IoT-PRIDS: leveraging packet representations for intrusion detection in IoT networks

Zohourian, Alireza; Dadkhah, Sajjad; Molyneaux, Heather; Neto, Euclides Carlos Pinto; Ghorbani, Ali A.

Publisher's version / Version de l'éditeur:
https://doi.org/10.1016/j.cose.2024.104034
Computers & Security, 146, C, 2024-08-05

Computers & Security 146 (2024) 104034

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

IoT-PRIDS: Leveraging packet representations for intrusion detection in IoT networks

Alireza Zohourian [a,*], Sajjad Dadkhah [a], Heather Molyneaux [b], Euclides Carlos Pinto Neto [a], Ali A. Ghorbani [a]

[a] Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada
[b] National Research Council Canada, Fredericton, NB, Canada

ARTICLE INFO

Keywords:
Internet of Things (IoT)
IoT security
IoT profiling
Intrusion detection
Intrusion Detection System (IDS)

ABSTRACT

The Internet of Things (IoT) devices have been integrated into almost all everyday applications of human life such as healthcare, transportation and agriculture. This widespread adoption of IoT has opened a large threat landscape to computer networks, leaving security gaps in IoT-enabled networks. These resource-constrained devices lack sufficient security mechanisms and become the weakest link in our in computer networks and jeopardize systems and data. To address this issue, Intrusion Detection Systems (IDS) have been proposed as one of many tools to mitigate IoT related intrusions. While IDS have proven to be a crucial tools for threat detection, their dependence on labeled data and their high computational costs have become obstacles to real life adoption. In this work, we present IoT-PRIDS, a new framework equipped with a host-based anomaly-based intrusion detection system that leverages "packet representations" to understand the typical behavior of devices, focusing on their communications, services, and packet header values. It is a lightweight non-ML model that relies solely on benign network traffic for intrusion detection and offers a practical way for securing IoT environments. Our results show that this model can detect the majority of abnormal flows while keeping false alarms at a minimum and is promising to be used in real-world applications.

1. Introduction

The internet of things has been experiencing an exponential growth in terms of devices, users and applications and has become a ubiquitous element of human life existing in various industries and use cases (Sarker et al., 2023). It is clearly visible that this technology has been widely adopted and integrated in various sectors each of which leverage certain capabilities for specific purposes (Thabit et al., 2023). For instance, in the healthcare domain, various medical IoT devices such as wearables and remote monitors have been extensively used to enhance patient monitoring and help doctors and practitioners in wellness management (Neto et al., 2023b). Similarly, is a smart city scenario, several IoT-enabled sensors are deployed in the streets to gather data about traffic conditions and environmental factors to improve traffic management (Alahi et al., 2023). These are a few examples that highlight the wide usage and diversity of IoT's impact.

IoT devices have become an inevitable component in our life and seamlessly connect various aspects of our physical world to the digital world. However, this widespread adoption has brought about several concerns about their vulnerabilities and threat landscape (Schiller et al., 2022). These vulnerabilities have emerged due to several factors such as lack of resources and standardization and therefore, lack of standardized security measures. More importantly, the heterogeneous nature of these devices has made this environment even more complex, as there are no standardized technology or protocol for IoT applications. Consequently, this ever-expanding threat landscape has opened new doors for adversaries and enlarged the attack surface. This surface, in turn, has introduced various attack vectors ranging from unauthorized access to disruptions of critical infrastructures (Rizvi et al., 2020). As a result, the complex nature of IoT environment, along with the lack of security and the large threat surface, creates several opportunities for bad actors to find new ways to exploit the vulnerabilities in IoT. This highlights the urgent need for strong security solutions and mechanisms to mitigate the ensuing risks of widespread IoT adoption and integration.

In the domain of IoT security, several cybersecurity solutions have been proposed to mitigate the risks and challenges discussed earlier. In this sense, the inherent deficiencies of the resource-constrained IoT devices require a proactive approach to keep these networks safe and secure (Ge et al., 2021). Among the many cybersecurity solutions, Intrusion Detection Systems (IDS) have emerged as a promising solution

* Corresponding author.
E-mail address: alireza.zohourian@unb.ca (A. Zohourian).

https://doi.org/10.1016/j.cose.2024.104034
Received 8 May 2024; Received in revised form 19 June 2024; Accepted 1 August 2024
Available online 5 August 2024

to fill the security gaps of IoT devices (Mishra and Pandya, 2021). An IDS serves as a surveillance component that constantly monitors the activities inside an IoT network. By identifying unusual patterns or abnormal behavior of IoT devices, IDS are capable of detecting potential attacks and intrusions. The significance of intrusion detection systems must not be underestimated as they elevate the security posture of IoT networks by identifying anomalies and notifying network admins about suspicious behavior in a timely manner.

In the dynamic landscape of IoT security, various types of intrusion detection systems (IDS) have emerged, each employing distinct approaches to safeguard networks and devices. Signature-based IDS, one of the conventional methods, relies on predefined patterns and known attack signatures to identify malicious activities (Khraisat and Alazab, 2021). Anomaly-based IDS, on the other hand, focuses on detecting deviations from established baselines, making it adept at spotting novel attacks (Alsoufi et al., 2021). Moreover, AI-powered IDS leverages algorithms to learn and adapt to evolving threat scenarios, enhancing its efficacy over time (DeMedeiros et al., 2023). These diverse approaches collectively contribute to a robust intrusion detection framework, fortifying IoT networks against a spectrum of potential threats.

In the domain of intrusion detection for IoT networks, several types of IDS have been proposed, each of which use a unique methodology to identify the anomalies and intrusions. On the one hand, signature-based IDS create signatures for known attacks and upon matching of a signature, an alert is generated to the system (Khraisat and Alazab, 2021). On the other hand, anomaly-based IDS establish a baseline from the normal behavior of the devices and use it to detect abnormal activities that can be potential attacks (Alsoufi et al., 2021). These methods are well-suited for detection of unknown attacks. Moreover, AI-based IDS leverage machine learning algorithms to find patterns in the network traffic and detect malicious behavior (DeMedeiros et al., 2023). All these approaches may work hand in hand to create a robust intrusion detection framework capable of detecting both known and unknown attacks.

While IDS are important tools for preventing malicious attacks, there are several remaining challenges in threat detection. Numerous intrusion detection techniques rely on labeled data, necessitating both normal and malicious traffic for training purposes. Additionally, many machine learning-based approaches often entail high computational costs. Hussain et al. (2020) In response to these challenges, we propose a lightweight non-ML method that operates efficiently, exclusively leveraging normal network traffic to discern legitimate patterns from abnormal ones. This approach streamlines the learning process and minimizes computational overhead, providing an effective solution for intrusion detection with a reduced resource burden.

While intrusion detection systems are necessary tools for detection of threats, they still face several challenges that need to be solved. Firstly, many IDS systems rely on labeled data and require both benign and attack samples to learn the normal and malicious behaviors. Secondly, several machine learning-based IDS require huge computational resources that prevents them from becoming a real-world applicable solution (Hussain et al., 2020). To address these challenges, we propose a lightweight non-ML model that only leverages the benign network traffic data to create a baseline profile of the devices behavior to efficiently detect and identify malicious packets with a minimal computational overhead.

IoT devices are often exploited as entry points to either relay in the network or conduct other cyberattacks such as botnet attacks (Kumari and Jain, 2023). Since IoT devices have simple network traffic compared to that of non-IoT devices, this research focuses on profiling of IoT devices in a host-based manner. This study aims to demonstrate that one can use packet-level profiling to achieve great performance for detection of various attacks in IoT environments. By focusing on important features of packets, we intend to identify malicious flows of packets and detect potential intrusions.

In this research, we propose IoT-PRIDS (Packet Representation IDS for IoT), a new profiling method that uses header fields of a packet to construct a "representation" that encompasses a large set of similar packets. We use this representation to narrow down the diversity of packets by choosing specific header field values that can represent a set of packets meaningfully. By storing representations of normal packets, any new packet representation that is not similar enough to the normal packets will be tagged as malicious and this will help detect intrusions.

The main contributions of this article are as follows:

- A new profiling technique (IoT-PRIDS) that leverages packet representations to detect abnormal packets and flows and works as an intrusion detection system for IoT networks and devices.
- A light-weight non-ML mechanism (IoT-PRIDS) for IoT intrusion detection that works as an anomaly-based IDS and can be deployed as both network and host intrusion detection systems.
- Comprehensive evaluation of IoT-PRIDS from both performance and efficiency point of view using two well-known IoT datasets, namely CICIoT2022 and CICIoT2023.

The paper is organized as follows: Section 2 introduces the topic of study, outlines its significance, and provides relevant background information. In Section 3, the methodology for creating packet representations and quantifying malicious packets is discussed. Section 5 provides detailed explanations of experiments and results. In Sections 6 and 7, limitations, future work and conclusions are presented.

2. Background

In this section, we present an overview of intrusion detection systems, their types and characteristics. Firstly, we analyze IDS from a deployment point of view and discuss Host-based IDS (HIDS) and Network-based IDS (NIDS). Secondly, we analyze IDS from a methodology perspective and categorize them into Signature-based IDS (SIDS) and Anomaly-based IDS (AIDS). Finally, we delve into different anomaly-based IDS with a focus on different types of Machine Learning-based IDS and its different approaches. Fig. 1 shows an overview of the reviewed intrusion detection systems.

2.1. Intrusion Detection Systems (IDS)

An Intrusion Detection System (IDS) is a security technology that monitors network or system activities to identify and respond to unauthorized or malicious activities (Albulayhi et al., 2021). IDSs play a significant role in maintaining the security of computer systems and networks by detecting and alerting administrators about potential security breaches or attacks.

2.2. Host-based vs. Network-based

Host-based intrusion detection methods focus on monitoring activities of single endpoints or hosts inside the network. These IDS need to install software agents on each host to analyze system and process logs, file manipulation and integrity, and application behavior for indications of unauthorized access or abnormal activity (Satilmiş et al., 2024).

Baz (2022) introduces SEHIDS, a Self Evolving Host-based Intrusion Detection System that weaponizes each IoT host with a basic artificial neural network structure and a process to continuously train and adapt it online. When node performance declines, the ANN evolves accordingly, enabling each node to detect specific threats it faces. This adaptability is necessary for SEHIDS to address diverse and dynamic IoT traffic patterns.

