and potential points of failure. Then, these interactions can be summarized into system metrics used to monitor the system performance throughout its runtime. This data collection solution generated from incoming interactions can be seen as normal instances that provide a real-time and contextual setting. However, it is vital to estimate how the model responds when facing cyber threats to evaluate the security system's effectiveness.

Given the growth and diversity of devices, the consequent lack of investment in security mechanisms, and their exposure to the Internet, connected and interoperable environments like IoT are becoming more susceptible to attacks. From this exposure, vulnerabilities are discovered as software errors and security holes, allowing attackers to take advantage and have access to the infrastructure. These vulnerabilities are defined under the name of CVE, which are cataloged as a list and maintained by the National Cybersecurity Federally Funded Research and Development Center (NFC). Therefore, this database can exploit cyber attacks for different platforms and label them as anomalies to test a security IDS.

As the experimental environment should be designed according to the ecosystem in which the final model is supposed to work, the setup should be configured in line with the specification of an IoT actuator device. Given the architecture of such devices, this setting is also valid for servers or workstations, validating the approaches in a broader range of applications. Therefore, as the designed data collection framework generates information, a considerable number of vulnerabilities can be exploited so that anomalous traffic is produced in a controlled environment. This strategy will provide ground truth for a whole and contextual evaluation scheme, meeting our third research question (RQ3).

5. Open source HIDS evaluation

Practical implementations of detection systems require some factors to be considered, such as attack vectors or the ability of the solution to adapt and recognize new threats. As a result, a review and comparison of different open-source HIDS solutions according to the proposed evaluation scheme in a use case application. This methodology will assign the robustness of the tested solutions by analyzing the specification and general coverage towards previously known attacks.

5.1. Use Case - Smart Campus

The IoT structure enhances the heterogeneity and availability of services and applications that define smart devices used in different consumer and organizational applications such as smart homes, healthcare, smart cities, and smart campuses.

In 2019, *Gartner* identified smart campuses as one of the top 10 strategic technologies impacting higher education. The organization defined a smart campus, a concept that has been applied in developed countries for several years, as "a physical or digital environment in which humans and technology-enabled systems interact to create more immersive and automated experiences for university stakeholders"[3]. Driven by smart campus applications, this IoT use case will instance a continuous environment simulation of a real-life HIDS dataset as envisioned in RQ3.

Sustaining the IoT operational effectiveness and efficiency, the smart campus concept, adapted from smart cities, focuses on smart education, parking, and administration supporting academic services for a more sustainable and successful institution. In this sense, smart campuses design insight-driven decisions to improve security, maximize resources and join people, devices, and applications. However, these smart context properties also reflect security challenges where many attacks aim to compromise the entire infrastructure.

Figure 3 represents the architecture of an IoT smart campus. On the bottom layer, campus information management, the physical and perception layer is represented, showing different data collected in this setting. From parking spaces, mapping technologies, detecting water leaks, controlling humidity, and facial and location intelligence to promoting engaging platforms and inclusive activities, this scenario allows a smart and automated monitoring attendance while maintaining a human-centric approach [109].

[3] https://www.gartner.com/en/newsroom/press-releases/2019-03-26-gartner-identifies-the-top-10-strategic-technologies-(May 2021)

IoT units receive and send wireless or Bluetooth connections responsible for sharing information to ensure communication between devices. These units incorporate hardware, software, and cloud services. The hardware component can be seen as a microcontroller board equipped with sensors, wireless, and other connection pins [110]. Thus, one way to mimic hardware equipment and emulate IoT components behavior in a smart setting is by using Raspberry Pi, a series of low-cost programmable computers. These components have GPIO (General Purpose Input/Output) pins that allow other electronic components to connect and gather information, expanding and integrating their communications, activities, and capabilities in an ad-hoc mode [111].

The information gathered in the perception layer is managed, processed, and routed to a specific application or service through networking protocols, system management tools, or web services and platforms. The management services layer, depicted in Figure 3, is responsible for complex event processing on edge or cloud, integrating streaming and real-time analysis to promote real-time solutions and improve the quality of offered services.

Apart from the physical exposure inbred in the perception layer, the information now handled and transmitted to a different part of the network faces challenging problems related to security and privacy introduced by physical interactions among devices, dealing with dynamic spatial and contextual data, or flawed development features where most of the available communications protocols are not suitable for resource-constraint devices [112]. Therefore, besides the necessity to balance performance and consumed resources, this heterogeneous network should not lower security standards, becoming imperative to provide regular software and firmware updates and secure communications, authentication, as well as user information [55, 60, 79].

Figure 3 now portrays several services, categorized according to the designed application from protocols providing network communication, system management, and services to visualization and monitoring applications without ruling out programming libraries and modules found in many software features. In the ad-hoc layer, services are exposed using HTTP services, such as *nhttpd* or *tomcat*, allowing users to query and manage software using these services and packages. Given the high number of moving users from different backgrounds and needs, the analytic data is fundamental for university employees and students to coordinate room and equipment usage, administration, and resource management. This layer represents important services found in an IoT environment, particularly in a smart campus setting, symbolizing some of its biggest challenges and risks such as outdated firmware, plaintext passwords or communication, development bugs, or default configurations. Such services can be found in known vulnerability datasets as susceptible versions due to poor configuration settings, incorrect parameters, or implementation flaws.

Due to the demand for interconnection and information interchange with technology, IoT systems struggle with security and privacy issues. However, many developers and companies feel reluctant to implement security solutions as available systems

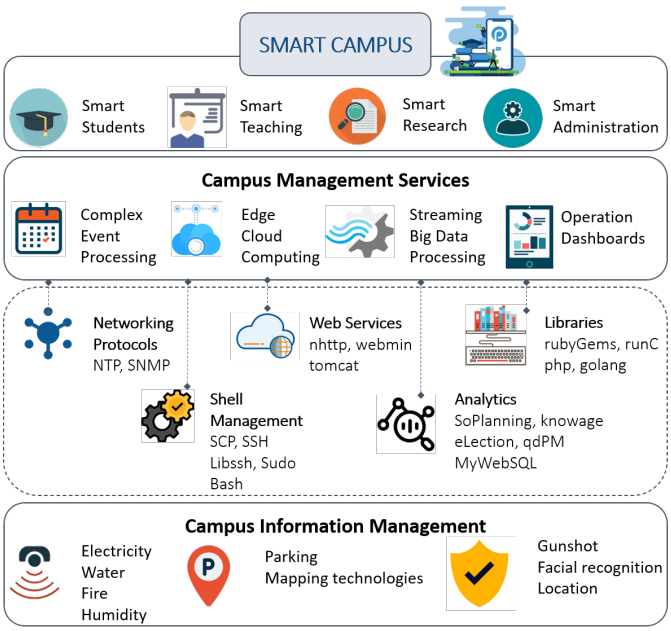

Figure 3: Smart Campus

are too complex for lightweight IoT devices [113].

