Intruders' Behavior Unveiled: A Dual-Tier Behavior-driven Model for Malicious Activity Detection in IoT Network Using Graph Learning

MohammadMoein Shafi

A Thesis Submitted to the Faculty of Graduate Studies
In Partial Fulfillment of the Requirements
for the Degree of Master of Computer Science

Graduate Program in Electrical Engineering and Computer Science

York University
Toronto, Ontario

December 2024

Abstract

In recent years, IoT technology has transformed smart homes, with most households now including several IoT devices that provide convenience and automation. However, the security of these smart homes is paramount, as vulnerabilities can expose residents to risks like unauthorized access, data breaches, and operational disruption. Network-based threats pose a particularly critical risk due to the numerous vulnerabilities in wireless communication between devices, making it possible for attackers to intercept data or do malicious activities. While traditional intrusion detection systems exist, they are often ineffective in detecting zero-day attacks and lack the ability to identify malicious patterns across diverse threat scenarios due to limited diversity in their detection models. Moreover, these systems are not designed to fully detect all types of intrusions, especially those involving both external network activities and internal IoT communications among smart home devices. This gap is made worse by the challenges in creating specialized IoT datasets that cover a diverse set of malicious activities and data types, which require extensive technical knowledge, a diverse range of devices, and expertise in capturing, executing, and labeling attack scenarios. Such datasets are crucial for data-driven intrusion detection systems. Addressing these challenges, this thesis introduces a dual-tier detection system that effectively can zero-day attacks, and is designed in a way to be scalable for learning the behavior of diverse malicious activities. the proposed solution leverages data from both the smart home hub's internet connection and the internal network communication of IoT devices to detect and profile malicious activities using a novel graph learning approach. Furthermore, to support this research, we have created the largest IoT smart home dataset, incorporating real-world data from over 50 devices and more than 100 carefully designed attack scenarios, captured over a five-month period. The analysis of this dataset and the performance of our detection model demonstrate promising results, providing a valuable resource and foundation for advancing smart home IoT security.

Dedication

This milestone is dedicated to the memory of my dear friend, Ebrahim Hadi, whose presence I deeply felt even though he is no longer with us. I also dedicate this work to my beloved family, whose encouragement and sacrifices have made this journey possible. Finally, I dedicate this thesis to all those who strive for knowledge and innovation, as a testament to the power of perseverance and collaboration in shaping a better future.

Acknowledgement

I would like to express my sincere gratitude to my supervisor, Prof. Arash Habibi Lashkari, for his invaluable guidance and support throughout this journey. I would also like to thank my friend, Ebrahim Hadi, without whose support this achievement would not have been possible, and to my family for their unwavering encouragement and constant support. Lastly, I extend my appreciation to all my friends for their constant encouragement and companionship along the way.

Contents

List of Tables

List of Figures

1 Introduction

The rapid advancement and accessibility of IoT (Internet of Things) technology have transformed everyday life, particularly within smart homes. From security cameras and smart thermostats to virtual assistants and interconnected lighting systems, IoT devices provide convenience, automation, and energy efficiency. Today, finding a home without at least a couple of IoT devices is uncommon, all contributing to a seamless and interconnected living environment. However, with this increased integration of IoT technology, there has been a parallel rise in the potential vulnerabilities associated with these devices, which can create serious security concerns [34, 35].

Securing smart homes has become increasingly critical due to the potential consequences of unauthorized access, data breaches, and service disruptions. A compromised IoT device can expose sensitive data, invade user privacy, and even facilitate further network breaches, affecting other connected devices and potentially impacting users' safety. Given the open nature of wireless communication and the evolving tactics of cyber attackers, smart home networks are a prime target for malicious activities. While IoT technology introduces countless benefits, it also requires robust security measures to mitigate these risks effectively [36].

Network-based threats represent a significant portion of IoT devices' risks in smart homes. The reliance on wireless communication protocols such as Wi-Fi, Zigbee, Z-Wave, and Bluetooth leaves these devices vulnerable to interception, data manipulation, and other forms of attack. Attackers can exploit these wireless networks to intercept data transmissions or inject harmful code, leading to unauthorized control over devices or leakage of personal data. Despite the critical nature of these network-based vulnerabilities, the current generation of intrusion detection systems (IDS) tends to focus primarily on traditional IT infrastructure security, overlooking the unique characteristics and requirements of IoT networks. The few IDS frameworks developed specifically for IoT environments often mirror traditional security methods without accommodating internal IoT device-to-device communication nuances [37, 38].

One major reason for this gap in security solutions is the scarcity of datasets that capture detailed IoT communication within smart home environments. Building a dataset for IoT networks is inherently challenging due to the diversity of devices, complex setup requirements, data capturing, and the need for a wide range of attack scenarios. Such a dataset requires comprehensive knowledge of IoT testbed setup, the execution of realistic attack scenarios, and detailed data labeling and analysis. Without high-quality, diverse data, developing effective and specialized security solutions becomes difficult for researchers and developers alike.

To address these challenges, this thesis introduces a multi-tier intrusion detection system designed to secure smart home environments. Our proposed system utilizes two primary data sources: the external internet traffic passing through the smart home hub and the internal communications between IoT devices. By analyzing these data streams, the system can detect and profile malicious activities more accurately, providing a layered approach to IoT security that considers the unique characteristics of smart home networks.

Furthermore, to overcome the limitations associated with IoT dataset availability, this work includes the creation of the largest IoT smart home dataset to date and to the best of our knowledge, featuring real-world data from over 50 distinct devices. During a five-month data capture period, we designed, implemented, and executed over 100 unique attack scenarios, covering many potential threat vectors. This dataset is foundational for analyzing malicious behaviors within smart home IoT networks. It enables the training and testing advanced detection models specifically tailored for IoT environments. The analysis of this dataset, coupled with the results from our proposed detection model, has demonstrated promising capabilities in identifying and mitigating threats within smart homes,

setting a strong foundation for future research in IoT security. The main contributions of this research are as follows:

- First, we conduct an in-depth analysis of the Z-Wave protocol and its associated devices, offering insights into its specific vulnerabilities within IoT networks.
- Second, to enhance smart home intrusion detections, we propose a novel multi-tier detection system that leverages internal and external IoT network traffic for more accurate and comprehensive detection and profiling of malicious activities.
- Third, to fill a significant gap in available labeled datasets for smart home security research, we present the largest (to the best of our knowledge) IoT smart home dataset, capturing real-world data from a wide range of devices and over 100 diverse attack scenarios.
- Fourth, to introduce a comprehensive taxonomy of smart home IoT devices, which can serve as a framework for future IoT security studies.
- Finally, we perform a detailed evaluation of current datasets, identifying their limitations and contributing to a clearer understanding of the data needs in IoT security. Collectively, these contributions aim to provide foundational advancements in IoT security research and support further development of robust, specialized solutions for smart home environments.

The structure of this thesis is as follows: Section 2 provides a comprehensive literature review, discussing smart home protocols, relevant research efforts, and identifying key shortcomings in existing security approaches. Section 3 introduces our motivation for the study and presents the proposed multi-tier detection model designed for enhancing smart home security. In Section 4, we describe our created dataset, the architecture of the testbed, detailed descriptions of benign and attack scenarios, a comparative analysis of available IoT datasets, and the proposed IoT smart home device taxonomy. Section 5 details the experimental setup and evaluates the proposed model's performance on the newly created dataset. Section 6 offers a thorough analysis and discussion of results, insights, and implications for future research. Finally, Section 7 concludes the study, summarizing key findings and outlining potential areas for future work.