In contrast, network-based intrusion detection operates at the network level by monitoring the traffic the goes through network devices such as routers, switches, and firewalls. NIDS scans packets inside the network in real-time and look for patterns that indicate known attack

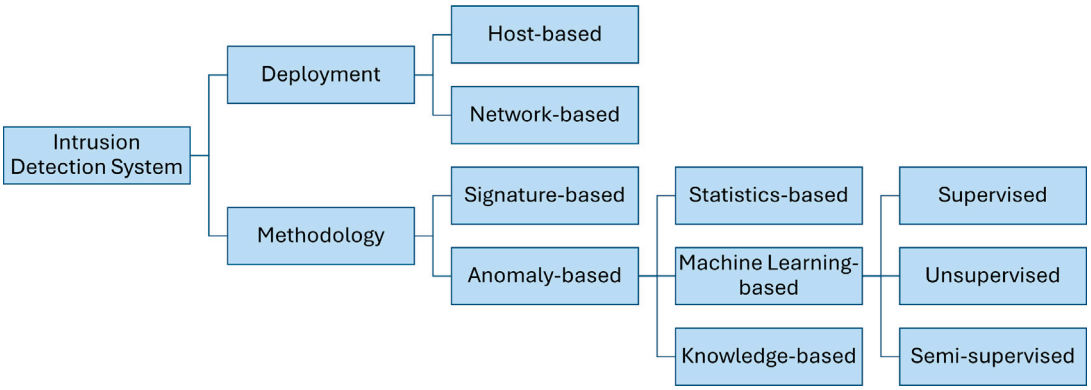


Fig. 1. Different types of Intrusion Detection Systems in the literature based on deployment and approach.

signatures or abnormal behavior that could be an intrusion (Gyamfi and Jurcut, 2022).

Fraihat et al. (2023) suggest an NIDS utilizing Machine Learning (ML) enhanced by a modified Arithmetic Optimization Algorithm (AOA) that identifies the best optimal set of features to utilize for training various machine learning models such as Random Forest and Extra Trees. This work succeeded in acquiring up to 99% accuracy for binary classification and 98% accuracy for multi-class classification while decreasing the number of utilized attributes by up to 84%.

A Hybrid IDS leverages both HIDS and NIDS capabilities to create a more holistic approach for monitoring of systems and networks. Hybrid IDS combine the advantages of both HIDS and NIDS and use the device-specific information from the HIDS and the network-level visibility of NIDS to correlate abnormal behavior and unravel malicious behavior patterns (Martins et al., 2022).

Harsha (2024) propose NHIDS-Net, an innovative network-host intrusion detection system utilizing advanced deep learning methods applied to the NSLKDD dataset. The architecture is designed to improve feature extraction from both network and host data, offering a comprehensive approach to intrusion detection. The system is capable of adjusting and generalizing to new attack scenarios making it effective in reducing false positives.

2.3. Signature-based vs. Anomaly-based

From a methodology viewpoint, there are two major types of IDS: Signature-based and Anomaly-based.

A Signature-based IDS (SIDS) creates a database of signatures for known attacks and uses these signatures against the incoming traffic and if there is a signature match, it will generate an alert. Heidari and Jabraeil Jamali (2023).

Li et al. (2019) propose CBSigIDS, a holistic framework for signature-based intrusion detection systems that are collaboratively blockchained and aims to gradually construct and refresh a reliable database of signatures within a collaborative IoT setting. CBSigIDS offers a verifiable approach in decentralized architectures, eliminating the necessity for a trusted intermediary.

On the contrary, Anomaly-based IDS (AIDS) generate a baseline profile of normal behavior for network traffic and then alert any deviation from the created baseline profile as a potential threat Adnan et al. (2021). We will discuss different types of AIDS in Section 2.4.

Bacha et al. (2024) suggest an anomaly-based IDS that utilizes kernel PCA to decrease the dimensionality of data attributes and employs the kernel ELM for binary classification to discern the state of the network traffic in terms of being benign or malicious or, for multiclass classification, to classify groups of intrusions into their certain types.

In a Hybrid IDS, both SIDS and AIDS are integrated to create a more comprehensive approach to intrusion detection to mitigate the shortcomings of each approach. In a Hybrid IDS setup, the SIDS scans the network for known attacks using signatures, while a binary AIDS tries to scan for deviations from normal to detect intrusions and a multi-class AIDS will classify the type of the intrusions (Saif et al., 2022). This configuration is illustrated in Fig. 2.

Otoum and Nayak (2021) propose the AS-IDS model, which integrates both known and unknown attack detection methods in IoT environments. The framework consists of three phases: filtering the network traffic, preprocessing the input data, and the hybrid intrusion detection component. In the traffic filtering stage, incoming packets are filtered inside the IoT gateway based on their attributes. The next step involves encoding, normalizing, and removing redundancy. The final hybrid IDS phase combines signature and anomaly detection. Signature-based detection is performed using Lightweight Neural Network (LightNet), while anomaly-based detection utilizes Deep Q-learning to recognize formerly unknown intrusions.

2.4. Statistics-based vs. Knowledge-based vs. Machine learning-based

Anomaly-based IDS (AIDS) are categorized into three classes: Statistics-based vs. Knowledge-based vs. Machine Learning-based.

A statistical-based IDS collects statistical features from the network traffic packets and flows and builds a statistical model that represents the benign traffic and normal interactions. When there is a distinction between two extracted statistical patterns, it will be considered as an intrusion (Spadaccino and Cuomo, 2020).

A Knowledge-based IDS is a type of expert system that utilizes a knowledge base that contains the signature of benign and normal traffic. This is different from a signature-based approach that extracts signatures from attack traffic. Regardless, any activity different from this knowledge base is considered anomalous (Adnan et al., 2021).

A machine learning-based IDS is one that utilizes machine learning algorithms to detect and identify suspicious activities or possible intrusions. This type of IDS learns the underlying patterns and characteristics from network traffic to identify anomalies or attacks (Thakkar and Lohiya, 2021). ML-based models are believed to be a promising solution, capable of detecting unseen attacks. Different types of machine learning IDS will be identified in Section 2.5.

2.5. Supervised vs. Unsupervised vs. Semi-supervised

Machine learning-based intrusion detection techniques can be categorized into three main approaches: supervised, unsupervised and semi-supervised learning (Faraj et al., 2020).

In supervised intrusion detection, the system relies on labeled data to learn both normal and malicious behavior and then detects the anomalies in a binary classification to *detect* the attack, or in a multi-class classification, to *classify* the attack type (Abdulla and Jameel, 2023).

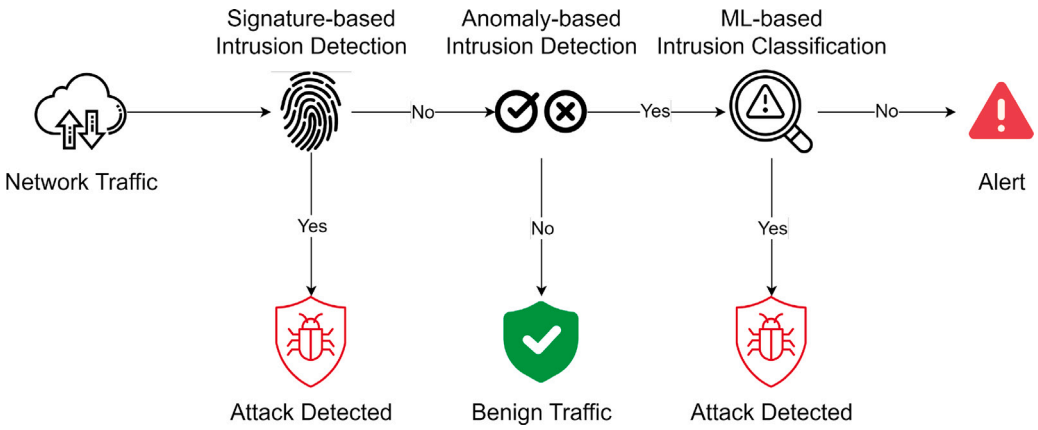


Fig. 2. Overview of a hybrid intrusion detection system.

Roy et al. (2022) present a new methodology utilizing ML to identify cyber-incidents and abnormalities in IoT environments. Using a variety of optimization techniques such as removing multicollinearity, sampling, and reducing the feature set dimensions, their method can pinpoint the crucial set of attributes for intrusion detection taking significantly less data and time for training.

Unlike supervised approaches, unsupervised intrusion detection does not rely on labeled data. Instead, it uses algorithms to discover patterns in the data and report as anomalies any instances that significantly differ from these learned patterns (Idrissi et al., 2020).

Bhatia et al. (2019) introduce a network-centric, behavior-learning based anomaly detection model for enhancing the security of IoT networks by leveraging the simplicity of TCP packets from IoT endpoints to identify several kinds of DDoS attacks in real-time, using unsupervised methods that only use a small number of attributes.

Semi-supervised IDS combines both supervised and unsupervised approaches. It utilizes a small amount of labeled data, typically including examples of known attacks or anomalies, together with a larger number of unlabeled records. During training, the IDS learns from the labeled data to classify known types of attacks, while also identifying patterns and anomalies in the unlabeled data. This hybrid approach allows the IDS to benefit from the accuracy of supervised learning while also being able to detect novel threats through unsupervised techniques (Chaabouni et al., 2019).

Abdel-Basset et al. (2021) introduce SS-Deep-ID, a semi-supervised deep learning method for threat detection that incorporates a multi-scale residual temporal convolutional (MS-Res) component to refine the network's ability to learn spatiotemporal representations. An enhanced traffic attention (TA) mechanism is integrated to gauge the significance metric, aiding the model in focusing on critical aspects of data while in the learning stage. Additionally, they propose a hierarchical semi-supervised approach that considers the sequential nature of IoT network packets during the training phase.

Discussion. Supervised and unsupervised anomaly-based intrusion detection methods have different advantages and disadvantages. Supervised methods rely on labeled data and perform well in detecting known attacks. However, they struggle in detecting unknown attacks, since their knowledge is limited to the labeled data they were trained on. In contrast, unsupervised methods do not require labeled data and this makes them capable of identifying unknown attacks. However, this freedom from labels can result in an increased risk of false positives since they decide only based on data patterns.

In this study, we employ an anomaly-based approach that solely relies on normal traffic and is capable of detecting unknown attacks. This approach is different from supervised methods as it does not rely on labeled attack data to learn both normal and attack behavior and is capable of detecting any kind of abnormal behavior. Moreover, it is different from unsupervised methods as it requires the input data to be labeled as normal, rather than having a mix of normal and abnormal data as input and separating them based on a distance. It is worth emphasizing that this is a novel approach that does not involve any machine learning or clustering methods. However, it is a light-weight system capable of detecting abnormal behavior by only relying on normal network traffic.

Research Questions. In this work, we seek to answer the following research questions that will guide the design, implementation and evaluation of the proposed work:

1. **RQ1:** To what extent is IoT-PRIDS accurate and precise for detection of anomalies and intrusions?
2. **RQ2:** Is IoT-PRIDS light-weight enough to be considered as a practical solution?