5.2. Evaluation Scheme

In this study, we will focus on the top used HIDS systems available on the market and have been indicated and reviewed as the main software tools compatible with the ARM architecture [114, 113]: *Samhain* [115], *Tripwire* [20], *Open Source Security (OSSEC)* [19], *Advanced Intrusion Detection Environment (AIDE)* [116], *Sagan* [117] and *Fail2Ban* [118]. Table 5 summarizes the significant aspects of these systems, where the tested vulnerabilities were all consistent with the HIDS stable release version presented in Table 5.

From the tested system, only *OSSEC* designs an open-source solution supported on all major OS platforms [119]. As these tools are host-based applications, they all offer filesystem monitoring by comparing file signatures, like *AIDE* [120], cryptographic hashes, like *TripWire* [121], or by monitoring checksums like *OSSEC* [119]. These solutions provide real-time monitoring tools with policies to customize rules and alerts. While *AIDE* develops a multi-threaded architecture for a lightweight service with easy rule management [122], *Samhain* provides a log file and port monitoring service, rootkit detection, or uncovering hidden processes based on a stealth approach to prevent adversarial attacks [123]. Although most security systems choose to raise an alert when a threatening event is targeting the system, *Fail2Ban* provides a prevention system and active response mechanisms that block IP addresses and become effective against DoS attacks [124]. As described previously, any security system must look to achieve real-time performance. *TripWire*, particularly used to monitor a designated set of files and directories for any changes, does not generate real-time alerts, despite monitoring permissions, internal file changes and timestamps, writing and storing details on a log file [121]. Finally, *OSSEC* is a leading, entirely free, and

a manager/agent architecture HIDS that analyzes information in real-time from event logs, firewall, antivirus logs, and traffic logs, providing active responses using both signature and anomaly detection methods [119].

Table 5: Open-source HIDS

	OS	Features
OSSEC 3.6 Feb2020	Unix, Windows	Log file processing Monitoring firewall and traffic log Defined alerts through policies
Fail2Ban 0.10.2 Jan2018	Unix	Log file monitoring Active response Prevention System
Tripwire 2.4.3.1 Apr2016	Unix	Unauthorized file changes Cryptographic hashes Permissions, internal file changes
Sagan 1.1.2 Jan2019	Unix	Log file monitoring Rule syntax as Snort Lightweight
Samhain 4.4.2 Oct2020	Unix	File integrity checking Port monitoring Stealth monitoring
AIDE 0.16.3 Sep2018	Unix	File signature comparisons File and directory integrity File attributes database

For the evaluation purpose, the setup was configured bearing in mind the hardware and operating system's availability, affordability, and facility. Therefore, following the requirements on how to build a real-time dataset, outlined in Section 4.3, the evaluation procedure of the HIDS was conducted on normal instances, represented by the expected behavior of an IoT ecosystem, expressed in Section 5.1, and abnormal traffic portrayed by cyber threats.

To attend a dynamic and evolving real IoT environment, a similar scenario to the use case described in Section 5.1 will generate system interactions and describe the expected behavior. In this setup, Raspberry Pi 4 devices (model B), with 4 GB of RAM, running Ubuntu 20.04 LTS 64 bits, will be used to replicate smart objects. The tests will be pursued on a client-server and ad-hoc mode, through the WiFi board and an ethernet connection to central servers.

A set of vulnerabilities from different types of attacks, currently threatening IoT infrastructures, will be fed for the anomalous traffic while the system performs its usual tasks. Similar to the idea of a trapdoor-enabled detection, presented in [125], instead of hiding the flaws, the vulnerabilities are expanded, creating adversarial examples that are easier to identify and provide valuable insights. These vulnerabilities are available in the CVE list of entries, which contains an identification number, a description, and, at least, one public reference for publicly known cybersecurity vulnerabilities [126]. These records are typically used in vulnerability scanners, inspecting systems and networks for potential problems, and report the results based on this dictionary of publicly known security exposures, such as works addressing intrusion systems for network data [111, 127] or recent anomaly-based system solutions [37].

With a smart campus environment in mind, described in Section 5.1, the generated anomalies from the CVE dataset, to validate the host-based systems were chosen based on the typical vulnerabilities of these ongoing services and applications an attacker wants to exploit to compromise or gain access to a particular device. As a result, we consider services from networking protocols (i.e., NTP and SNMP), Shell Management (i.e., SSH), Web services (i.e., nhtop, webmin, and tomcat), analytics (i.e., SQL services), and generic vulnerabilities from programming libraries (i.e., PHP) reported in generic software for IoT. Furthermore, according to its impact, the base score assigned to each threat also played an important role in the final decision [128]. In this sense, each CVE explored will be described by the attack vector and its base score, a severity rating, ranging from 0 to 10 (0 - None, $[0.1, 3.9]$ - Low, $[4.0, 6.9]$ - Medium, $[7.0, 8.9]$ - High, $[9.0, 10]$ - Critical). In these trials, all vulnerabilities recorded medium or higher base score levels and were all available in the tested network, even if not present simultaneously.

5.3. Discussion

Table 6 shows the results of evaluation tests of six open-source HIDS, using the CVE vulnerabilities, specified in each row. When a security solution is not designed to detect a particular vulnerability and the required mechanisms are missing, the event is depicted as impractical. The list of CVE with a detailed description and reproduction scripts can be found in a GitHub repository[4].

Considering software injection attacks, in the SQL-injection technique, most of the HIDS were capable of detecting the attack, regardless of their variant. However, *Sagan* and *Samhain* obtained the worst performance. In particular, *Sagan* could not detect either of the attacks, and *Samhain* did not uncover CVE-2020-9268 exploiting the vulnerable OrderBy clause in *SoPlanning* (v. 1.45) application. In fact, only the signature mode employed by these security systems was able to detect the temporary files uploaded by *sqlmap* used to verify and exploit SQL injection flaws [129]. The anomaly-based variant detected the abnormal requests and blocked the connection after 3 to 4 requests, depending on the time difference between requests. However, only *OSSEC* and *Fail2Ban*, implementing signature variants, were able to detect CVE-2019-13189 when monitoring *start_url* and *user_id* fields of *Knowage* application. The anomaly-based approach failed to detect XSS and XEE attacks in all systems.