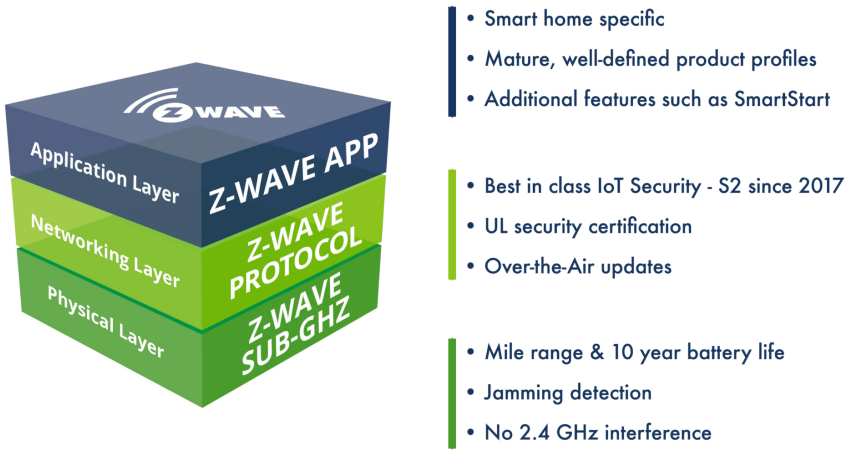


Figure 1: Z-wave protocol stack [1]

2 Literature Review

This section offers a thorough examination of contemporary literature on IoT Smart Homes. Initially, we explore the protocols employed by smart home devices. Subsequently, we delve into an in-depth analysis of leading studies in smart home profiling. Furthermore, leveraging previous studies on smart home architecture, a taxonomy for categorizing smart home devices is developed. Finally, the review identifies potential areas within the current literature that warrant further investigation and improvement.

2.1 Smart Home Devices' Protocols

Smart home devices rely on diverse communication protocols to establish interoperability within connected environments [39, 40]. Each protocol is strategically designed to address specific requirements, offering unique advantages tailored to different device types [41]. The selection of communication protocols in a smart home ecosystem is a critical determinant influencing the efficacy of device integration, network reliability, and overall system performance [40, 42, 43, 44]. Among the prominent protocols in this domain, Z-Wave, Zigbee, and Wi-Fi stand out due to their extensive adoption, versatile applications, and inherent significance in the realm of smart home technologies [45, 46, 47, 48, 49]. Subsequently, a brief explanation of these protocols is provided in the next subsection, with a comparison presented in Table 1. Following this, the subsequent subsection delves into examining and evaluating notable previous works that considered these protocols.

2.1.1 Z-Wave

Z-Wave is a wireless communication protocol optimized for low-power, home automation devices. In the sub-1GHz frequency range, Z-Wave devices form a mesh network, facilitating reliable communication with minimal power consumption. This protocol utilizes a master-slave architecture and employs various modulation techniques to ensure efficient and secure data transmission. Key applications include smart lighting, door locks, and sensors [50, 51, 52, 53, 54]. Fig. 1 illustrates the hierarchical structure of the Z-Wave protocol.

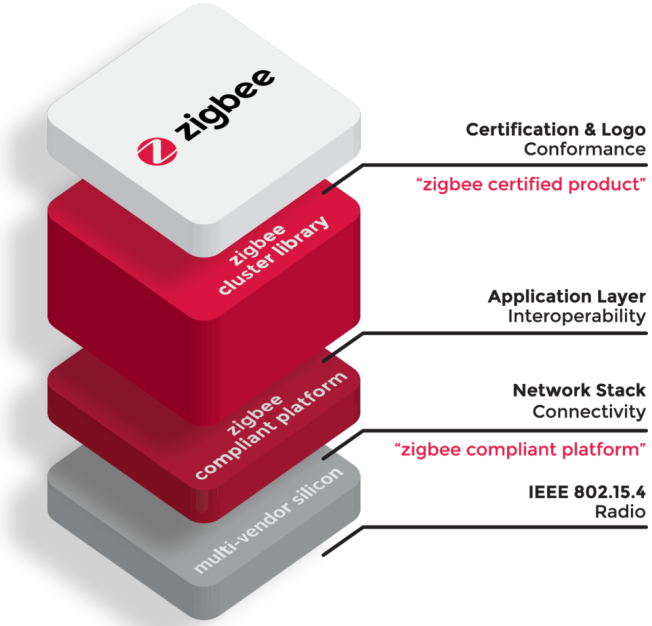


Figure 2: Zigbee protocol stack [2]

2.1.2 Zigbee

Zigbee, a low-power, mesh networking protocol operating on the 2.4 GHz frequency, is renowned for its scalability and flexibility. Zigbee devices use a multi-hop communication approach to create self-healing networks, enhancing overall reliability. Zigbee supports various topologies, including star, tree, and mesh configurations, making it suitable for diverse smart home applications such as smart lighting systems, sensors, and home automation hubs [55, 56, 57, 58, 59, 60]. The hierarchical structure of Zigbee is illustrated in Fig. 2.

2.1.3 Wi-Fi

Wi-Fi, operating on the IEEE 802.11 standard, is a ubiquitous high-bandwidth communication solution for smart home devices. Devices such as smartphones, smart TVs, and cameras leverage Wi-Fi for robust connectivity. The protocol's ability to handle large data transfers and its widespread adoption make it a cornerstone for integrating diverse devices into a unified smart home network [61, 62, 63, 47, 64, 65]. The protocol structure of Wi-Fi is depicted in Fig. 3.

In summary, carefully considering communication protocols is paramount in designing and implementing smart home ecosystems. The comparison presented in Table 1 sheds light on the distinctive features and performance metrics of Z-Wave, Zigbee, and Wi-Fi. The analysis underscores the significance of aligning protocol choices with specific use cases, balancing power consumption, data transfer rates, and network range.

2.2 Previous Works

This section delves into an in-depth analysis of leading IoT smart home profiling studies. The review is structured according to the specific threat models emphasized in the literature. Each reviewed paper is systematically assessed based on the problem statement, proposed solutions, experimental findings, derived insights, and limitations.

Table 1: Protocols' Comparison

Feature	***Z-Wave***	***Zigbee***	***Wi-Fi***
Network Topology	Mesh network	Mesh network	Point-to-point
Frequency Band	908.42 MHz (varies by region)	2.4 GHz (global)	2.4 GHz and 5 GHz (dual-band)
Range	Up to 100 meters (line of sight)	Up to 70 meters (line of sight)	Depends on the Wi-Fi standard (b/g/n/ac) 35 to 70 meters (line of sight)
Data Rate	9.6 to 100 kbps	250 kbps (Zigbee 2007), 1 Mbps (Zigbee Pro)	Up to 3.5 Gbps (Wi-Fi 6)
Power Consumption	Low	Low to Moderate	Moderate to High
Interference	Minimal	Moderate	Susceptible
Security	AES-128 encryption S0 devices are non-secure	AES-128 encryption	WPA3 security
Scalability	Up to 232	Up to 100	Up to 250 (depend on configs)
Installation	Requires a hub/gateway	Requires a hub/gateway	Direct connection to a Wi-Fi router
Cost	More expensive due to licensing fees	Moderately priced devices	Relatively low-cost, as Wi-Fi is ubiquitous
Popular Use Cases	Home automation, lighting control, security systems	Home automation, lighting control, smart appliances	Broad range of smart home devices, streaming, gaming
Standards	Proprietary standard maintained by Silicon Labs	IEEE 802.15.4 standard maintained by IEEE	IEEE 802.11 standards maintained by IEEE
Power Source Options	Devices often support battery-powered and low-power options	Supports a variety of power sources, including battery-operated devices	Devices are typically powered by electrical outlets, limiting portability
Signal Penetration	Excellent penetration through walls and obstacles, benefiting from the sub-1 GHz frequency	Signal penetration may be hindered by walls and obstacles, especially in environments with interference	Signal penetration may be hindered by walls and obstacles, especially in environments with interference