3. Methodology

As previously mentioned, our methodology involves the mapping of each packet into a distinct representation, effectively capturing its critical attributes. In this section, we delve into the details of this mapping process, elucidating how it is employed to identify abnormal packets. Furthermore, we define a distance between these representations. This distance metric becomes pivotal in quantifying the degree of maliciousness associated with a packet.

3.1. Packet representation

The packet mapping process relies on three primary criteria: communication, service, and header fields. The underlying principle of this mapping is to eliminate the inherent randomness found in packets and summarize the multitude of similar packets, which essentially perform identical functions, into a single packet representation. This mapping enables us to summarize a large volume of packets into just a handful of distinct packet representations. These few representations serve as a baseline profile that characterize the typical behavior of device packets, allowing us to capture the essential features of normal packet behavior effectively while reducing complexity.

3.1.1. Communication

IoT devices, due to their limited communication capabilities and resource constraints, typically engage in communication with a restricted number of specific endpoints, each serving a particular purpose. These devices feature straightforward configurations, resulting in minimal communication variety. To systematically categorize the communication patterns of IoT devices, we employ a set of subcriteria:

A **Endpoints**: the communication endpoint; IP address and MAC address.
B **Direction**: the direction of the packet; inbound or outbound.

C **Scope**: whether it is an internal communication (LAN) or an external one (WAN).
D **Distribution**: whether it is a unicast, multicast or a broadcast message.

These subcriteria characterize the communication of a packet for a device. If a device initiates a communication with a new endpoint, whether on LAN or on WAN or makes a communication with a known endpoint in an abnormal fashion, it must be flagged as an anomaly.

3.1.2. Service

IoT devices employ particular protocols and port numbers for their necessary services when communicating with an endpoint. Hence, we utilize the following subcriteria to differentiate between various services:

A **Protocol**: the protocol(s) being used for the service; UDP, TCP, etc.
B **Port Number**: the designated port number being used for the service.
C **Service**: the service that is being used for a communication; HTTP, DNS, etc.

We characterize the service being used for a communication using these criteria. If a device uses an unknown service or a known service with an endpoint that was not typical, it must then be flagged as an anomaly.

3.1.3. Header values

One of our key observations pertains to the tendency of IoT devices to employ specific packet header field values during communication for a given service. Consequently, we utilize the following subcriteria to differentiate between packet representations:

A **L2 Header**: Specific header field values for the Layer 2 protocols, e.g. ARP.
B **L3 Header**: Specific header field values for the Layer 3 protocols, e.g. IP, ICMP and IGMP.
C **L4 Header**: Specific header field values for the Layer 4 protocols, e.g. UDP and TCP.
D **L5 Header**: Specific header field values for the Layer 5 protocols, e.g. HTTP, DNS, etc.

Using these subcriteria, the header values of a packet for a service in a communication is determined. If a new packet shows an unknown trait in packet header values, it must be flagged as abnormal.

3.2. Packet mapping

In this section, we formally describe the mapping from packets to representations.

Let $\mathcal{P}$ be the set of all packets and $p \in \mathcal{P}$. We define p as an array of header values and payload:

$$p = \langle h_1, h_2, h_3, \dots, h_n, d \rangle$$

where $h_i, d \in \mathbb{Z}^+$.

Define $\mathcal{R}$ the set of packet representations and $r \in \mathcal{R}$ to be the representation of p:

$$r = \langle h_1^*, h_2^*, h_3^*, \dots, h_m^* \rangle$$

in which each h_i is either eliminated, kept as is, transformed to another value or synthesized, and the payload d is eliminated. Section 4.3 explains the headers used to build the representations in this work. It is worth noting that $m \leq n$ as we would like to eliminate the complexity and randomness of packets and summarize them into a representation. Therefore, it will be important to have $m \ll n$.

Finally, define the many-to-one mapping $f : \mathcal{P} \to \mathcal{R}$ such that $f(p) = r$ and

$$\forall p \in \mathcal{P}, \exists r \in \mathcal{R}, f(p) = r.$$

Now, let $\mathcal{P}_D$ be the set of normal packets of device D. Then, all the packets in $\mathcal{P}_D$ are mapped to their corresponding representations, producing the normal profile $\mathcal{R}_D$ the set of normal representations of device D. It is evident that the ultimate count of packet representations is significantly less than the overall number of packets, as we would like to try to keep the header values as few as possible. To put it differently, numerous packets are mapped to a unique representation that encapsulates all of those packets for the device. From an implementation stand point, the entire mapping can be thought of as a series of deletion, insertion and conversion processes that maps a packet to its corresponding representation. Fig. 3 provides a schematic overview illustrating the entire processes followed in this work. As it is illustrated, this decrease in the number of features and therefore the representations is the main reason for the model to be light-weight.

For a new packet p from device D, p is mapped to its representation r, i.e. $r = f(p)$. If $r \in \mathcal{R}_D$, the packet is considered normal. Otherwise, it may be flagged as abnormal, based on the representation distance defined in the upcoming Section 3.3.

3.3. Representation distance

As previously mentioned, a packet may deviate from the norm for three primary reasons: communication, service, and header values. For instance, anomalies can arise from communicating with an unfamiliar endpoint, utilizing an unseen service, or employing a different header configuration. What further exacerbates the situation is when these factors combine. For example, when a device communicates with an unfamiliar endpoint, employs an unobserved service, and exhibits a new header structure simultaneously.

Furthermore, it is important to note that deviating from any of these criteria does not automatically imply abnormality; it might simply be a previously unseen but perfectly normal packet within the scope of normal traffic. Therefore, only checking if a packet representation belongs to the normal profile is not sufficient to conclude that it is abnormal and it will introduce a huge false positive rate. To address these complexities and reduce false positives, we establish a distance metric between two representations, quantifying the degree of abnormality in cases where a packet does not match any of the established normal representations. This distance metric serves as a measure of abnormality, enabling us to effectively differentiate true abnormal packets (True Positives) from instances that might otherwise be incorrectly labeled as false (False Positives).

We use the Hamming distance to count the number of non-matching positions in two representations. Suppose $u, v \in \mathcal{R}$ are two packet representations. We define

$$d(u, v) = \sum_{i=1}^{n} (u_i \neq v_i)$$

Now, we define the distance of a packet from a representation set to be the minimum distance from all the representations. Let $\mathcal{R}_D$ be the set of normal representations of device D and p be a new packet whose representation is u. Define

$$d(u, \mathcal{R}_D) = \min_{v \in \mathcal{R}_D} (d(u, v)) \tag{1}$$

We then use this distance to make a decision about the packet's abnormality, which helps reduce the number of false positives introduced by the lack of enough packets from a specific device.

3.4. Profiling and monitoring

Our model includes two main phases: Profiling and Monitoring. During the profiling stage, the benign network traffic packets are mapped to their respective representations and archived in a database of device profiles. During the monitoring stage, every incoming packet is mapped to its corresponding representation, and its distance from

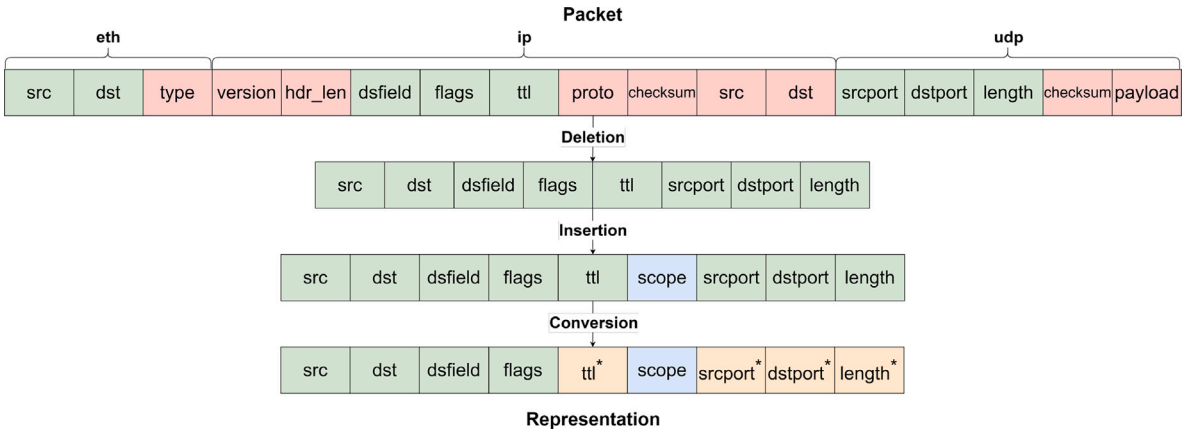


Fig. 3. Mapping a UDP packet to its corresponding representation.

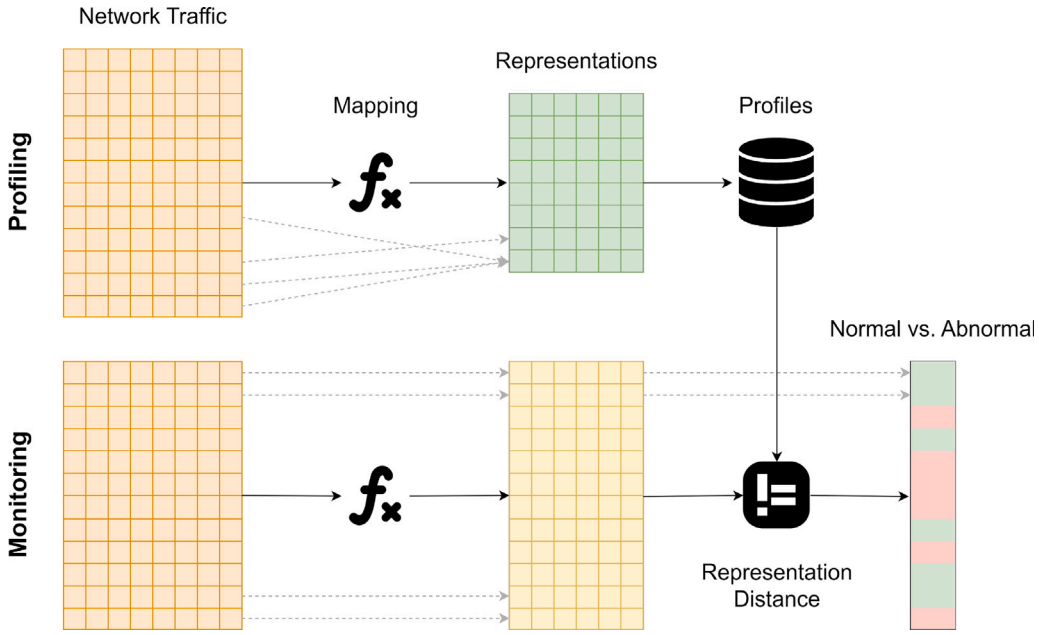


Fig. 4. Profiling and monitoring overview.

the representation set (device profile) is computed. Should the distance exceed a predefined threshold, the packet is identified as abnormal. (see Fig. 4)

As a final mitigation for reducing the number of false positives, we also performed a flow-based intrusion detection using the packet representations by taking the mode of the packets belonging to each flow. This is specifically helpful as it considers all the packets that belong to the same flow as a whole and decides the abnormality of a flow based on a voting among single packets. Also, from an intrusion detection point of view, it is not desirable to get one alert per packet, as this can be extremely overwhelming for the end user. The results in Section 5 shows a significance improvement resulting from the employment of flow-based intrusion detection.