Regarding vulnerabilities threatening the confidentiality principle, when exposed to RCE attacks, the most general result was the successful detection of the attack except for CVE-2019-9624, detected only by *TripWire*. This vulnerability, which allows the execution of arbitrary code by leveraging the Java

[4]Available after the publication: https://github.com/simao-silva/iot-cves
[5]https://nvd.nist.gov/vuln/detail/[ID]

Table 6: Experimental trials

Attack	CVE	Base Score	OSSEC	Fail2Ban	TripWire	Sagan	Samhain	AIDE
XSS	CVE-2019-13189[5]	6.1	●	●	⊗	○	⊗	⊗
XXE	CVE-2019-15641[5]	6.5	○	○	⊗	○	⊗	⊗
SQL-Injection	CVE-2020-9340[5]	7.2	●	●	●	○	●	●
	CVE-2020-9268[5]	7.5	●	●	●	○	○	●
Bypass Control	CVE-2019-13188[5]	9.8	○	○	○	○	○	○
Improper file access	CVE-2018-8712[5]	9.8	●	●	●	⊗	○	○
Unauthorized file real	CVE-2020-1938[5]	9.8	●	○	⊗	○	⊗	⊗
Privilege Escalation	CVE-2019-14287[5]	8.8	●	○	⊗	○	⊗	⊗
	CVE-2019-9891[5]	9.8	⊗	⊗	⊗	⊗	⊗	⊗
	CVE-2019-8320[5]	7.4	●	○	⊗	○	⊗	⊗
	CVE-2019-5736[5]	8.6	●	○	⊗	○	⊗	⊗
	CVE-2018-10933[5]	9.1	●	○	⊗	○	⊗	⊗
	CVE-2019-18634[5]	7.8	⊗	⊗	⊗	⊗	⊗	⊗
RCE	CVE-2020-7246[5]	8.8	●	●	●	○	●	●
	CVE-2019-16278[5]	9.8	●	●	●	○	●	●
	CVE-2019-15642[5]	8.8	●	●	●	○	●	●
	CVE-2019-15107[5]	9.8	●	●	●	○	●	●
	CVE-2019-12840[5]	8.8	●	●	●	○	●	●
	CVE-2019-11043[5]	8.7	●	●	●	○	●	●
	CVE-2019-9624[5]	7.8	⊗	⊗	●	⊗	○	●
	CVE-2019-7731[5]	9.8	●	●	●	○	●	●
DoS	CVE-2020-9283[5]	7.5	○	○	⊗	○	⊗	⊗
	CVE-2020-6060[5]	7.5	○	○	⊗	○	⊗	⊗
	CVE-2019-17498[5]	8.1	○	○	⊗	○	⊗	⊗
	CVE-2019-16279[5]	7.5	●	●	⊗	○	⊗	⊗
	CVE-2019-13115[5]	8.1	○	○	⊗	○	⊗	⊗
	CVE-2018-7182[5]	7.5	○	○	⊗	○	⊗	⊗

● Detected ○ Undetected ⊗ Impractical

file manager on *webmin* 1.9, is uncovered only if the partition was mounted with *strictatime* option, explicitly requesting full *atime* updates. Only the anomaly-based approach was considered in privilege escalation threats since signature-based IDS does not have the features required to detect this attack. In most cases, only *OSSEC* was able to spot this attack vector. The results show that CVE-2019-9891, where the shell function *getopt_simple* allows execution of attacker-controlled commands, and CVE-2019-18634 that triggers a stack-based buffer overflow when the *pwfeedback* option to allow password visual feedback [130] is enabled in *Sudo* (v<1.8.26) were not detected by any system. Since these vulnerabilities score a high severity value, these results show that the most common solutions cannot cope with some of the most concerning threats.

Related to jeopardizing the availability principle, DoS attacks have been tested only considering anomaly-based approaches. In most cases, most HIDS showed their inability to cope with these attacks by either failing to detect or not designing a solution robust enough to deal with these specifications. Particularly, in all exploits of CVE-2019-16279, where a memory error in the function *SSL_accept* in *nostromo nhttpd* through 1.9.6 is used to trigger a DoS attack via a crafted HTTP request, we were able to disrupt the service without being detected by the security systems.

In essence, HIDS systems with a monitoring approach and predefined rules were only able to detect massively used attacks such as brute-force or SQL injection. Moreover, when anomaly-based detection is integrated, and customized rules are generated, HIDS solutions become more successful and robust to RCE attacks.

Regarding the algorithms used for file integrity, many systems still use deprecated algorithms vulnerable to collision attacks. In the tested HIDS, popular file integrity algorithms MD5 and SHA1 were compromised when known attacks like *SHAttered* were applied. If the HIDS only relies on this mechanism, these attacks can be imperceptible, making the security system obsolete. Therefore, applying different methods and considering their merits and demerits show that a hybrid solution like *OSSEC* is a promising direction.

6. Future research directions

In an interconnected infrastructure that provides diverse functionalities and services according to user requests and data feedback, security and privacy concerns of physical objects inflict crucial requirements.

Intrusion detection systems in IoT should account for broader attack vectors and adversarial threats in a streaming environment to develop a cost-sensitive and context-aware method that analyzes potential targets and damage costs. In this sense, the following procedure will focus on implementing a security system that follows the proposed guidelines to design a realistic and resilient approach from the data processing and methodology to its context-aware and evaluation settings.

The idealized strategy of a hybrid detection system formed by signature and contextual host analysis as a first and second defense line, as well as the urge to reduce false alarm rates and increase robustness with human expertise to identify points of failure, hidden relations, or improve alert descriptions, is a promising future direction.

In recent approaches in the literature, reinforcement learning is gaining more popularity [131]. This technique can learn the environment, investigate interactions, and dynamically adapt parameters on the fly by associating possible outcomes with detection, diffusion, and privacy gain and costs [104, 101]. Although this strategy is gaining attention, most attempts design their methods in architectures and devices with high computational capacities to perform computational-intensive tasks that are incompatible with IoT infrastructures. Thus, this alternative is also a potential future direction.

Another challenge that has not been addressed enough is the privacy demand that smart applications handling user data and behavior patterns impose. Privacy-preserving embodies authentication, data collection, processing, sharing, and storage from access control mechanisms to encryption schemes. However, while applying privacy measures to reduce information leakage, an IoT system must always guarantee its practicality [132].