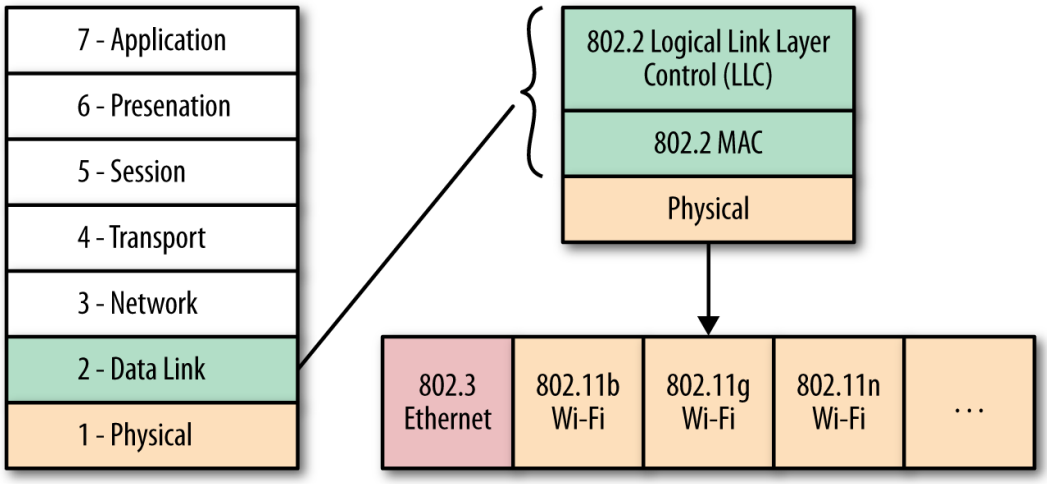


Figure 3: Wifi protocol stack [3]

2.2.1 Device Profiling Works

In this subsection, we explore research efforts focused on device profiling within the context of smart homes. Device profiling aims to characterize the behavior and attributes of individual devices connected to the IoT network. By understanding the distinctive features of each device, researchers seek to enhance security measures and improve overall system resilience.

In [4], the authors address the critical issue of enhancing security in IoT networks through device fingerprint recognition. The identified problem revolves around accurately identifying and classifying IoT devices to prevent unauthorized access and potential cyber threats. To tackle this challenge, the authors propose a novel recognition scheme, presented in Fig. 4, based on deep learning specifically tailored for Z-Wave protocol devices. By leveraging the error Back Propagation (BP) classification algorithm and enhancing the gradient descent strategy, the proposed solution aims to improve the accuracy and efficiency of device recognition.

The authors have thoroughly evaluated the proposed recognition scheme's performance through a series of rigorous experiments. The results are not just promising, but they showcase a significant increase in recognition accuracy. The improved adaptive BP algorithm has outperformed the standard BP algorithm by 5.55%, instilling confidence in the proposed solution's effectiveness. The insights gained from these experiments, particularly the concept of confidence interval, have proven instrumental in mitigating the issue of overlapping identifications, thereby bolstering the overall security of the Z-Wave network system.

While the proposed solution demonstrates promising results in device recognition and security enhancement, the PDF also acknowledges certain limitations. One notable limitation is the lack of an open-source dataset for Z-Wave devices, necessitating the construction of a Z-Wave network for real-time data acquisition. Additionally, the complexity of the experimental process and the requirement for expensive equipment pose practical challenges for widespread implementation. These limitations underscore the need for further research and development to address scalability and accessibility issues in deploying the proposed fingerprint recognition scheme for IoT devices.

In [5], authors address the security problem in IoT smart homes by proposing a solution based on constructing behavioral device templates. The proposed solution involves combining statistical and machine learning techniques to calculate behavioral device templates according to the network behavior captured within a smart home. The statistical metrics generated are processed to produce features used for constructing clusters of devices. The main

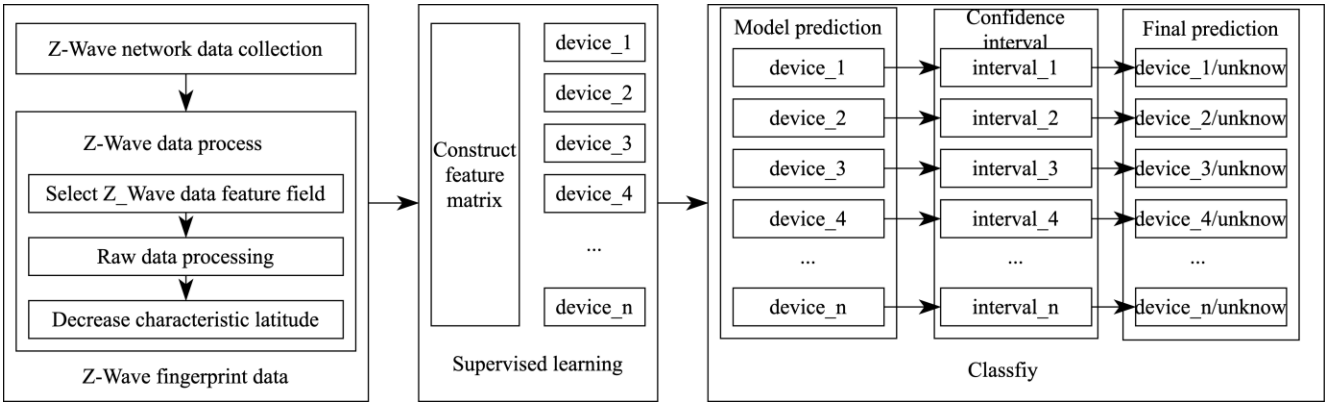


Figure 4: Proposed model by [4]

idea is that during an abnormal event, the device will be moved away from the cluster's center, generating an alert for further mitigation actions. The methodology, as illustrated in Fig. 5, consists of a Training Phase and a Running Phase, where statistical and machine learning methodologies are combined to detect abnormal behavior of IoT devices. The Training Phase periodically updates the models, while the Running Phase continuously uses the models from the Training Phase. The proposed methodology was validated on a real smart home dataset, and experiments were conducted to detect two types of attacks: Physical Damage and Mechanical Exhaustion.

The experiment results presented in the paper are encouraging, as the proposed methodology managed to detect both types of attacks. However, the paper also highlights certain limitations. It was observed that the detection of Mechanical Exhaustion was more accessible than the detection of Physical Damage, indicating that the normal network behavior of smart home devices is relatively rare, making the detection of Physical Damage more challenging. This insight suggests that the methodology's performance may vary depending on the type of attack and the specific characteristics of the devices involved. Also, there is a need for more experiments to validate the methodology's ability to detect different types of attacks. Additionally, the testbed is limited to only five devices in a laboratory environment.

The identified problem in [6] is the lack of mechanisms to identify unauthorized IoT devices communicating via ZigBee or Z-Wave networks. While there are mechanisms to identify unauthorized IoT devices communicating via IP networks, these protocols remain vulnerable. This poses a security risk as attackers can potentially spoof these devices and gain unauthorized access to the network. To overcome this, the proposed solution, as shown in Fig. 6, is the introduction of Z-IoT, a device-class fingerprinting framework for IoT devices utilizing ZigBee and Z-Wave protocols. Z-IoT leverages passive packet capturing tools, optimal filtering, and machine learning algorithms to passively monitor idle network traffic and classify different IoT device types based on the inter-arrival times of packets.