3.5. Benign vs. Attack

Benign network traffic refers to legitimate and non-malicious data exchanges within a network. It encompasses the routine and authorized communication between devices, where data packets are generated and transmitted in accordance with standard protocols and established communication patterns.

In contrast to Benign network traffic, Attack network traffic refers to malicious and unauthorized data exchanges that occur within a network. It involves actions aimed at compromising the integrity, confidentiality, or availability of the network or its connected devices.

From the perspective of network packet capture (pcap), the distinction between a benign pcap and an attack one lies in the inclusion of both legitimate and illegitimate packets in attack traffic. Essentially, while many datasets categorize a pcap as either benign or attack in its entirety, an attack pcap is essentially a combination of both benign and malicious packets. Consequently, there is an ensuing discourse on how to differentiate between a benign pcap and an attack one. Of greater significance is the concern regarding the labeling of packets and/or flows in a given network traffic to determine their malicious nature.

Accordingly, this study focuses on the precise identification of malicious packets within network traffic, irrespective of whether the traffic, in its entirety, has been labeled as benign or attack. In essence, the authors aim to meticulously identify every abnormal and possibly malicious packet in the network traffic.

4. Data preprocessing

4.1. Datasets

In this study, we conduct experiments using the CICIoT2022 (Dadkhah et al., 2022) and CICIoT2023 (Neto et al., 2023a) datasets. We incorporate power, interactions, active and idle traffic from the former, and we concentrate on the normal traffic from the latter to establish a baseline profile of typical behavior. All the normal packets from these datasets were mapped to representations and stored as profiles of devices and the network.

4.2. Data acquisition

The CICIoT2022 dataset serves as a profiling dataset containing packet captures of network traffic from IoT devices, with the primary purpose of facilitating the behavioral analysis of these devices. The dataset encompasses diverse types of traffic, including power-related, interactive, idle, active, and scenario-based traffic, providing a comprehensive range for establishing a baseline profile of device behavior. Since it was collected within the same network as CICIoT2023, we leverage the normal traffic from CICIoT2022 to construct the baseline profile.

The CICIoT2023 dataset is designed as an attack dataset, featuring packet captures of network traffic from IoT devices. Its primary purpose is to support security analysis and intrusion detection efforts. The dataset encompasses both benign and attack traffic, showcasing a wide array of attacks ranging from reconnaissance and spoofing to DDoS and Web-based attacks. In our experiments, we utilize the benign traffic for the profiling phase to establish the normal behavior of IoT devices. Subsequently, the attack traffic from CICIoT2023 is employed for the monitoring phase and intrusion detection purposes.

4.3. Packet header engineering

We followed a rigorous process in selecting appropriate header values for the mapping of packets to their representations. We start with a packet including all its header values from ETH, IP, UDP and TCP headers and remove inappropriate and/or ineffective headers gradually. The subsequent sections provide a comprehensive explanation of each step in detail:

4.3.1. Remove random headers

Certain header values in packets exhibit a random nature that may not inherently reveal information about the device or its behavior. Firstly, most headers relating to **length** and **size** are contingent on the network status and data being transferred at a specific time, often appearing random from an application standpoint. Secondly, **sequence numbers** and **offsets** are generated randomly and are contingent on the number of segments or fragments. Lastly, **checksums**, employed for data integrity, do not indicate any specific device behavior. Considering these observations, the following headers – which follow the naming conventions from Wireshark – will be excluded from a representation: ip-total-len, identification, fragment-offset, ip-checksum, udp-checksum, seq-num, ack-num, tcp-window, tcp-checksum, urgent-pointer.

4.3.2. Remove unchanged headers

In our pursuit of identifying distinguishing features in packets, we exclude certain headers that exhibit constancy either due to the straightforward characteristics of IoT devices or the protocols under consideration for our analysis. Given our exclusive focus on TCP and UDP packets, headers such as **type** in the ETH layer and **version** and **protocol** in the IP layer become inconsequential. For the remaining headers that did not exhibit evident variation and raised suspicions of constancy, a comprehensive analysis of the entire dataset packets was conducted to confirm that they do not serve as differentiating factors. Following this assessment, the following headers were removed: type, version, ip-hdr-len, diff-ser-field, protocol.

4.3.3. Convert headers with diverse values

Port numbers play a significant role in identifying the normal traits of devices, such as what services they normally use. However, in each UDP/TCP communication, the client side chooses an ephemeral port number for the corresponding socket. First, keeping all the ephemeral port numbers will result in multifariousness of the profiles' representations, lack of generalizability and poor time performance. Second, this ephemeral port is sometimes chosen in a specific range due to the underlying operating system. Therefore, after identifying the service port, the ephemeral port is simply ranges of 10,000 ports, i.e. [0–10,000], [10,001–20,000], . . . , [60,001–65,535]. Moreover, many devices show repetitive payload lengths that can be used as a distinguishing feature. However, keeping all the payload lengths will result in huge diversity of profile representations. As a result, udp and tcp payload lengths were mapped to the nearest higher power of 2.

4.3.4. Remove unreliable headers

Within the remaining headers, both source and destination IP are considered unreliable, given their potential to change over time. Consequently, these headers are excluded from the representation. Nonetheless, they prove valuable in the synthesis of a novel feature, "scope" which indicates whether the communication is internal (occurs within the local network) or external (extends to the global internet). Therefore, this new feature titled scope is added to the representations.

4.3.5. Headers for communication purposes

As mentioned earlier, one important component of a packet is the communication. In order to keep the required communication characteristics, namely endpoints, direction, scope and distribution, it is sufficient to only keep the source and destination mac addresses along with the "scope" feature engineered earlier. Keeping these two mac addresses will preserve the endpoints, the direction of the communication, i.e. source to destination, the distribution of the message, i.e. unicast, multicast and broadcast as there are reserved ranges of mac addresses for multicast and broadcast communications.

Discussion. The fundamental idea behind these representations is to summarize the key features of packets into concise representations, thereby reducing the multitude of packets from several million to just a few thousand representations. Essentially, each representation encapsulates a substantial number of packets that share similar characteristics and nature. This condensation is achievable by eliminating the packet headers mentioned earlier. A discerning reader might recognize that adjusting the number of features necessitates finding a trade-off between accuracy and efficiency. Retaining more features results in a greater number of representations, leading to reduced efficiency. Conversely, keeping fewer features yields fewer representations, thereby sacrificing some accuracy and introducing some false positives.

5. Evaluation

In this section, we detail the procedures involved in profiling and monitoring. During the profiling phase, we capture all distinctive packet representations from the normal traffic to establish the baseline profile. Subsequently, in the monitoring phase, incoming packets are categorized as either normal or abnormal. The rationale behind this methodology is to extract and retain critical information from normal packets, utilizing them as a benchmark to identify packets that significantly deviate from the established norm.

Table 1
Total number of packets and representations.

	Duration	Packets	Representations
CICIoT2022	~ 545 h	88,810,001	113,680
CICIoT2023	~ 28 h	11,102,704	22,592
Total	~ 572 h	99,912,705	130,715

5.1. Profiling

In the profiling phase, we associate every packet originating from the normal network traffic of CICIoT2022 and CICIoT2023 with its corresponding representation, storing all the distinct representations to establish the baseline profile. Table 1 provides statistics regarding the total count of packets and representations. The mapping of packets to their representations is evidently advantageous, as it condenses the packet space, resulting in a more lightweight and efficient method. It is obvious that by adding more features and making the feature conversions more complex - e.g. keeping all the exact port numbers or packet lengths — the number of representations will increase and conversely, if one removes some features and makes the feature conversions less complex, this number will drop. The representation set is a search space which the model will use to find representation distances. This will impact the efficiency of the model; therefore, there needs to be a balance between the complexity and generalizability of the representations to detect anomalies both efficiently and accurately.

5.2. Labeling

As we are following a packet-based methodology for detecting intrusions, for a new attack .pcap file, all the packets from the attacker to the victim has been labeled as attack, based on our knowledge about the attacker, the victim and the attack type. It is worth noting that in some .pcap files, not all the traffic between the attacker and the victim are "actually" malicious, as the attacker and the victim might have had their normal traffic in advance. For example, the attacker raspberry pi makes communications with google nest mini speaker over port 8009 for casting purposes. Therefore, the labeling has been meticulously performed with a high level of details.

5.3. Monitoring

During the monitoring phase, involving new network traffic such as attack traffic from CICIoT2023, every packet is associated with its respective representation and then compared against the set of normal representations (normal profile) established during the training phase. In our experiments, we captured the profiles on all the normal traffic once from the CICIoT2022 dataset and once from the CICIoT2023 dataset, and tested it on all the attack data of the CICIoT2023 dataset with the following considerations: Firstly, we only tested the models on the attacks that involves tcp or udp packets. Secondly, we tested the model in both packet-based intrusion detection and flow-based intrusion detection. Therefore, for each attack type, we conducted the intrusion detection for four rounds based on the benign data for profiling, CICIoT2022 and CICIoT2023, and based on the type of intrusion detection, packet-based or flow-based. Lastly, we tested the model on a maximum of 10,000 packets of attack data red in order to ensure that there will be no effect on the generalization of the model. In most of the cases the results remained unaffected.

5.4. Performance of IoT-PRIDS (RQ1)

As IoT-PRIDS is a baseline model that stores the representations of normal packets for each device as their profile, it will compare the new incoming traffic with the normal representations of the same device and in case the new packets are so much different than the normal representations, they will be considered abnormal/malicious. In other words, our model leverages the fact that IoT devices have a considerably simple traffic and they tend to send the same traffic for their functions and capabilities. Therefore, any traffic that conflicts this simple behavior will be identified as an anomaly. For instance, if you see SSH traffic to/from a camera that communicates with the cloud over HTTPS, the SSH packets are not similar to the normal packets stored in the camera's profile and therefore, all the SSH traffic will be considered malicious.