E-commerce solutions [133], location-based services [134], and information retrieval applications [135] use a dummy-based approach for privacy protection, whose basic idea is to use well-designed dummy queries to cover up user queries and protect the real request. Moreover, all proposals recognize the importance of not compromising the accuracy and efficiency of

the services from dummy queries to secure query index through searchable encryption ideas and stemmer mechanisms [136].

Furthermore, a current trend towards ensuring accuracy and privacy policies, denominated federated learning-based, have been discussed in the literature. Federated learning is a collaborative ML technique that does not require centralized training data in one device or a data center. For IoT security, this approach trains and keeps the data in local devices, only sharing learned weights to the central server and ensuring privacy [5]. As evidenced in other ML techniques, this alternative develops its strategy in operating systems and specialized processing units that are not common to find in lightweight and simple architectural devices as encountered in IoT. Moreover, as new techniques and strategies emerge, new methods and possibilities surface to protect IoT security in modern applications that commonly fail to match all the demands enforced by the environment and the task at hand.

7. Conclusion

The objective and priority of this work are to reinforce the importance of defining the main domain characteristics for a better and robust real-time solution. The potential risks and the ultimate purpose of each threat for a clear perception of the scope and complexity of the working environment complete the first research question (RQ1).

Anomaly detection solutions in IoT mainly focus on comparisons with hashes or log file entries based on the expected behavior, failing to uncover hidden patterns, learn from experience, and potentially detect future threat vectors. In this sense, base recommendations to develop a security system regarding the inherent attributes were proposed to answer RQ2.

While anomaly detection proposals do not consider the IoT context and limitations about heterogeneous and fast formats with memory and time restrictions, cybersecurity does not evaluate or collect the right datasets, choosing to design solutions focused on common patterns and known threats. This work delineates a third research question (RQ3) by presenting a real-time and contextual dataset with the ongoing traffic and operations reflecting the normal behavior and the exploited vulnerabilities from CVE lists as anomalies.

When exploiting some of the latest vulnerabilities to the most common open-source HIDS, these systems appear limited. They do not embrace all services or applications, focusing on those popularly installed in systems. In fact, the services are expected to be installed with default parameters or will be forced to add custom rules or modify the available ones.

As a result, a real-time and complete representative intrusion detection system in IoT is a demanding and urgent solution where the main research challenges remain.

Acknowledgement

The work of Inês Martins has been supported by Fundação para a Ciência e Tecnologia (FCT), Portugal - 2021.04908.BD, and partially funded by the SafeCities POCI-01-0247-FEDER-041435 project through COMPETE 2020 program.

The work of João S. Resende has been supported by the EU H2020-SU-ICT-03-2018 Project No. 830929 CyberSec4Europe.

The work of Patrícia R. Sousa has been supported by the Project "City Catalyst – Catalisador para cidades sustentáveis", with reference POCI-01-0247-FEDER-046112, financed by Fundo Europeu de Desenvolvimento Regional (FEDER), through COMPETE 2020 and Portugal 2020 programs.

The work of Simão Silva was partially funded by the SafeCities POCI-01-0247-FEDER-041435 project through COMPETE 2020 program.

The work of João Gama was partially supported by the European Commission-funded project Humane AI: Toward AI Systems That Augment and Empower Humans by Understanding Us, our Society and the World Around Us.

The work of Luís Antunes has been supported by the Project "CNCS - Centro Nacional de Cibersegurança - Serviço de Gestão Alargada do Conhecimento Situacional e Operacional do Ciberespaço Nacional", with reference POCI-05-5762-FSE-000229, financed by Agência para a Modernização Administrativa.

All the supports mentioned above are gratefully acknowledged.

References

[1] Y. Lu, L. D. Xu, Internet of things (iot) cybersecurity research: A review of current research topics, IEEE Internet of Things Journal 6 (2) (2019) 2103–2115. doi:10.1109/JIOT.2018.2869847.
URL https://ieeexplore.ieee.org/document/8462745

[2] S. L. Keoh, S. S. Kumar, H. Tschofenig, Securing the internet of things: A standardization perspective, IEEE Internet of things Journal 1 (3) (2014) 265–275.

[3] M. Cherian, M. Chatterjee, Survey of security threats in iot and emerging countermeasures, International Symposium on Security in Computing and Communication (2019) 591–604doi:10.1007/978-981-13-5826-5_46.

[4] L. Leenen, T. Meyer, Artificial intelligence and big data analytics in support of cyber defense, Developments in Information Security and Cybernetic Wars (2019) 42–63doi:10.4018/978-1-5225-8304-2.ch002.

[5] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, G. Srivastava, Federated learning-based anomaly detection for iot security attacks, IEEE Internet of Things Journal (2021).

[6] H. Hindy, D. Brosset, E. Bayne, A. Seeam, C. Tachtatzis, R. Atkinson, X. Bellekens, A taxonomy of network threats and the effect of current datasets on intrusion detection systems, IEEE Access PP (2020) 1–1. doi:10.1109/ACCESS.2020.3000179.

[7] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, K.-Y. Tung, Intrusion detection system: A comprehensive review, Journal of Network and Computer Applications 36 (1) (2013) 16–24.

[8] V. Hajisalem, S. Babaie, A hybrid intrusion detection system based on abc-afs algorithm for misuse and anomaly detection, Computer Networks 136 (2018) 37–50.

[9] P. Branco, L. Torgo, R. P. Ribeiro, A survey of predictive modeling on imbalanced domains, ACM Computing Surveys (CSUR) 49 (2) (2016) 1–50.

[10] A. Adnan, A. Muhammed, A. A. Abd Ghani, A. Abdullah, F. Hakim, An intrusion detection system for the internet of things based on machine learning: Review and challenges, Symmetry 13 (6) (2021) 1011.

[11] I. Corona, G. Giacinto, F. Roli, Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues, Information Sciences 239 (2013) 201–225.

[12] N. M. Kumar, P. K. Mallick, The internet of things: Insights into the building blocks, component interactions, and architecture layers, Procedia Computer Science 132 (2018) 109–117. doi:10.1016/j.procs.2018.05.170.

[13] T. Elsaleh, M. Bermudez-Edo, S. Enshaeifar, S. T. Acton, R. Rezvani, P. Barnaghi, Iot-stream: A lightweight ontology for internet of things data streams, 2019 Global IoT Summit (GIoTS) 132 (2019) 1–6. doi:10.1109/GIOTS.2019.8766367.