The experiment results demonstrate the efficacy and robustness of Z-IoT. The framework was tested with 39 ZigBee and Z-Wave IoT devices and exhibited superior performance in identifying different types of devices. The average precision and recall of Z-IoT were reported to be over 91%. Implementing Z-IoT provides insights into the potential of using passive packet capturing and machine learning algorithms for device-class fingerprinting in IoT networks. The framework's ability to accurately identify and classify different types of IoT devices highlights the importance of considering inter-arrival times of packets as a distinguishing feature. This insight can contribute to developing more secure and reliable IoT networks. However, it is important to note that the effectiveness of Z-IoT may depend on the specific network environment and the types of IoT devices being analyzed. Additionally, the performance of Z-IoT may be influenced by factors such as applying different types of attacks and variations in device behavior.

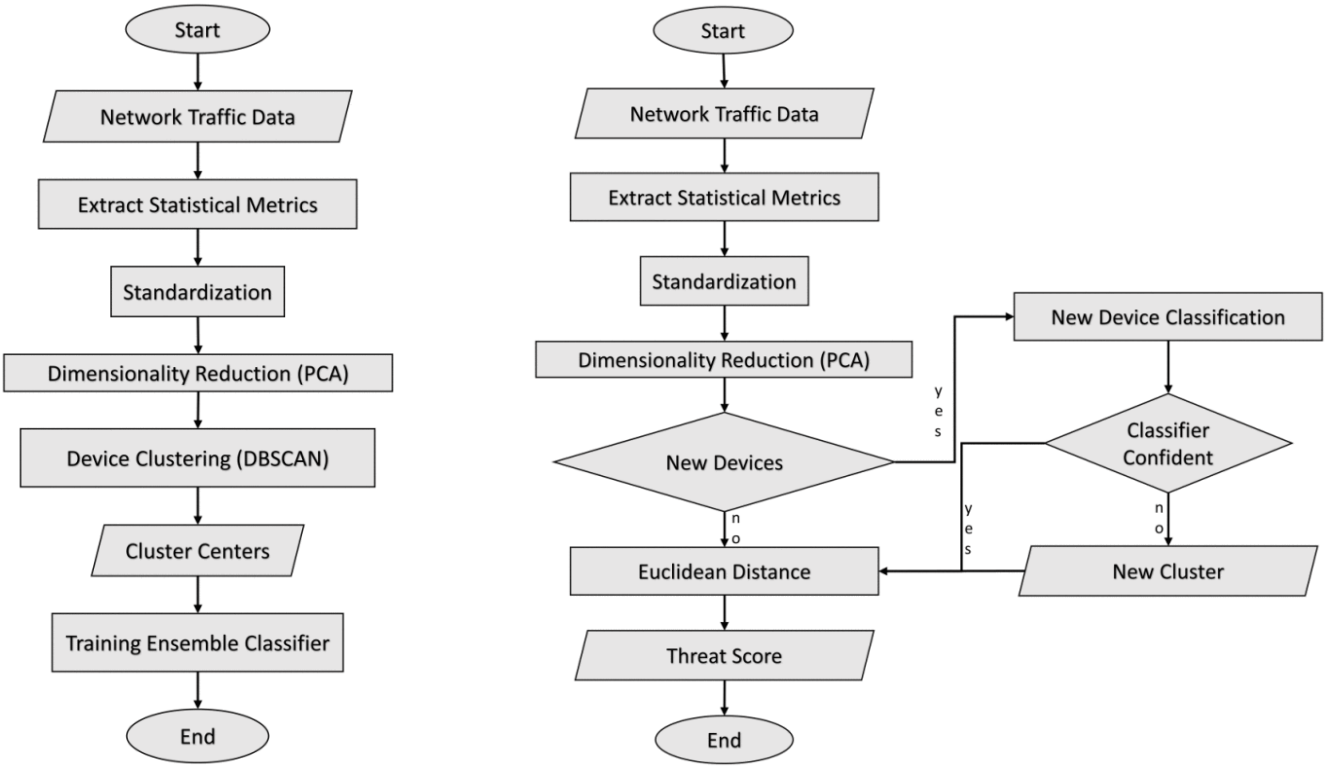


Figure 5: Proposed model by [5]

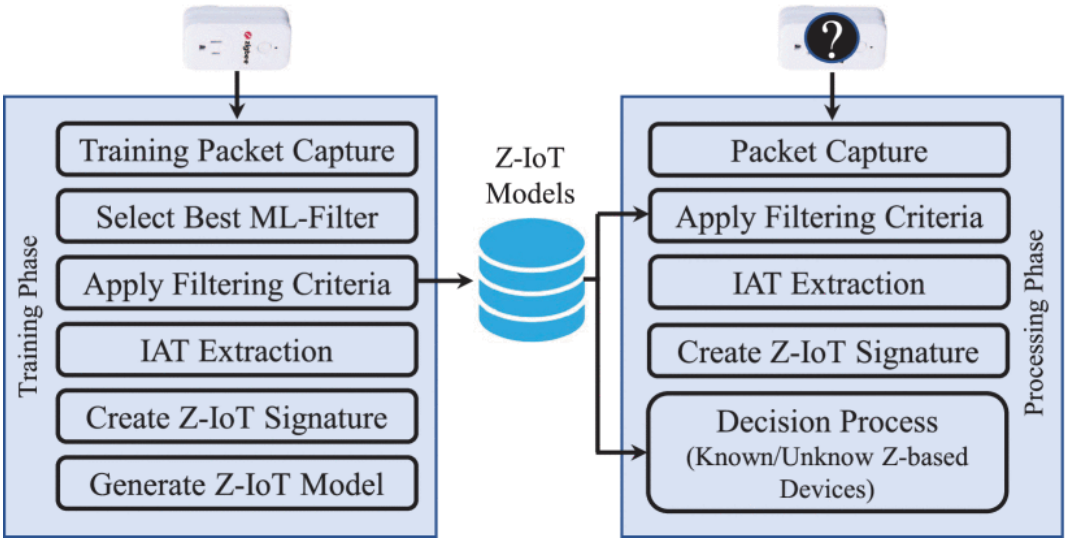


Figure 6: Proposed model by [6]

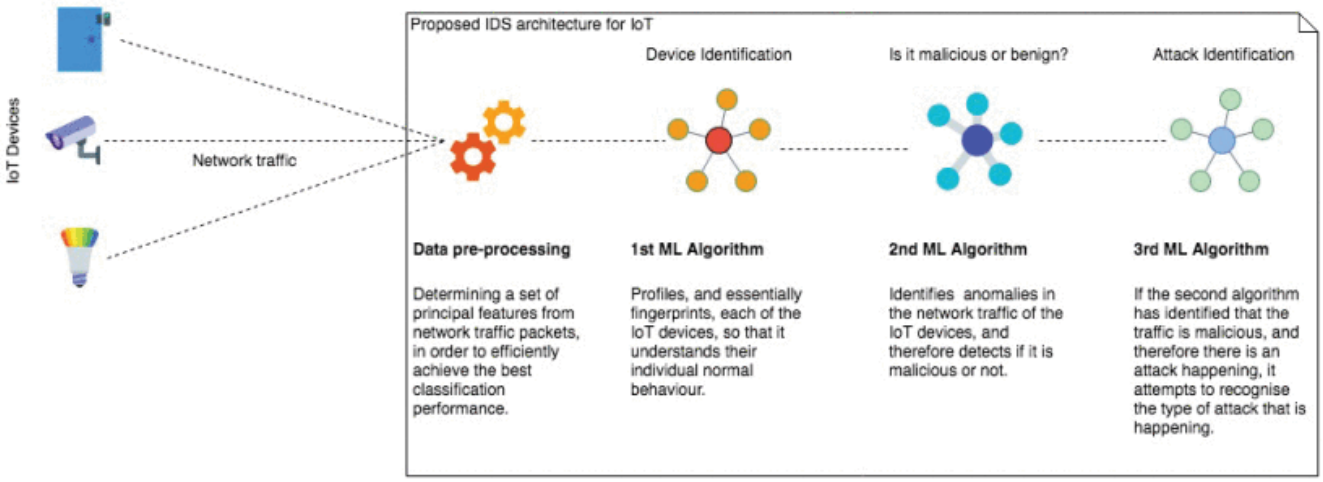


Figure 7: Proposed model by [7]