Table 2
Performance of the model for detection of Web Attacks using CICIoT2022 and CICIoT2023 datasets for packet-based and flow-based intrusion detection.

Attack	Metrics	Packets		Flows	
		2022	2023	2022	2023
Backdoor Malware	Accuracy	0.9819	1.0000	1.0000	1.0000
	Precision	0.7674	1.0000	1.0000	1.0000
	Recall	0.9429	1.0000	1.0000	1.0000
	F1-score	0.8462	1.0000	1.0000	1.0000
Browser Hijacking	Accuracy	0.9841	0.9815	0.9977	0.9983
	Precision	0.9055	0.4737	0.9851	0.9890
	Recall	1.0000	1.0000	1.0000	1.0000
	F1-score	0.9504	0.6429	0.9925	0.9944
Command Injection	Accuracy	0.9852	0.9900	0.9998	1.0000
	Precision	0.5764	0.7391	0.9899	1.0000
	Recall	0.9235	1.0000	1.0000	1.0000
	F1-score	0.7098	0.8500	0.9949	1.0000
SQL Injection	Accuracy	0.9127	0.9907	1.0000	1.0000
	Precision	0.1251	0.6087	1.0000	1.0000
	Recall	0.9044	1.0000	1.0000	1.0000
	F1-score	0.2198	0.7568	1.0000	1.0000
Uploading Attack	Accuracy	0.9847	0.9917	0.9997	1.0000
	Precision	0.6501	0.7857	0.9889	1.0000
	Recall	0.9291	1.0000	1.0000	1.0000
	F1-score	0.7650	0.8800	0.9944	1.0000
Cross Site Scripting	Accuracy	0.9881	0.9925	0.9998	1.0000
	Precision	0.5238	0.7333	0.9848	1.0000
	Recall	0.9308	1.0000	1.0000	1.0000
	F1-score	0.6704	0.8462	0.9924	1.0000

In this section, we provide extensive details for analyzing the performance of IoT-PRIDS using the well-known metrics accuracy, precision, recall and F1-score. In the following, we detail the procedures and results of the experiments based on different attack types including: Web attacks; Reconnaissance attacks; Dictionary brute force attacks; DoS attacks; and DDoS attacks.

5.4.1. Web attacks

Each attack includes a .pcap file that has been fed to the model and both packet-based and flow-based intrusion detection has been performed. Table 2 shows the results for web attacks. Obviously, using the CICIoT2023 dataset gives a better performance overall as it includes representations for most of the devices and the CICIoT2022 dataset lacks representations for many devices that are in the CICIoT2023 dataset. However, even without the knowledge about many devices, the profiling model with CICIoT2022 performs really well in terms of detecting the attacks with a recall of at least 90%, but with the downside of having a large number of false positives. More importantly, flow-based intrusion detection outperforms the packet-based one because taking the mode of packets for a flow will eliminate all the packets that were detected as normal but belong to a flow whose most packets have been detected as abnormal. Consequently, the recall for all the attacks is 100% for flow-based detection, regardless of which dataset was used as the baseline profile.

Fig. 5 shows the frequency of both benign and attack flows for all the web attacks. It is obvious from the graph that the web attack data is highly imbalanced towards the benign traffic. Regardless, our model is capable of detecting these attacks perfectly. This highlights the fact that

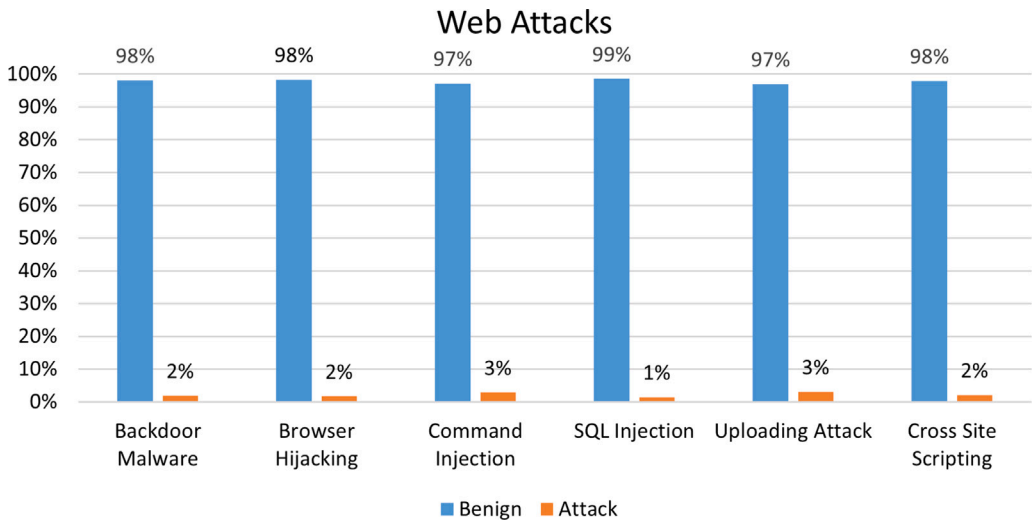


Fig. 5. Percentage of benign and attack for web attacks.

Table 3
Performance of the model for detection of Reconnaissance Attacks using CICIoT2022 and CICIoT2023 datasets for packet-based and flow-based intrusion detection.

Attack types	Metrics	Packets		Flows	
		2022	2023	2022	2023
Host discovery	Accuracy	0.9784	0.9665	0.9690	0.9151
	Precision	0.9719	0.9840	0.9856	0.9866
	Recall	0.9863	0.9500	0.9797	0.9176
	F1-score	0.9791	0.9667	0.9826	0.9508
OS scan	Accuracy	0.9242	0.9550	0.9880	0.9901
	Precision	0.7987	0.8699	0.9862	0.9885
	Recall	1.0000	1.0000	1.0000	1.0000
	F1-score	0.8881	0.9304	0.9931	0.9942
Port scan	Accuracy	0.8885	0.9326	0.9835	0.9865
	Precision	0.7540	0.8351	0.9805	0.9839
	Recall	1.0000	1.0000	1.0000	1.0000
	F1-score	0.8597	0.9102	0.9901	0.9919
Vulnerability scan	Accuracy	0.9332	1.0000	0.9799	1.0000
	Precision	0.8786	1.0000	0.9635	1.0000
	Recall	1.0000	1.0000	1.0000	1.0000
	F1-score	0.9345	1.0000	0.9814	1.0000

Table 4
Performance of the model for detection of Dictionary Attacks using CICIoT2022 and CICIoT2023 datasets for packet-based and flow-based intrusion detection.

Attack types	Metrics	Packets		Flows	
		2022	2023	2022	2023
SSH BF 1	Accuracy	0.9449	0.8228	0.9515	0.9757
	Precision	0.5160	0.2490	0.5000	0.6667
	Recall	1.0000	1.0000	1.0000	1.0000
	F1-score	0.6807	0.3988	0.6667	0.8000
SSH BF 2	Accuracy	0.9399	0.9294	0.9692	0.8092
	Precision	0.5286	0.4883	0.5000	0.1389
	Recall	1.0000	1.0000	1.0000	1.0000
	F1-score	0.6916	0.6562	0.6667	0.2439
SSH BF 3	Accuracy	0.9099	1.0000	0.9656	1.0000
	Precision	0.7657	1.0000	0.7879	1.0000
	Recall	1.0000	1.0000	1.0000	1.0000
	F1-score	0.8673	1.0000	0.8814	1.0000
RTSP BF 4	Accuracy	0.9332	0.9490	0.9799	1.0000
	Precision	0.8786	1.0000	0.9635	1.0000
	Recall	1.0000	0.8945	1.0000	1.0000
	F1-score	0.9345	0.9443	0.9814	1.0000
RTSP BF 5	Accuracy	0.9520	1.0000	0.9873	1.0000
	Precision	0.9240	1.0000	0.9807	1.0000
	Recall	1.0000	1.0000	1.0000	1.0000
	F1-score	0.9605	1.0000	0.9903	1.0000

detecting attacks from a packet level in this manner, does not require the data to be balanced.

5.4.2. Reconnaissance attacks

Each type of reconnaissance attack in the CICIoT2023 dataset has one .pcap file that consists several scanning attempts on different hosts. We tested the model on all these attacks except for the Ping Sweep attack as it is an ARP/ICMP-based attack and does not fall into the scope of this work. Table 3 shows the results on these attacks. Almost the same trend is visible to reconnaissance attacks as in web attacks, CICIoT2023 outperforming the 2022 dataset and the flow-based approach, surpassing the packet-based approach overall. However, we still see a minimum of 91% recall and 80% precision in each attack type, In most cases all the malicious flows have been detected with a very low rate in false positives.

5.4.3. Dictionary brute force

The CICIoT2023 dataset includes a .pcap file for all the dictionary attacks. However, after further investigation into the file, we realized that it is composed of 5 separate dictionary attacks (three SSH Brute Force attacks and two RTSP URL Brute Force attacks) merged into one .pcap file. This was confirmed by sorting the packets based on time delta on Wireshark. Consequently, all 5 attacks were separately tested with the model and the results are shown in Table 4. Regardless of extremely high detection rates, the first two attacks introduced a huge number of false positives leading to a very low precision. After further investigation, we realized that the false positives belong to a laptop in the environment that produced packets with different configurations. This highlights the fact that this method is well suited with IoT devices, that produce fairly simple network traffic.

5.4.4. DoS attacks

There are other ways to detect DoS attacks as they have extremely high transmission rates. However, we tested the model on the first attack of the first .pcap file of all DoS attacks in the CICIoT2023 dataset. The results are shown in Table 5. We detected very high performance in DoS attack and the main reason is the huge amount of attack traffic that makes the data imbalanced. This is also another reason that we chose to only detect 10,000 packets per attacks as the performance will increase even further. However, the method is able to detect almost all the abnormal flows with a very low number of false positives.

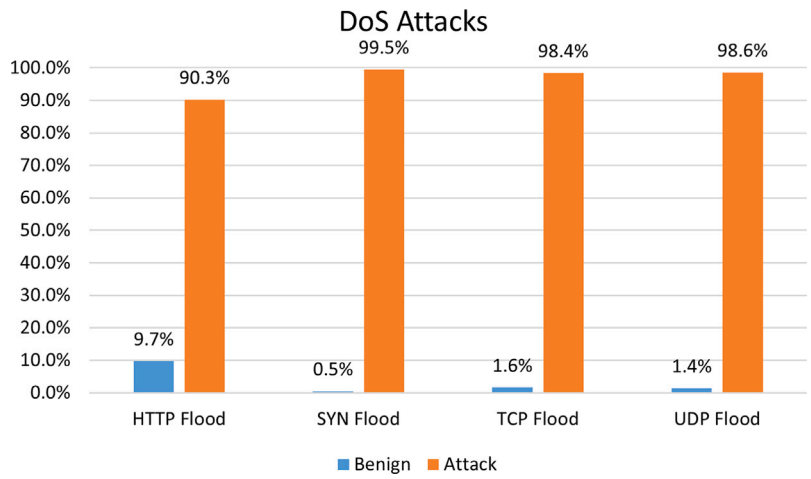


Fig. 6. Percentage of benign and attack for DoS attacks.