[14] P. H. dos Santos Teixeira, R. L. Milidiú, Data stream anomaly detection through principal subspace tracking, in: Proceedings of the 2010 ACM Symposium on Applied Computing, 2010, pp. 1609–1616.

[15] P. Mulinka, P. Casas, Stream-based machine learning for network security and anomaly detection, Proceedings of the 2018 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks (2018) 1–7doi:10.1145/3229607.3229612.

[16] S. Sen, A survey of intrusion detection systems using evolutionary computation, in: Bio-inspired computation in telecommunications, Elsevier, 2015, pp. 73–94.

[17] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, B. D. Payne, Evaluating computer intrusion detection systems: A survey of common practices, ACM Computing Surveys (CSUR) 48 (1) (2015) 1–41.

[18] E. Vasilomanolakis, S. Karuppayah, M. Mühlhäuser, M. Fischer, Taxonomy and survey of collaborative intrusion detection, ACM Computing Surveys (CSUR) 47 (4) (2015) 1–33.

[19] R. Bray, D. Cid, A. Hay, OSSEC host-based intrusion detection guide, Syngress, 2008.

[20] Tripwire, Cybersecurity for enterprise and industrial organizations, https://www.tripwire.com/ [Accessed: October 28, 2020] (2020).

[21] Snort, Snort - network intrusion detection and prevention system, https://snort.org/ [Accessed: October 28, 2020] (2020).

[22] Suricata, Suricata - open source ids and ips, https://suricata-ids.org/ [Accessed: October 28, 2020] (2020).

[23] E. Albin, N. C. Rowe, A realistic experimental comparison of the suricata and snort intrusion-detection systems, in: 2012 26th International Conference on Advanced Information Networking and Applications Workshops, IEEE, 2012, pp. 122–127.

[24] N. Das, T. Sarkar, Survey on host and network based intrusion detection system, International Journal of Advanced Networking and Applications 6 (2) (2014) 2266.

[25] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, Survey of intrusion detection systems: techniques, datasets and challenges, Cybersecurity 2 (1) (2019) 1–22.

[26] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, S. C. de Alvarenga, A survey of intrusion detection in internet of things, Journal of Network and Computer Applications 84 (2017) 25–37.

[27] P. Uppuluri, R. Sekar, Experiences with specification-based intrusion detection, in: International Workshop on Recent Advances in Intrusion Detection, Springer, 2001, pp. 172–189.

[28] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu, V. C. Leung, A survey on security threats and defensive techniques of machine learning: A data driven view, IEEE access 6 (2018) 12103–12117.

[29] A. L. Buczak, E. Guven, A survey of data mining and machine learning methods for cyber security intrusion detection, IEEE Communications surveys & tutorials 18 (2) (2015) 1153–1176.

[30] P. Aravamudhan, T. Kanimozhi, A survey on intrusion detection system and prerequisite demands in iot networks, in: Journal of Physics: Conference Series, Vol. 1916, IOP Publishing, 2021, p. 012179.

[31] W. Li, W. Meng, M. H. Au, Enhancing collaborative intrusion detection via disagreement-based semi-supervised learning in iot environments, Journal of Network and Computer Applications 161 (2020) 102631.

[32] J. Guo, A. Marshall, B. Zhou, A new trust management framework for detecting malicious and selfish behaviour for mobile ad hoc networks, in: 2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, IEEE, 2011, pp. 142–149.

[33] E. Vasilomanolakis, S. Karuppayah, M. Mühlhäuser, M. Fischer, Taxonomy and survey of collaborative intrusion detection, ACM Computing Surveys (CSUR) 47 (4) (2015) 1–33.

[34] D. M. Hawkins, Identification of outliers, Monographs on statistics and applied probability (1980) 1–194.
URL https://link.springer.com/content/pdf/10.1007/978-94-015-3994-4.pdf

[35] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM computing surveys (CSUR) 41 (3) (2009) 1–58.

[36] S. A. Aljawarneh, R. Vangipuram, Garuda: Gaussian dissimilarity measure for feature representation and anomaly detection in internet of things, The Journal of Supercomputing 76 (6) (2020) 4376–4413.

[37] M. Eskandari, Z. H. Janjua, M. Vecchio, F. Antonelli, Passban ids: An intelligent anomaly-based intrusion detection system for iot edge devices, IEEE Internet of Things Journal 7 (8) (2020) 6882–6897. doi:10.1109/JIOT.2020.2970501.

[38] P. A. A. Resende, A. C. Drummond, A survey of random forest based methods for intrusion detection systems, ACM Computing Surveys (CSUR) 51 (3) (2018) 1–36.

[39] J. Gama, A survey on learning from data streams: current and future trends, Progress in Artificial Intelligence 1 (1) (2012) 45–55.

[40] A. Bifet, J. Read, G. Holmes, B. Pfahringer, Streaming data mining with massive online analytics (moa), Data Mining in Time Series and Streaming (2018) 1–25doi:10.1142/9789813228047_0001.

[41] L. Yang, A. Shami, A lightweight concept drift detection and adaptation framework for iot data streams, IEEE Internet of Things Magazine (2021).

[42] L. Yang, W. Guo, Q. Hao, A. Ciptadi, A. Ahmadzadeh, X. Xing, G. Wang, {CADE}: Detecting and explaining concept drift samples for security applications, in: 30th {USENIX} Security Symposium ({USENIX} Security 21), 2021.

[43] P. Deshpande, S. C. Sharma, S. K. Peddoju, S. Junaid, Hids: A host based intrusion detection system for cloud computing environment, International Journal of System Assurance Engineering and Management 9 (3) (2018) 567–576.

[44] A. Yahyaoui, T. Abdellatif, R. Attia, Hierarchical anomaly based intrusion detection and localization in iot, in: 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), IEEE, 2019, pp. 108–113.

[45] J. Noble, N. M. Adams, Correlation-based streaming anomaly detection in cyber-security, in: 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), 2016, pp. 311–318. doi:10.1109/ICDMW.2016.0051.

[46] A. G. P. Lobato, M. A. Lopez, I. J. Sanz, A. A. Cardenas, O. C. M. Duarte, G. Pujolle, An adaptive real-time architecture for zero-day threat detection, in: 2018 IEEE international conference on communications (ICC), IEEE, 2018, pp. 1–6.

[47] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, G. Bing, Learning from class-imbalanced data: Review of methods and applications, Expert Systems with Applications 73 (2017) 220–239.