In [7], authors address the challenge of detecting and classifying network-based cyber-attacks on smart home IoT devices. The identified problem is the limited focus of existing systems on detecting a narrow set of attacks, which makes it challenging to identify complex combinations of attack behavior, particularly multistage attacks. To address this, the paper proposes a three-layer architecture for a lightweight, standalone intrusion detection system (IDS) tailored to IoT devices within a smart home network. The proposed solution, depicted in Fig. 7, involves the development of an IDS that can automatically distinguish between IoT devices on the network, identify malicious or benign network activity, and detect the type of attack deployed on each device. The experiment results demonstrate the effectiveness of the proposed architecture, with the system's core functions achieving high average F-measures, indicating successful device classification, detection of malicious network activity, and identification of attack types. The insights from the experiments highlight the importance of decision trees as the best algorithm for the task, the significance of IP and TCP flags as important features, and the potential impact of unsophisticated attacks on classification accuracy. However, the study's limitations include reduced accuracy when using unseen validation datasets and the need to evaluate more complex and sophisticated attacks further.

2.2.2 D/DoS Detection Works

This subsection delves into studies concerning detecting and mitigating Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks in smart home environments. With the proliferation of connected devices, the susceptibility to such attacks has increased, necessitating robust detection mechanisms. We examine various approaches proposed in the literature to identify and mitigate these disruptive threats effectively.

In [8], authors address the pressing issue of DDoS attacks launched by zombie IoT devices in smart home networks. These attacks pose a significant threat due to their stealth and apparent legitimacy, making them difficult to detect, especially at the application layer. To combat this, the authors propose ForChaos, a lightweight detection algorithm based on forecasting and chaos theory, presented in Fig. 8. The algorithm utilizes forecasting techniques to make short-term predictions and identify attacks by constructing Lyapunov exponents for every time-series interval. The proposed solution is evaluated through experiments in flooding and slow-rate DDoS attacks, demonstrating its effectiveness and robustness.

The experiment results presented in the paper provide valuable insights into the performance of the ForChaos

1: **Input:** Set of 8 Features f, Time Series t, α, Window Size w Traffic State A_{t-1}

2: **Output:** alert, not-alert

3: **for** $t = 1$`ton` **do**

4: **for** $f = 1$`to8` **do**

5: $F_t = aA_{t-1} + (1-a)F_{t-1}$

6: $e_t = |F_i - A_i|$

7: $e_{total_t} = e_{total_t} + \frac{1}{8}\sum_{i=1}^{n} e_i$

8: $\lambda_t = \frac{1}{t} \ln \left| \frac{\Delta e_{total_t}}{\Delta e_0} \right|$

9: *If* $\lambda_t > 0$

10: *return alert*

11: *else return not − alert*

Figure 8: Proposed model by [8]

algorithm. The authors report a detection rate of 93.75% in detecting DDoS attacks, showcasing the algorithm's ability to identify malicious behavior accurately. Furthermore, the study compares the ForChaos algorithm with related studies, highlighting its superiority in terms of the number of features used and the ability to detect both flooding and slow-rate attacks, which are not similar to each other. The comprehensive evaluation of the algorithm through machine learning algorithms such as Bayesian Networks, Support Vector Machines, and Neural Networks further strengthens the credibility of the proposed solution.

While the ForChaos algorithm demonstrates promising results, it is essential to acknowledge its limitations. The study recognizes that the algorithm's performance decreases against application layer DDoS attacks compared to attacks in lower layers, attributing this to the attacks' similarity to legitimate behavior and the exploitation of the time factor. Additionally, the authors highlight the need to reduce the complexity of the detection method, considering the resource constraints of IoT networks. Furthermore, the paper emphasizes the importance of protecting smart home networks from external and internal threats, indicating a potential limitation in the scope of the proposed solution. The study also discusses the impact of dataset size and composition on the performance of machine learning algorithms, shedding light on the challenges associated with constructing accurate and robust probabilities for detecting DDoS attacks.

In [9], authors address the problem of detecting anomalies in network traffic caused by distributed denial of service (DDoS) attacks, particularly focusing on the detection of illegitimate DDoS traffic generated by Internet of Things (IoT) devices. The authors highlight the emergence of the IoT concept, which has led to numerous terminal devices with a low level of implemented protection, making them increasingly susceptible to being used as a platform for generating DDoS traffic. As depicted in Fig. 9, the proposed solution is a novel approach for detecting DDoS traffic generated by IoT devices using a conceptual network anomaly detection model. This model is based on device classes dependent on individual device traffic characteristics, aiming to detect the deviation in the value of the traffic characteristics generated by the observed IoT device from the initially determined class or the legitimate traffic profile for that class.

The experiment results are not explicitly provided in the paper. However, the proposed DDoS traffic detection conceptual model based on a novel approach implies the class affiliation of IoT devices. The gained insights from the research include identifying differences in the characteristics of the traffic generated by IoT devices and traffic generated by Human Type Communication (HTC) devices and the limitations of existing methods in detecting

this specific form of DDoS traffic. The limitations of the research include the lack of explicit consideration of the experiment results and the specific challenges encountered during the implementation of the proposed conceptual network anomaly detection model.

The study by [10] addresses the problem of countering multi-vector DDoS attacks in IoT networks. The proposed solution, the OTI-IoT framework, introduces a novel "Prevent-then-Detect" methodology, integrating Intrusion Prevention Systems (IPS) and Intrusion Detection Systems (IDS) to achieve Operational Threat Intelligence. The framework, illustrated in Fig. 10, utilizes a consortium Blockchain network for IPS and deep learning-based IDS models, demonstrating a comprehensive approach to mitigating multi-vector DDoS attacks in IoT networks. The authors have identified the limited research on multi-vector DDoS attacks, synchronization overhead in IoT environments, and the lack of threat intelligence frameworks for identifying the source of such attacks as significant limitations in the existing literature.

The authors employed the CICDDoS2019 dataset, which comprises benign, portmap, NetBIOS, MSSQL, UDP flood, and SYN flood labels. The dataset was preprocessed to enhance efficiency, consistency, and accuracy, which is crucial for training and testing the IDS models. Data merging and cleaning techniques were applied to integrate information from multiple sources and handle missing and duplicated data, ensuring the quality and reliability of the dataset.