Table 5
Performance of the model for detection of DoS Attacks using CICIoT2022 and CICIoT2023 datasets for packet-based and flow-based intrusion detection.

Attack types	Metrics	Packets		Flows	
		2022	2023	2022	2023
HTTP Flood	Accuracy	0.8020	1.0000	0.5584	1.0000
	Precision	0.9480	1.0000	0.9944	1.0000
	Recall	0.8247	1.0000	0.5136	1.0000
	F1-score	0.8821	1.0000	0.6774	1.0000
SYN Flood	Accuracy	0.9997	0.9997	0.9996	0.9996
	Precision	0.9997	0.9997	0.9996	0.9996
	Recall	1.0000	1.0000	1.0000	1.0000
	F1-score	0.9998	0.9998	0.9998	0.9998
TCP Flood	Accuracy	0.9836	0.9959	0.9973	0.9993
	Precision	0.9824	0.9955	0.9972	0.9992
	Recall	1.0000	1.0000	1.0000	1.0000
	F1-score	0.9911	0.9978	0.9986	0.9996
UDP Flood	Accuracy	0.9935	1.0000	0.9997	1.0000
	Precision	0.9929	1.0000	0.9997	1.0000
	Recall	1.0000	1.0000	1.0000	1.0000
	F1-score	0.9964	1.0000	0.9998	1.0000

Fig. 6 also shows the percentage of benign and attack data for DoS attacks. Since DoS attacks have significantly higher attack rates, we present the stats based on the number of flows. Nevertheless, the data is way more imbalanced packet-wise. Regardless, our model is also able to detect the attacks with a very high accuracy for data imbalanced towards attack traffic.

5.4.5. DDoS attacks

The same experiments were conducted on the DDoS attacks of the CICIoT2023 dataset. We excluded the synonymous IP flood attack from our experiments as it: uses the victims IP address for both source and destination IPs; can be easily detected using rules; and is not in the scope of this work (as we focus on tcp and udp attacks). The results are shown in Table 6. The same explanations about DoS attacks apply to DDoS attacks as they have higher transmission rates. In these experiments, we notice a very low performance on the detection of HTTP Flood and SlowLoris while profiling with the CICIoT2023 dataset. After investigation, we realized that in these two samples of attacks, there have been some packets transmitted between the attacker and victim, and therefore there are attack representations in the normal profiles. This is why the performance using the 2022 profiles is impressive as it highlights the importance of the normal data used for profiling while not having attack packets. Regardless, other HTTP Flood and SlowLoris attacks directed towards many other endpoints gave great results.

Table 6
Performance of the model for detection of DDoS Attacks using CICIoT2022 and CICIoT2023 datasets for packet-based and flow-based intrusion detection.

Attack types	Metrics	Packets		Flows	
		2022	2023	2022	2023
ACK Fragment	Accuracy	0.9942	1.0000	0.9993	1.0000
	Precision	0.9937	1.0000	0.9993	1.0000
	Recall	1.0000	1.0000	1.0000	1.0000
	F1-score	0.9969	1.0000	0.9997	1.0000
HTTP Flood	Accuracy	0.9918	0.9948	0.4919	0.1418
	Precision	0.9914	0.9945	0.9980	0.9887
	Recall	1.0000	1.0000	0.4353	0.1129
	F1-score	0.9957	0.9973	0.6062	0.2026
PSH ACK Flood	Accuracy	0.9798	1.0000	0.9987	1.0000
	Precision	0.9694	1.0000	0.9987	1.0000
	Recall	1.0000	1.0000	1.0000	1.0000
	F1-score	0.9844	1.0000	0.9994	1.0000
RST FIN Flood	Accuracy	0.9754	0.9994	0.9959	0.9983
	Precision	0.9378	0.9984	0.9956	0.9982
	Recall	1.0000	1.0000	1.0000	1.0000
	F1-score	0.9679	0.9992	0.9978	0.9991
Slow Loris	Accuracy	0.9782	0.9976	0.3838	0.1998
	Precision	0.9711	0.9967	0.9461	0.0000
	Recall	1.0000	1.0000	0.0959	0.0000
	F1-score	0.9853	0.9984	0.1742	0.0000
SYN Flood	Accuracy	0.8083	1.0000	0.9974	1.0000
	Precision	0.6849	1.0000	0.9967	1.0000
	Recall	1.0000	1.0000	1.0000	1.0000
	F1-score	0.8130	1.0000	0.9983	1.0000
TCP Flood	Accuracy	1.0000	0.9997	1.0000	0.9997
	Precision	1.0000	0.9997	1.0000	0.9997
	Recall	1.0000	1.0000	1.0000	1.0000
	F1-score	1.0000	0.9998	1.0000	0.9998
UDP Flood	Accuracy	0.9857	0.9973	0.9701	0.9957
	Precision	0.9825	0.9967	0.6316	0.9231
	Recall	1.0000	1.0000	1.0000	1.0000
	F1-score	0.9912	0.9983	0.7742	0.9600
UDP Fragment	Accuracy	0.9941	1.0000	0.9577	1.0000
	Precision	0.9939	1.0000	0.7500	1.0000
	Recall	1.0000	1.0000	1.0000	1.0000
	F1-score	0.9970	1.0000	0.8571	1.0000

5.4.6. Performance comparison

In this section, we compare our results with the state-of-the-art. Table 7 shows the results from recent works on the CICIoT2023 dataset. Compared to other approaches, our model is an anomaly-based approach in which it has only seen the benign traffic and comparison would not be completely fair to our model. Regardless, our model

Table 7
Comparison of the performance our method with the state-of-the-art.

Author	Accuracy	Precision	Recall	F1-score
Neto et al. (2023a)	0.9968	0.9654	0.9652	0.9653
Yao et al. (2023)	0.9899	0.9850	0.9742	0.9792
Roshan and Zafar (2024)	0.9893	0.9950	0.9940	0.9945
Khan and Alkhathami (2024)	0.9955	0.9955	0.9955	0.9955
Jeffrey et al. (2024)	0.9319	0.9353	0.9319	0.9324
Our method	0.9874	0.9384	**0.9971**	0.9529

Table 8
Attack Traffic Duration (TD) vs. Detection Duration (DD).

Attacks		# Endpoints	TD (s)	DD (s)
Web Attacks	Backdoor Malware	57	24	2
	Browser Hijacking	82	224	25
	Command Injection	81	502	26
	SQL Injection	82	183	35
	Uploading Attack	75	698	27
	Cross Site Scripting	78	541	27
Recon Attacks	Host Discovery	85	107	13
	OS scan	81	70	26
	Port scan	76	60	15
	Vulnerability scan	83	152	30
Dictionary Attacks	SSH BF 1	79	61	24
	SSH BF 2	79	61	19
	SSH BF 3	79	195	26
	RTSP BF 1	69	60	13
	RTSP BF 2	76	60	10
DoS Attacks	HTTP Flood	66	12	8
	SYN Flood	29	2	4
	TCP Flood	67	451	29
	UDP Flood	70	14	5
DDoS Attacks	ACK Fragment	70	17	14
	HTTP Flood	59	8	16
	PSH ACK Flood	83	36	27
	RST FIN Flood	85	46	31
	Slow Loris	81	44	26
	SYN Flood	76	40	26
	TCP Flood	19	1	13
	UDP Flood	77	29	9
	UDP Fragment	66	10	2

strongly competes with the previous work. This is while it is a light-weight method and requires much less time for profiling and monitoring, compared to training and testing in ML-based approaches.

5.5. Efficiency of IoT-PRIDS (RQ2)

In this section, we provide details about the network traffic duration and execution time to see if IoT-PRIDS can be used in real world applications. We used a mid-level Dell laptop with a 13th Gen Intel(R) Core(TM) i7-1360P 2.20 GHz processor and 16.0 GB of RAM running a 64-bit Windows 11 OS. This device has been used to test the efficiency of IoT-PRIDS in a network of around 80 endpoints including all single-cast, multicast and broadcast Ethernet addresses. Table 8 specifies the number of endpoints, duration of the test traffic and the duration of the detection for all the attacks tested in the CICIoT2023 dataset.

Fig. 7 gives a visual graph for better comparison. It is obvious that for the majority of the attacks, the duration of the detection is significantly less than the traffic duration which signifies that IoT-PRIDS is light-weight enough to be deployed in a real-world application for a near real time intrusion detection. The execution time includes mapping packets to representations, finding the representation distance for each of them to find malicious packets and finally, finding all malicious flows based on the packet-based analysis.

5.5.1. Efficiency comparison

So far, we have shown how our model can be used in a real-time situation by comparing the traffic duration and detection duration. Now, we proceed to compare our model with a machine learning model. As mentioned earlier, IoT-PRIDS works by creating a baseline from the benign traffic in the profiling phase and tests packets for anomalies in the monitoring phase. In order to make a fair comparison of time efficiency with other approaches, we decided to compare our model with autoencoders as they can also be used in anomaly detection by creating a baseline from the benign traffic in the training phase and detecting anomalies in the testing phase. We first develop a formal time-complexity analysis of both approaches and then provide runtime analysis of both methods on the same machine. Fig. 8 shows the correspondence between IoT-PRIDS and autoencoders:

Mapping-feature extraction. Given benign traffic with n packets, we have the following equations:

$$\begin{aligned} T_{mapping} &= n \cdot T_{p\to r} \\ &= n \cdot T_{p\to v} \\ &= T_{featureExtraction} \end{aligned} \tag{2}$$

where $T_{p\to r}$ is the time it takes to map a packet to a representation and $T_{p\to v}$ is the time it takes to extract features from a packet. It is obvious that the time complexity of mapping and feature extraction are equal as we choose/engineer different features from all packets.

Profiling-training. Given benign traffic with n packets, the profiling phase only involves storing the representations and therefore is $O(1)$. However, the training time complexity of an autoendcoder can be found using the following equations:

$$\begin{aligned} T_{training} &= n \cdot (T_{FeedForward} + T_{ReconstructionError} + T_{BackPropagation}) \\ &\le n \cdot (O(n^2) + O(n) + O(n^2)) \\ &\le O(n^3) \end{aligned} \tag{3}$$

It is evident that IoT-PRIDS outperforms an autoencoder (or any other ML model that involves training) in terms of profiling/training.