[48] C. Promper, D. Engel, R. C. Green, Anomaly detection in smart grids with imbalanced data methods, in: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2017, pp. 1–8.

[49] I. Ullah, Q. H. Mahmoud, A hybrid model for anomaly-based intrusion detection in scada networks, in: 2017 IEEE International Conference on Big Data (Big Data), IEEE, 2017, pp. 2160–2167.

[50] J. M. Beaver, R. C. Borges-Hink, M. A. Buckner, An evaluation of machine learning methods to detect malicious scada communications, in: 2013 12th international conference on machine learning and applications, Vol. 2, IEEE, 2013, pp. 54–59.

[51] C. Kreibich, J. Crowcroft, Honeycomb: Creating intrusion detection signatures using honeypots, SIGCOMM Comput. Commun. Rev. 34 (1) (2004) 51–56. doi:10.1145/972374.972384.
URL https://doi.org/10.1145/972374.972384

[52] M. Zolanvari, M. A. Teixeira, R. Jain, Effect of imbalanced datasets on security of industrial iot using machine learning, in: 2018 IEEE International Conference on Intelligence and Security Informatics (ISI), IEEE, 2018, pp. 112–117.

[53] J. Sharma, C. Giri, O.-C. Granmo, M. Goodwin, Multi-layer intrusion detection system with extratrees feature selection, extreme learning machine ensemble, and softmax aggregation, EURASIP Journal on Information Security 2019 (1) (2019) 1–16.

[54] H. Bostani, M. Sheikhan, Hybrid of anomaly-based and specification-based ids for internet of things using unsupervised opf based on mapreduce approach, Computer Communications 98 (2017) 52–71.

[55] W. Ding, H. Hu, L. Cheng, Iotsafe: Enforcing safety and security policy with real iot physical interaction discovery, in: 28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually,

February 21-25, 2021, The Internet Society, 2021.
URL https://www.ndss-symposium.org/ndss-paper/iotsafe-enforcing-safety-and-security-policy-with-real-iot-physical-interaction-discovery/

[56] Y. Otoum, A. Nayak, As-ids: Anomaly and signature based ids for the internet of things, Journal of Network and Systems Management 29 (3) (2021) 1–26.

[57] W. Zhou, Y. Jia, A. Peng, Y. Zhang, P. Liu, The effect of iot new features on security and privacy: New threats, existing solutions, and challenges yet to be solved, IEEE Internet of Things Journal 6 (2) (2019) 1606–1616. doi:10.1109/JIOT.2018.2847733.

[58] B. Ali, A. I. Awad, Cyber and physical security vulnerability assessment for iot-based smart homes, Sensors 18 (2018) 1–17. doi:10.3390/s18030817.

[59] L. Shinder, M. Cross, Chapter 12 - understanding cybercrime prevention, in: L. Shinder, M. Cross (Eds.), Scene of the Cybercrime (Second Edition), second edition Edition, Syngress, Burlington, 2008, pp. 505 – 554. doi:https://doi.org/10.1016/B978-1-59749-276-8.00012-1.
URL http://www.sciencedirect.com/science/article/pii/B9781597492768000121

[60] M. A. Khan, K. Salah, Iot security: Review, blockchain solutions, and open challenges, Future generation computer systems 82 (2018) 395–411.

[61] H. Plate, C. Basile, S. Paraboschi, Policy-driven system management, in: Computer and information security handbook, Elsevier, 2013, pp. 427–460.

[62] Y. Liang, H. V. Poor, S. Shamai, Information theoretic security, Now Publishers Inc, 2009.

[63] OWASP, Owasp testing guide appendix c: Fuzz vectors, https://wiki.owasp.org/index.php/OWASP_Testing_Guide_Appendix_C:_Fuzz_Vectors#SQL_Injection [Accessed: January 05, 2021] (2014).

[64] S. Rizvi, A. Kurtz, J. Pfeffer, M. Rizvi, Securing the internet of things (iot): A security taxonomy for iot, 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE) (2018) 163–168doi:10.1109/TrustCom/BigDataSE.2018.00034.

[65] C. Yang, C. Yang, T. Peng, X. Yang, W. Gui, A fault-injection strategy for traction drive control systems, IEEE Transactions on Industrial Electronics 64 (7) (2017) 5719–5727.

[66] A.-u. Rehman, S. U. Rehman, H. Raheem, Sinkhole attacks in wireless sensor networks: A survey, Wireless Personal Communications 106 (4) (2019) 2291–2313.

[67] M. Nawir, A. Amir, N. Yaakob, O. B. Lynn, Internet of things (iot): Taxonomy of security attacks, in: 2016 3rd International Conference on Electronic Design (ICED), 2016, pp. 321–326. doi:10.1109/ICED.2016.7804660.

[68] Cybersecurtiy, I. S. Agency, Security tip (st04-015) - understanding denial-of-service attacks, https://us-cert.cisa.gov/ncas/tips/ST04-015 [Accessed: January 01, 2021].

[69] C. C. S. R. Center, Cybersecurity for enterprise and industrial organizations, https://csrc.nist.gov/glossary/term/man_in_the_middle_attack [Accessed: July 09, 2021].

[70] K. L. Chiew, K. S. C. Yong, C. L. Tan, A survey of phishing attacks: Their types, vectors and technical approaches, Expert Systems with Applications 106 (2018) 1–20.

[71] N. Alsaedi, F. Hashim, A. Sali, F. Z. Rokhani, Detecting sybil attacks in clustered wireless sensor networks based on energy trust system (ets), Computer communications 110 (2017) 75–82.

[72] T. Loise, X. Devroey, G. Perrouin, M. Papadakis, P. Heymans, Towards security-aware mutation testing, in: 2017 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), IEEE, 2017, pp. 97–102.

[73] M. M. Hassan, U. Mustain, S. Khatun, M. S. A. Karim, N. Nishat, M. Rahman, Quantitative assessment of remote code execution vulnerability in web apps, in: InECCE2019, Springer, 2020, pp. 633–642.

[74] I. Andrea, C. Chrysostomou, G. Hadjichristofi, Internet of things: Security vulnerabilities and challenges, 2015 IEEE Symposium on Computers and Communication (ISCC) (2015) 180–187.

[75] J. Deogirikar, A. Vidhate, Security attacks in iot: A survey, in: 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), IEEE, 2017, pp. 32–37.

[76] M. Frustaci, P. Pace, G. Aloi, G. Fortino, Evaluating critical security issues of the iot world: Present and future challenges, IEEE Internet of things journal 5 (4) (2017) 2483–2495.