The experiment results demonstrate the effectiveness of the proposed OTI-IoT framework in achieving lower attack detection time, minimized block validation time, and higher attack prevention rates compared to state-of-the-art techniques. The insights gained from the experiment results and the proposed framework underscore the significance of amalgamating IPS and IDS modules using blockchain technology and deep learning to achieve operational threat intelligence in IoT networks. However, the authors acknowledge that the proposed framework does not address the intensity of the detected attacks, indicating a potential area for future research and development. The limitations identified in the paper include the excessive overhead of Blockchain-based CTI frameworks, the difficulty in classifying vast amounts of unstructured IoT traffic, and the lack of threat intelligence frameworks for identifying the source of multi-vector DDoS attacks in IoT networks.

In [11], authors address the growing concern of distributed denial of service (DDoS) attacks originating from insecure consumer Internet of Things (IoT) devices. The identified problem revolves around the vulnerability of IoT devices, which botnets like Mirai have exploited to launch DDoS attacks on critical Internet infrastructure. To tackle this issue, the paper proposes a solution, illustrated in Fig. 11, that involves leveraging IoT-specific network behaviors to inform feature selection for high-accuracy DDoS detection in IoT network traffic using machine learning algorithms. The proposed solution aims to enable home gateway routers or other network middleboxes to automatically detect local IoT device sources of DDoS attacks using low-cost machine learning algorithms and flow-based, protocol-agnostic traffic data.

The experiment results demonstrate the effectiveness of the proposed solution, with classifiers successfully identifying attack traffic with high accuracy, particularly through the use of random forest, K-nearest neighbors, and neural net classifiers. The insights gained from the experiment highlight the potential of deep learning classifiers and the importance of incorporating IoT-specific features for anomaly detection. However, the paper also acknowledges limitations, such as the lack of public datasets of consumer IoT attack traffic and the need for further research to evaluate IoT DDoS detection in more real-world settings. Additionally, the paper emphasizes the importance of replicating the results with normal traffic from additional IoT devices and with attack traffic recorded from a real DDoS attack to enhance the external validity of the proposed solution.

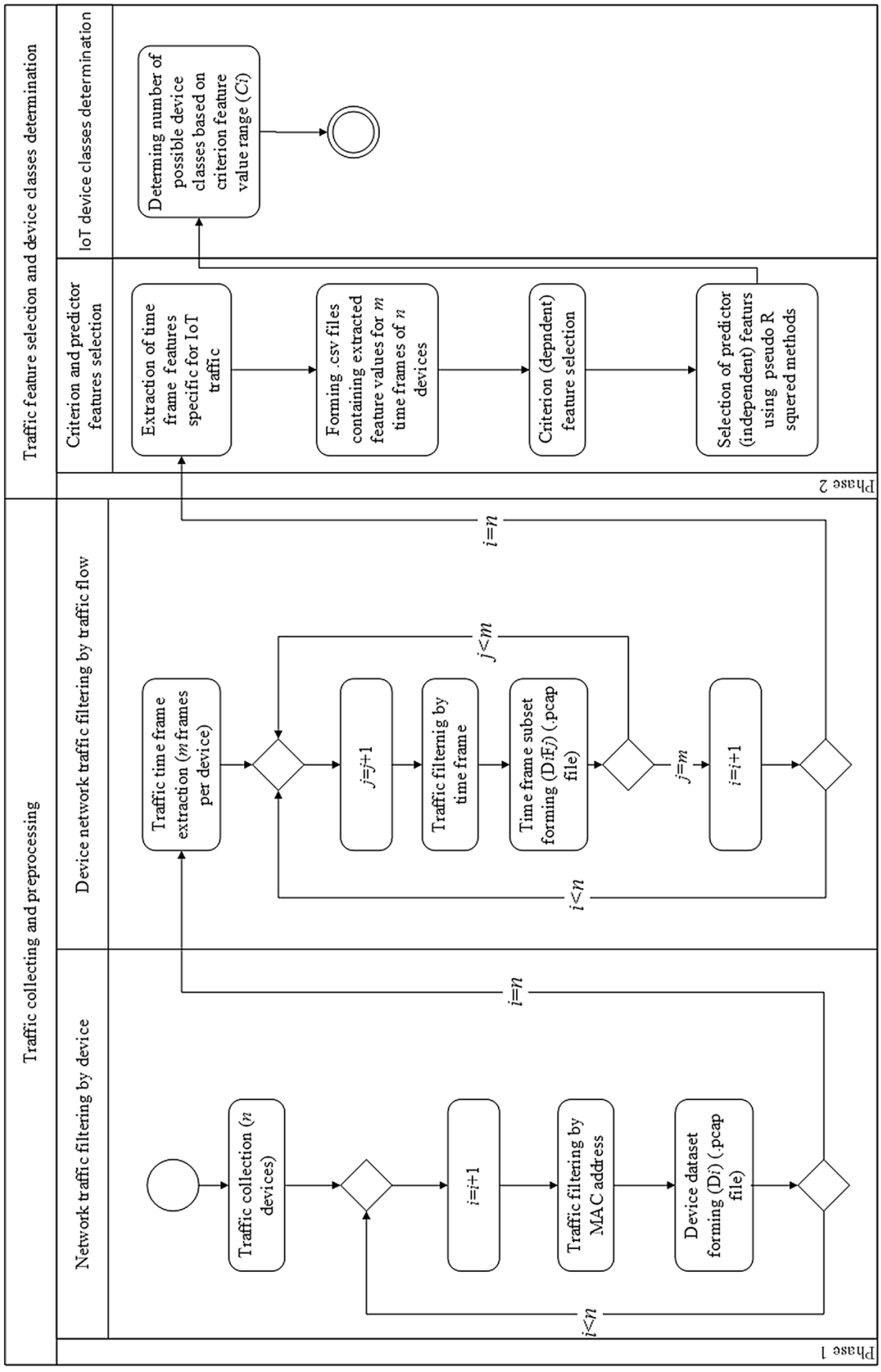


Figure 9: Proposed model by [9]