Monitoring-testing. Given unknown traffic with n packets and assuming from each packet m features are extracted, the monitoring phase of IoT-PRIDS involves mapping all the packets and finding their distance with the stored representations in the profiling phase. Therefore, we have:

$$\begin{aligned} T_{profiling} &= n \cdot T_{p\to r} + n \cdot T_{distance} \\ &= n \cdot T_{p\to r} + n \cdot k \cdot m \\ &= O(n) + O(n \cdot k \cdot m) \\ &\le O(n^3) \end{aligned} \tag{4}$$

where k is the size of the largest set of representations among all device profiles. In other words, for each packet we first map it to the corresponding representation, and then find the distance using Eq. (1) by finding the minimum Hemming distance of the packet representation from other normal representations for the device that sent the packet. It is obvious that $m, k < n$. Now, consider the simplest autoencoder with one input layer of m neurons, one hidden layer of $h < m$ nodes and one output layer of m neurons. The time complexity of such autoencoder can be found using the following equations:

$$\begin{aligned} T_{testing} &= n \cdot (T_{p\to v} + T_{FeedForward} + T_{ReconstructionError}) \\ &= n \cdot (O(1) + O(m \cdot k + k \cdot m) + O(n)) \\ &\le O(n) + O(n^3) + O(n^2) \\ &\le O(n^3) \end{aligned} \tag{5}$$

where the time for the feed forwarding of the input vector involves two consecutive vector-by-matrix multiplications.

Conclusion of the comparison. Based on the previous analyses, it is evident that IoT-PRIDS is much lighter in general. Although the time complexity of both methods are the same theoretically, the autoencoder does not necessarily give the same results and a high accuracy as it

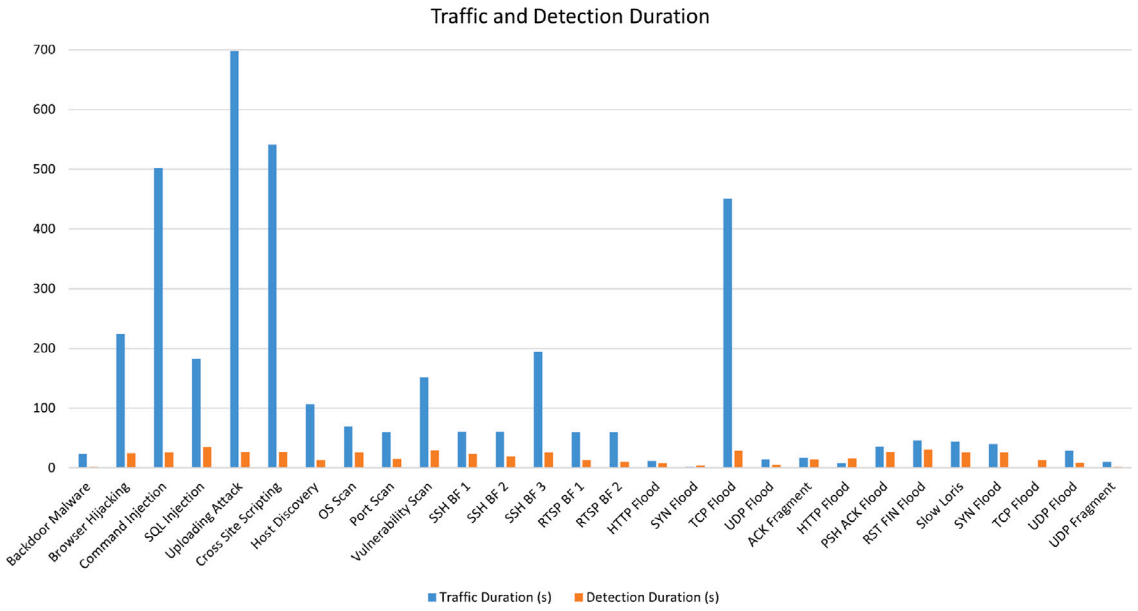


Fig. 7. Comparing duration of attacks with their detection time.

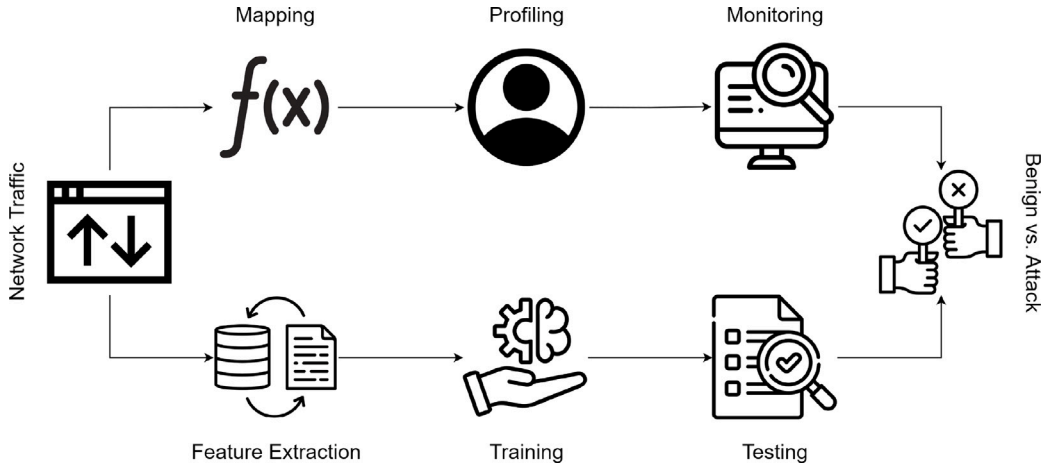


Fig. 8. Comparing IoT-PRIDS with machine learning.

might need more layers with more neurons to work as well as our model. Moreover, the number of features required for an autoencoder might be much more to obtain the same results in a real-world setting. Lastly, updating IoT-PRIDS takes as much as mapping new benign packets and adding them to the representation set, while updating an autoencoder will take another round of training to update. Overall, the time complexity and runtime overhead of an autoencoder will be much more than those of IoT-PRIDS.

6. Limitations and future work

While our system has great potential for real-time intrusion detection with a low false positive rate, there are still some limitations and challenges that open up directions for future work.

Since this is a host-based method, the results heavily rely on the knowledge of the model about each individual device; i.e. the model needs to have profiled a decent number of representations to be able to summarize the behavior of the device. And this number can be different for each device due to different complexities of IoT devices. This dependence may cause a huge number of false positives for the unseen devices. Therefore, it is well-applicable in the case of IoT networks with known static IoT devices. However, the more dynamism added in terms of joining and leaving of new devices to the network, the more false positives the model will introduce.

Furthermore, the representation mapping and distance defined is a baseline approach that can be further elaborated. One approach would be to investigate application layer protocol headers that will indeed improve the accuracy of the model, although one needs to make a balance to not generate representations with a large number of features to reduce time complexity, as discussed previously. Secondly, the representation distance can be further enhanced by giving weights to more important representation features and defining new distance thresholds to separate normal and attack packets further. This is a direction for future research on this approach.

Lastly, in order to improve the accuracy of the model over time, one may use the model as a network intrusion detection system in an incremental learning process by introducing a human-in-the-loop agent that gives feedback to the model and enhances the detection capabilities. This will continue until the device profile reaches a point of equilibrium in which all the possible normal traffic from a device has been seen and added to the profile.

IoT-PRIDS can be thought of as a host intrusion detection system to be deployed on IoT devices as it does not require sophisticated

calculations and processing power. Future research could focus on reducing false positives.

7. Conclusion

In this work, we proposed a novel approach for host-based IoT device intrusion detection that uses the concept of "packet representation", which is a fixed size vector that represents a large number of packets from a single device, to create a normal profile for the device's behavior. This profile can then be used in a monitoring phase as a reference to the device's normal behavior to detect abnormal and suspicious network traffic.

IoT-PRIDS is a light-weight method that does not use any heavy Machine Learning methods and can operate in near real-time scenarios, as it only requires parsing packets, mapping them to representations and searching the representation sets (normal profiles) to find the distance. Therefore, it is well-suited to be deployed on IoT devices, hubs, gateways, etc. as both a host IDS or a network IDS.

We tested our model on 5 different categories of attacks from the CICIoT2023 dataset and achieved significant results based on precision and recall which demonstrated that most attacks are detected while minimizing the overall number of false positives. Furthermore, we analyzed the efficiency of our model to see if it can be adopted for real-world use cases. The results show that this approach is light-weight enough to be promising for real time intrusion detection in real world applications.

CRediT authorship contribution statement

Alireza Zohourian: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Sajjad Dadkhah:** Writing – review & editing, Supervision, Project administration, Methodology, Data curation. **Heather Molyneaux:** Writing – review & editing, Visualization, Supervision. **Euclides Carlos Pinto Neto:** Writing – review & editing, Visualization, Supervision. **Ali A. Ghorbani:** Writing – review & editing, Supervision, Project administration, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors graciously acknowledge the support from the Canadian Institute for Cybersecurity (CIC). This project was also supported in part by collaborative research funding from the National Research Council of Canada's Artificial Intelligence for Logistics Program.

References

Abdel-Basset, M., Hawash, H., Chakrabortty, R.K., Ryan, M.J., 2021. Semi-supervised spatiotemporal deep learning for intrusions detection in IoT networks. IEEE Internet Things J. 8 (15), 12251–12265.

Abdulla, A.R., Jameel, N.G.M., 2023. A review on IoT intrusion detection systems using supervised machine learning: Techniques, datasets, and algorithms. UHD J. Sci. Technol. 7 (1), 53–65.

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

Alahi, M.E.E., Sukkuea, A., Tina, F.W., Nag, A., Kurdthongmee, W., Suwannarat, K., Mukhopadhyay, S.C., 2023. Integration of IoT-enabled technologies and artificial intelligence (AI) for smart city scenario: recent advancements and future trends. Sensors 23 (11), 5206.

Albulayhi, K., Smadi, A.A., Sheldon, F.T., Abercrombie, R.K., 2021. IoT intrusion detection taxonomy, reference architecture, and analyses. Sensors 21 (19), 6432.

Alsoufi, M.A., Razak, S., Siraj, M.M., Nafea, I., Ghaleb, F.A., Saeed, F., Nasser, M., 2021. Anomaly-based intrusion detection systems in iot using deep learning: A systematic literature review. Appl. Sci. 11 (18), 8383.