[77] M. Gregg, S. Watkins, G. Mays, C. Ries, R. M. Bandes, B. Franklin, Hack the stack: Using snort and ethereal to master the 8 layers of an insecure network, Elsevier, 2006.

[78] D. Antonioli, N. O. Tippenhauer, K. Rasmussen, Bias: bluetooth impersonation attacks, in: 2020 IEEE Symposium on Security and Privacy (SP), IEEE, 2020, pp. 549–562.

[79] J. C. S. Sicato, S. K. Singh, S. Rathore, J. H. Park, A comprehensive analyses of intrusion detection system for iot environment, Journal of Information Processing Systems 16 (4) (2020) 975–990.

[80] M. Liu, Z. Xue, X. Xu, C. Zhong, J. Chen, Host-based intrusion detection system with system calls: Review and future trends, ACM Computing Surveys (CSUR) 51 (5) (2018) 1–36.

[81] A. Khraisat, A. Alazab, A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges, Cybersecurity 4 (1) (2021) 1–27.

[82] N. Das, T. Sarkar, Survey on host and network based intrusion detection system, International Journal of Advanced Networking and Applications 6 (2) (2014) 2266.

[83] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak, F. Herrera, A survey on data preprocessing for data stream mining: Current status and future directions, Neurocomputing 239 (2017) 39–57.

[84] S. Ahmad, A. Lavin, S. Purdy, Z. Agha, Unsupervised real-time anomaly detection for streaming data, Neurocomputing 262 (2017) 134–147.

[85] Y. Dong, N. Japkowicz, Threaded ensembles of supervised and unsupervised neural networks for stream learning, in: R. Khoury, C. Drummond (Eds.), Advances in Artificial Intelligence, Springer International Publishing, Cham, 2016, pp. 304–315.

[86] I. Sutskever, R. Jozefowicz, K. Gregor, D. Rezende, T. Lillicrap, O. Vinyals, Towards principled unsupervised learning, arXiv preprint arXiv:1511.06440 (2015).

[87] D. B. Araya, K. Grolinger, H. F. ElYamany, M. A. Capretz, G. Bitsuamlak, An ensemble learning framework for anomaly detection in building energy consumption, Energy and Buildings 144 (2017) 191–206.

[88] H. Tian, N. L. D. Khoa, A. Anaissi, Y. Wang, F. Chen, Concept drift adaption for online anomaly detection in structural health monitoring, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 2813–2821.

[89] M. Salehi, L. Rashidi, A survey on anomaly detection in evolving data: [with application to forest fire risk prediction], ACM SIGKDD Explorations Newsletter 20 (1) (2018) 13–23.

[90] P. Alaei, F. Noorbehbahani, Incremental anomaly-based intrusion detection system using limited labeled data, in: 2017 3th International Conference on Web Research (ICWR), IEEE, 2017, pp. 178–184.

[91] F. J. Aparicio-Navarro, K. G. Kyriakopoulos, Y. Gong, D. J. Parish, J. A. Chambers, Using pattern-of-life as contextual information for anomaly-based intrusion detection systems, IEEE Access 5 (2017) 22177–22193.

[92] S. Axelsson, The base-rate fallacy and the difficulty of intrusion detection, ACM Transactions on Information and System Security (TISSEC) 3 (3) (2000) 186–205.

[93] J. Wang, L. Yang, J. Wu, J. H. Abawajy, Clustering analysis for malicious network traffic, in: 2017 IEEE International Conference on Communications (ICC), IEEE, 2017, pp. 1–6.

[94] A. Saied, R. E. Overill, T. Radzik, Detection of known and unknown ddos attacks using artificial neural networks, Neurocomputing 172 (2016) 385–393.

[95] E. Besharati, M. Naderan, E. Namjoo, Lr-hids: logistic regression host-based intrusion detection system for cloud environments, Journal of Ambient Intelligence and Humanized Computing 10 (9) (2019) 3669–3692.

[96] A. Chawla, B. Lee, S. Fallon, P. Jacob, Host based intrusion detection system with combined cnn/rnn model, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2018, pp. 149–158.

[97] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, An intrusion detection system for the onem2m service layer based on edge machine

learning, in: International Conference on Ad-Hoc Networks and Wireless, Springer, 2019, pp. 508–523.

[98] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, A. Alazab, A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks, Electronics 8 (11) (2019) 1210.

[99] G. Srivastava, N. Deepa, B. Prabadevi, P. K. Reddy M, An ensemble model for intrusion detection in the internet of softwarized things, in: Adjunct Proceedings of the 2021 International Conference on Distributed Computing and Networking, 2021, pp. 25–30.

[100] D. Wagner, D. Kopp, M. Wichtlhuber, C. Dietzel, O. Hohlfeld, G. Smaragdakis, A. Feldmann, United We Stand: Collaborative Detection and Mitigation of Amplification DDoS Attacks at Scale, in: Proceedings of ACM CCS 2021, Virtual Event, 2021.

[101] J. Liu, X. Wang, S. Shen, G. Yue, S. Yu, M. Li, A bayesian q-learning game for dependable task offloading against ddos attacks in sensor edge cloud, IEEE Internet Things J. 8 (9) (2021) 7546–7561. doi:10.1109/JIOT.2020.3038554.
URL https://doi.org/10.1109/JIOT.2020.3038554

[102] S. M. Milajerdi, B. Eshete, R. Gjomemo, V. Venkatakrishnan, Poirot: Aligning attack behavior with kernel audit records for cyber threat hunting, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 1795–1812.

[103] M. Du, Z. Chen, C. Liu, R. Oak, D. Song, Lifelong anomaly detection through unlearning, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 1283–1297.

[104] S. Shen, L. Huang, H. Zhou, S. Yu, E. Fan, Q. Cao, Multistage signaling game-based optimal detection strategies for suppressing malware diffusion in fog-cloud-based iot networks, IEEE Internet Things J. 5 (2) (2018) 1043–1054. doi:10.1109/JIOT.2018.2795549.
URL https://doi.org/10.1109/JIOT.2018.2795549

[105] M. A. Aydın, A. H. Zaim, K. G. Ceylan, A hybrid intrusion detection system design for computer network security, Computers & Electrical Engineering 35 (3) (2009) 517–526.

[106] J. Ernst, T. Hamed, S. Kremer, A survey and comparison of performance evaluation in intrusion detection systems, in: Computer and network security essentials, Springer, 2018, pp. 555–568.