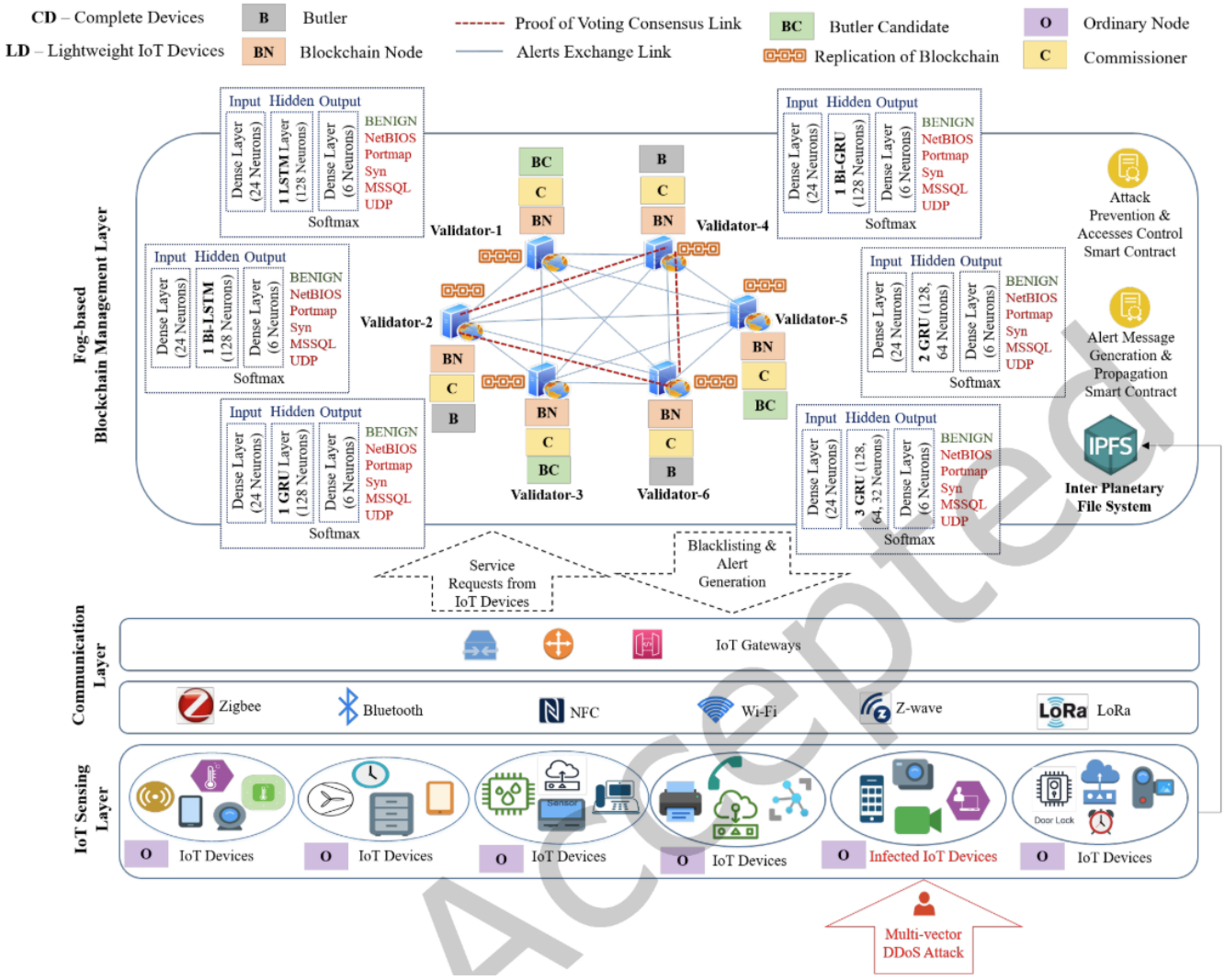


Figure 10: Proposed model by [10]

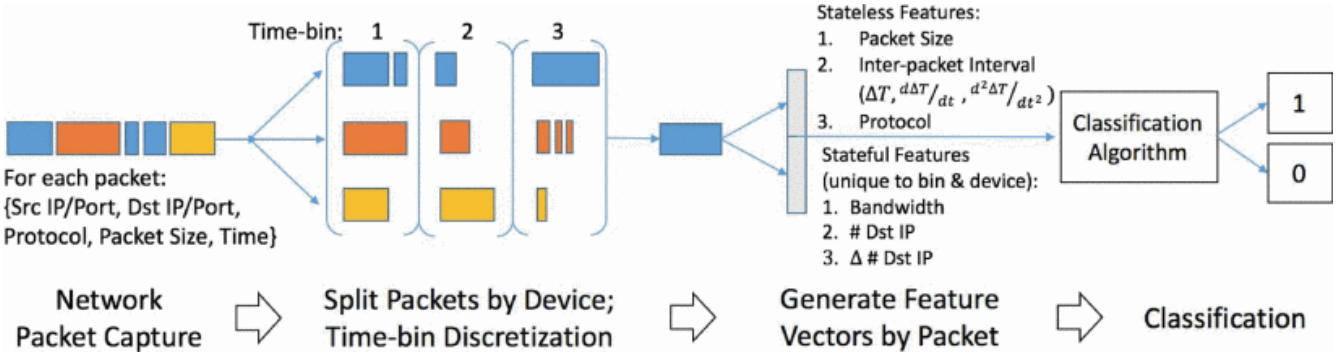


Figure 11: Proposed model by [11]

In [66], authors address the critical issue of security in smart home systems, which has often been overlooked despite the increasing reliance on Internet of Things (IoT) devices in households. The authors highlight the vulnerability of IoT-based systems and the need for a robust security framework to ensure the privacy and safety of residential infrastructure. The proposed solution involves the development of a machine learning (ML)-based model to detect Distributed Denial of Service (DDoS) attacks, a common threat to smart home systems. The authors engineered features based on the unique characteristics of IoT traffic to enable ML algorithms to classify DDoS traffic from normal/benign traffic accurately.

The experiment results presented in the paper demonstrate the effectiveness of the ML-based model in distinguishing between regular IoT traffic and DDoS attack traffic. The authors conducted comprehensive testing using various ML algorithms, including K-nearest neighbors, Support Vector Machines, Decision Trees, Random Forests, and Multi-Layer Perceptron. The results showed that simple ML algorithms with lightweight features could accurately classify regular IoT traffic from DDoS attack traffic. Additionally, the paper provides insights into the importance of feature engineering and the potential of ML algorithms in enhancing the security of smart home networks.

However, it is important to acknowledge the limitations of the study. The paper does not delve into the real-world deployment of the ML-based model for DDoS detection in smart homes. The authors also highlight the need for future work to extend the study to detect IoT DDoS in more realistic settings, such as running the Mirai code within a controlled network domain. Furthermore, the impact of the size of the dataset on classification accuracy and the potential application of deep learning in securing smart homes are identified as areas for future research.

In [12], authors address the escalating threat of denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks on Internet of Things (IoT) networks. The authors highlight the limitations of existing supervised learning models in detecting unknown attacks, which can have severe consequences. To address this, the paper proposes a novel approach, represented in Fig. 12 that combines a soft-ordering convolutional neural network (SOCNN) model with local outlier factor (LOF) and isolation-based anomaly detection using nearest-neighbor ensembles (iNNE) models. This hybrid model leverages supervised and unsupervised learning methods to detect unknown attacks accurately, outperforming state-of-the-art competitors. The proposed model also demonstrates resilience against adversarial attacks, such as the fast gradient sign method (FGSM) and Carlini Wagner (CW) adversarial attacks, thus enhancing IoT network security against DoS/DDoS attacks in unknown attack scenarios.

The experiment results presented in the paper showcase the effectiveness of the proposed hybrid learning model in detecting unknown DoS/DDoS attacks. The model achieved high accuracy in detecting unknown attacks across three benchmark datasets, with an average F1-score of 98.94%, 91.68%, and 96.07% on BoT-IoT, CIC-IDS-2017, and CIC-IDS-2018 datasets, respectively. Furthermore, the model's resilience against adversarial attacks was demonstrated through experimental testing, highlighting its potential to counter sophisticated attack techniques.

The insights gained from the research emphasize the significance of leveraging a hybrid learning model that combines supervised and unsupervised learning methods to detect unknown DoS/DDoS attacks effectively. The proposed model's ability to achieve high accuracy in detecting unknown attacks and resisting adversarial attacks underscores its potential to enhance the security of IoT networks. The research also identifies the need for further experiments to evaluate the model's performance in different contexts, such as resistance to poisoning attacks or countering GAN-based attacks and subtypes of adversarial attacks. The paper has some possibilities for future works, including developing frameworks that support network intrusion detection system (NIDS) testing in real-world network environments and expanding the study to build a hybrid NIDS that can leverage unsupervised and supervised learning strengths.