Bacha, S., Aljuhani, A., Abdellafou, K.B., Taouali, O., Liouane, N., Alazab, M., 2024. Anomaly-based intrusion detection system in IoT using kernel extreme learning machine. J. Ambient Intell. Humaniz. Comput. 15 (1), 231–242.

Baz, M., 2022. SEHIDS: Self evolving host-based intrusion detection system for IoT networks. Sensors 22 (17), 6505.

Bhatia, R., Benno, S., Esteban, J., Lakshman, T., Grogan, J., 2019. Unsupervised machine learning for network-centric anomaly detection in IoT. In: Proceedings of the 3rd Acm Conext Workshop on Big Data, Machine Learning and Artificial Intelligence for Data Communication Networks. pp. 42–48.

Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., Faruki, P., 2019. Network intrusion detection for IoT security based on learning techniques. IEEE Commun. Surv. Tutor. 21 (3), 2671–2701.

Dadkhah, S., Mahdikhani, H., Danso, P.K., Zohourian, A., Truong, K.A., Ghorbani, A.A., 2022. Towards the development of a realistic multidimensional IoT profiling dataset. In: 2022 19th Annual International Conference on Privacy, Security & Trust. PST, IEEE, pp. 1–11.

DeMedeiros, K., Hendawi, A., Alvarez, M., 2023. A survey of AI-based anomaly detection in IoT and sensor networks. Sensors 23 (3), 1352.

Faraj, O., Megías, D., Ahmad, A.-M., Garcia-Alfaro, J., 2020. Taxonomy and challenges in machine learning-based approaches to detect attacks in the internet of things. In: Proceedings of the 15th International Conference on Availability, Reliability and Security. pp. 1–10.

Fraihat, S., Makhadmeh, S., Awad, M., Al-Betar, M.A., Al-Redhaei, A., 2023. Intrusion detection system for large-scale IoT NetFlow networks using machine learning with modified Arithmetic Optimization Algorithm. Internet Things 100819.

Ge, M., Cho, J.-H., Kim, D., Dixit, G., Chen, I.-R., 2021. Proactive defense for internet-of-things: moving target defense with cyberdeception. ACM Trans. Internet Technol. (TOIT) 22 (1), 1–31.

Gyamfi, E., Jurcut, A., 2022. Intrusion detection in internet of things systems: a review on design approaches leveraging multi-access edge computing, machine learning, and datasets. Sensors 22 (10), 3744.

Harsha, P., 2024. Deep neural networks-based combined network and host intrusion classification system in internet of things environment. J. Interdiscip. Cycle Res. 16 (1), 691–701.

Heidari, A., Jabraeil Jamali, M.A., 2023. Internet of things intrusion detection systems: a comprehensive review and future directions. Cluster Comput. 26 (6), 3753–3780.

Hussain, F., Hussain, R., Hassan, S.A., Hossain, E., 2020. Machine learning in IoT security: Current solutions and future challenges. IEEE Commun. Surv. Tutor. 22 (3), 1686–1721.

Idrissi, I., Azizi, M., Moussaoui, O., 2020. IoT security with deep learning-based intrusion detection systems: A systematic literature review. In: 2020 Fourth International Conference on Intelligent Computing in Data Sciences. ICDS, IEEE, pp. 1–10.

Jeffrey, N., Tan, Q., Villar, J.R., 2024. Using ensemble learning for anomaly detection in cyber–physical systems. Electronics 13 (7), 1391.

Khan, M.M., Alkhathami, M., 2024. Anomaly detection in IoT-based healthcare: machine learning for enhanced security. Sci. Rep. 14 (1), 5872.

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

Kumari, P., Jain, A.K., 2023. A comprehensive study of DDoS attacks over IoT network and their countermeasures. Comput. Secur. 127, 103096.

Li, W., Tug, S., Meng, W., Wang, Y., 2019. Designing collaborative blockchained signature-based intrusion detection in IoT environments. Future Gener. Comput. Syst. 96, 481–489.

Martins, I., Resende, J.S., Sousa, P.R., Silva, S., Antunes, L., Gama, J., 2022. Host-based IDS: A review and open issues of an anomaly detection system in IoT. Future Gener. Comput. Syst. 133, 95–113.

Mishra, N., Pandya, S., 2021. Internet of things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review. IEEE Access 9, 59353–59377.

Neto, E.C.P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., Ghorbani, A.A., 2023a. CicIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment. Sensors 23 (13), 5941.

Neto, E.C.P., Dadkhah, S., Sadeghi, S., Molyneaux, H., Ghorbani, A.A., 2023b. A review of machine learning (ML)-based IoT security in healthcare: A dataset perspective. Comput. Commun..

Otoum, Y., Nayak, A., 2021. As-ids: Anomaly and signature based ids for the internet of things. J. Netw. Syst. Manage. 29 (3), 23.

Rizvi, S., Orr, R., Cox, A., Ashokkumar, P., Rizvi, M.R., 2020. Identifying the attack surface for IoT network. Internet Things 9, 100162.

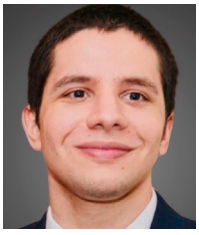
Roshan, K., Zafar, A., 2024. Ensemble adaptive online machine learning in data stream: a case study in cyber intrusion detection system. Int. J. Inf. Technol. 1–14.

Roy, S., Li, J., Choi, B.-J., Bai, Y., 2022. A lightweight supervised intrusion detection mechanism for IoT networks. Future Gener. Comput. Syst. 127, 276–285.
Saif, S., Das, P., Biswas, S., Khari, M., Shanmuganathan, V., 2022. HIIDS: Hybrid intelligent intrusion detection system empowered with machine learning and metaheuristic algorithms for application in IoT based healthcare. Microprocess. Microsyst. 104622.
Sarker, I.H., Khan, A.I., Abushark, Y.B., Alsolami, F., 2023. Internet of things (iot) security intelligence: a comprehensive overview, machine learning solutions and research directions. Mob. Netw. Appl. 28 (1), 296–312.
Satilmiş, H., Akleylek, S., Tok, Z.Y., 2024. A systematic literature review on host-based intrusion detection systems. IEEE Access 12, 27237–27266.
Schiller, E., Aidoo, A., Fuhrer, J., Stahl, J., Ziörjen, M., Stiller, B., 2022. Landscape of IoT security. Comp. Sci. Rev. 44, 100467.
Spadaccino, P., Cuomo, F., 2020. Intrusion detection systems for IoT: opportunities and challenges offered by edge computing and machine learning. arXiv preprint arXiv:2012.01174.
Thabit, F., Can, O., Aljahdali, A.O., Al-Gaphari, G.H., Alkhzaimi, H.A., 2023. A comprehensive literature survey of cryptography algorithms for improving the iot security. Internet Things 100759.
Thakkar, A., Lohiya, R., 2021. A review on machine learning and deep learning perspectives of IDS for IoT: recent updates, security issues, and challenges. Arch. Comput. Methods Eng. 28 (4), 3211–3243.
Yao, W., Zhao, H., Shi, H., 2023. Privacy-preserving collaborative intrusion detection in edge of internet of things: A robust and efficient deep generative learning approach. IEEE Internet Things J..

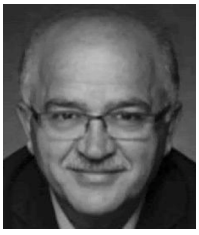
Alireza Zohourian is a Ph.D. student at the University of New Brunswick (UNB) and a Cybersecurity Researcher at the Canadian Institute for Cybersecurity (CIC). His main research is on the Internet of Things (IoT) Security. He started his research by studying the security of low-power low-rate wireless protocols such as Zigbee and extended his research to state-of-the-art solutions such as Artificial Intelligence, Blockchain and Software-Defined Networking to secure IoT networks. His main research is on profiling, fingerprinting and identification techniques for behavioral monitoring of IoT networks. He is currently conducting research on IoT Anomaly/Intrusion Detection to address the challenges caused by the proliferation, heterogeneity, and resource constraints of IoT devices in different IoT applications.

Dr. Sajjad Dadkhah is currently an Assistant professor of computer science and cybersecurity research & Development (R&D) team leader at the Canadian Institute for Cybersecurity, University of New Brunswick. He has more than ten years of experience in different fields of cybersecurity hand and has made significant contributions to digital multimedia security, artificial intelligence security, natural language processing, IoT security, and machine learning-based detection systems. His expertise as a team leader has been pivotal in various prestigious organizations, including Kyushu University (Japan), University Malaya (UM), IRIS Smart Technology Complex, and Kyushu Institute of Technology (Japan). He is the founder of several cybersecurity datasets in the IoT and NLP security domains. Dr. Dadkhah served as the Managing Editor and Board member of the highest-rank Q1 Journal of Applied Soft Computing (ASOC) for over eight years. His current interests include Human-centric Cybersecurity, AI-driven threat modeling, Security architectures with large language models and Gen- AI algorithms.

Heather Molyneaux is a research council officer at the National Research Council of Canada (NRC) in the Cybersecurity team of the Digital Technologies research center. Prior to joining the Cybersecurity team, Heather had more than a decade of experience working with the Human Computer Interaction (HCI) team at the NRC. Heather Molyneaux has experience in dozens of NRC projects and collaborations and is the author of more than 60 publications as well as numerous internal reports. Her research interests are in the field of human factors and human computer interaction in the creation of usable security, behavioral information security, user's security and privacy perceptions, authentication and access management, human centric cybersecurity and ethics. She applies these interests to projects in the fields of education, healthcare, and critical infrastructure.



Euclides Carlos Pinto Neto is a Postdoctoral Fellow at the University of New Brunswick (UNB). He holds a B.Sc. in Computer science (Federal Rural University of Pernambuco – UFRPE - & Athlone Institute of Technology - AIT), and an M.Sc. and Ph.D. in Computer Engineering (2018 and 2021, Digital Systems Department at Poli-USP, University of São Paulo). His research is focused on the applications of artificial intelligence techniques to security and safety–critical systems.



Prof. Ali A. Ghorbani is currently a Professor of computer science, the Tier 1 Canada Research Chair in cybersecurity, and the Director of the Canadian Institute for Cybersecurity, which he established, in 2016. He is the Co-Inventor of three awarded and one filed patent in the fields of cybersecurity and web intelligence. He has published over 280 peer-reviewed articles during his career. He has supervised over 190 research associates, postdoctoral fellows, and students during his career. His book Intrusion Detection and Prevention Systems: Concepts and Techniques (Springer, October 2010). He is the Co-Founder of the Privacy, Security, Trust (PST) Network in Canada and its annual international conference. He has served as the Co-Editor-in-Chief for the International Journal of Computational Intelligence, from 2007 to 2017.