[107] M. A. Waller, S. E. Fawcett, Data science, predictive analytics, and big data: a revolution that will transform supply chain design and management, Journal of Business Logistics 34 (2) (2013) 77–84.

[108] A. Emmott, S. Das, T. G. Dietterich, A. Fern, W. Wong, Systematic construction of anomaly detection benchmarks from real data, CoRR abs/1503.01158 (2015). arXiv:1503.01158.
URL http://arxiv.org/abs/1503.01158

[109] P. Martins, S. I. Lopes, A. M. Rosado da Cruz, A. Curado, Towards a smart & sustainable campus: An application-oriented architecture to streamline digitization and strengthen sustainability in academia, Sustainability 13 (6) (2021) 3189.

[110] M. W. Sari, P. W. Ciptadi, R. H. Hardyanto, Study of smart campus development using internet of things technology, in: IOP Conference Series: Materials Science and Engineering, Vol. 190, IOP Publishing, 2017, p. 012032.

[111] A. Sforzin, F. G. Mármol, M. Conti, J.-M. Bohli, Rpids: Raspberry pi ids—a fruitful intrusion detection system for iot, in: 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), IEEE, 2016, pp. 440–448.

[112] A. Alrawais, A. Alhothaily, C. Hu, X. Cheng, Fog computing for the internet of things: Security and privacy issues, IEEE Internet Computing 21 (2) (2017) 34–42.

[113] R.-L. J. Jesús, P.-V. O. Cristhian, R.-G. M. René, F.-M. Heberto, How to improve the iot security implementing ids/ips tool using raspberry pi 3b, Editorial Preface From the Desk of Managing Editor... 10 (9) (2019).

[114] S. Cooper, Intrusion detection systems explained: 13 best ids software tools reviewed, https://www.comparitech.com/net-admin/network-intrusion-detection-tools/ [Accessed: March 11, 2021] (2020).

[115] Samhain, Samhain - the samhain file integrity/host-based intrusion detection system, https://la-samhna.de/samhain/ [Accessed: February 06, 2020].

[116] AIDE, Aide (advanced intrusion detection environment), https://aide.github.io/ [Accessed: February 06, 2020].

[117] quadrantsec, Sagan, https://github.com/quadrantsec/sagan [Accessed: February 06, 2020].

[118] Fail2Ban, Fail2ban, https://www.fail2ban.org [Accessed: February 06, 2020].

[119] R. Bray, D. Cid, A. Hay, OSSEC host-based intrusion detection guide, Syngress, 2008.

[120] C. L. Smith, Aide-advanced intrusion detection environment, Tech. rep., Pacific Northwest National Lab.(PNNL), Richland, WA (United States) (2013).

[121] G. H. Kim, E. H. Spafford, The design and implementation of tripwire: A file system integrity checker, in: Proceedings of the 2nd ACM Conference on Computer and Communications Security, 1994, pp. 18–29.

[122] T. Zitta, M. Neruda, L. Vojtech, The security of rfid readers with ids/ips solution using raspberry pi, in: 2017 18th International Carpathian Control Conference (ICCC), IEEE, 2017, pp. 316–320.

[123] X. Wang, A. Kordas, L. Hu, M. Gaedke, D. Smith, Administrative evaluation of intrusion detection system, in: Proceedings of the 2nd annual conference on Research in information technology, 2013, pp. 47–52.

[124] M. Ford, C. Mallery, F. Palmasani, M. Rabb, R. Turner, L. Soles, D. Snider, A process to transfer fail2ban data to an adaptive enterprise intrusion detection and prevention system, in: SoutheastCon 2016, IEEE, 2016, pp. 1–4.

[125] S. Shan, E. Wenger, B. Wang, B. Li, H. Zheng, B. Y. Zhao, Gotta catch'em all: Using honeypots to catch adversarial attacks on neural networks, in: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, 2020, pp. 67–83.

[126] CVE, Common vulnerabilities and exposures, https://cve.mitre.org/ [Accessed: February 09, 2021] (2020).

[127] R.-L. J. Jesús, P.-V. O. Cristhian, R.-G. M. René, F.-M. Heberto, How to improve the iot security implementing ids/ips tool using raspberry pi 3b, Editorial Preface From the Desk of Managing Editor... 10 (9) (2019).

[128] NIST, Common vulnerability scoring system calculator, https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator [Accessed: February 09, 2021] (2020).

[129] sqlmap, sqlmap - automatic sql injection and database takeover tool, http://sqlmap.org/ [Accessed: February 20, 2021] (2020).

[130] sudo, Buffer overflow when pwfeedback is set in sudoers, https://www.sudo.ws/alerts/pwfeedback.html [Accessed: February 20, 2021] (2020).

[131] A. Uprety, D. B. Rawat, Reinforcement learning for iot security: A comprehensive survey, IEEE Internet Things J. 8 (11) (2021) 8693–8706. doi:10.1109/JIOT.2020.3040957.
URL https://doi.org/10.1109/JIOT.2020.3040957

[132] X. Yao, F. Farha, R. Li, I. Psychoula, L. Chen, H. Ning, Security and privacy issues of physical objects in the iot: Challenges and opportunities, Digital Communications and Networks 7 (3) (2021) 373–384.

[133] Z. Wu, S. Shen, H. Zhou, H. Li, C. Lu, D. Zou, An effective approach for the protection of user commodity viewing privacy in e-commerce website, Knowledge-Based Systems 220 (2021) 106952.

[134] Z. Wu, G. Li, S. Shen, X. Lian, E. Chen, G. Xu, Constructing dummy query sequences to protect location privacy and query privacy in location-based services, World Wide Web 24 (1) (2021) 25–49. doi:10.1007/s11280-020-00830-x.
URL https://doi.org/10.1007/s11280-020-00830-x

[135] Z. Wu, S. Shen, X. Lian, X. Su, E. Chen, A dummy-based user privacy protection approach for text information retrieval, Knowl. Based Syst. 195 (2020) 105679. doi:10.1016/j.knosys.2020.105679.
URL https://doi.org/10.1016/j.knosys.2020.105679

[136] B. Wu, X. Chen, Z. Wu, Z. Zhao, Z. Mei, C. Zhang, Privacy-guarding optimal route finding with support for semantic search on encrypted graph in cloud computing scenario, Wirel. Commun. Mob. Comput. 2021 (2021) 6617959:1–6617959:12. doi:10.1155/2021/6617959.
URL https://doi.org/10.1155/2021/6617959