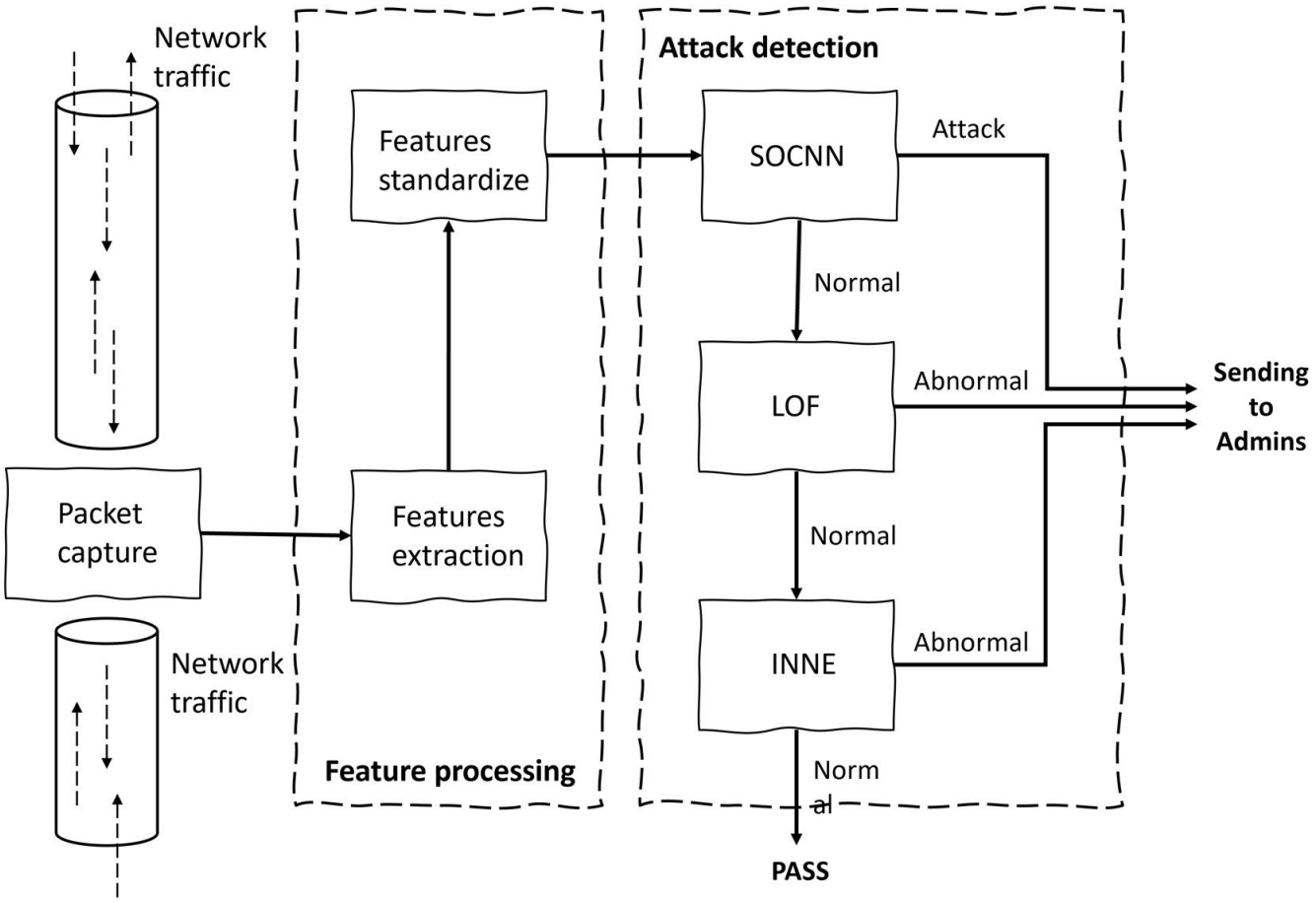


Figure 12: Proposed model by [12]

In [13], authors address the critical issue of emerging stealthy Distributed Denial of Service (DDoS) attacks in Internet of Things (IoT) networks. The identified problem revolves around the susceptibility of IoT devices to being compromised and participating in Mongolian DDoS attacks, characterized by their widely distributed nature and small attack size from each source. The proposed solution, depicted in Fig. 13, is a novel anomaly-based Intrusion Detection System (IDS) capable of timely detecting and mitigating these emerging DDoS attacks, even with very low attack size per source. The proposed IDS is thoroughly analyzed regarding time and space complexity, asymptotic optimality, and performance evaluation using testbed implementation, the N-BaIoT dataset, and simulations.

The experiment results presented in the paper provide valuable insights into the effectiveness of the proposed IDS in detecting and mitigating stealthy DDoS attacks. The performance evaluation encompasses a comprehensive analysis of the proposed technique using a testbed implementation, the N-BaIoT dataset, and simulations, demonstrating the capability of the IDS to detect and mitigate stealthy DDoS attacks with minimal detection delay and false alarm rates. However, the paper also acknowledges certain limitations that need to be addressed to enhance the robustness of the proposed system. These limitations include assumptions regarding the nominal behavior of devices, the need for periodic updates of the IDS in real system implementations, and the importance of feature extraction in accurately representing the characteristics of a real network.

2.2.3 Intrusion Detection Works

Intrusion detection is paramount for safeguarding smart home networks against unauthorized access and malicious activities. This subsection discusses research endeavors to develop intrusion detection systems tailored for smart

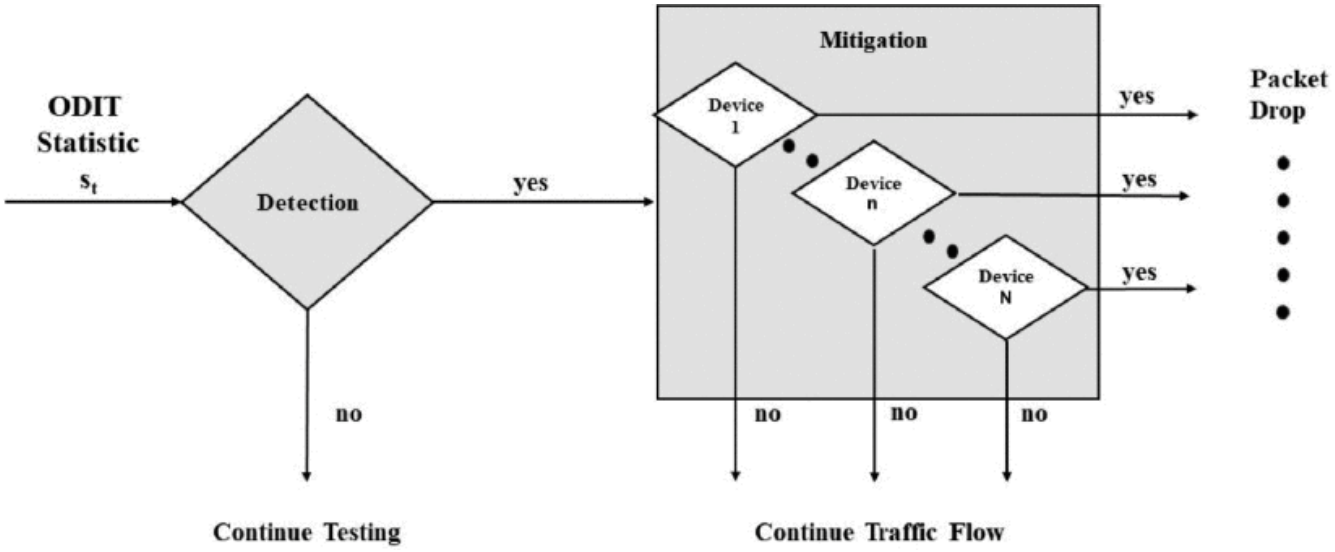


Figure 13: Proposed model by [13]

home environments. We review methodologies and algorithms for detecting and responding to anomalous behavior indicative of potential intrusions.

In the study by [14], the authors address the problem of traditional intrusion detection methods struggling to effectively process and extract meaningful features from dynamic and complex IoT network data. The authors highlight the challenges posed by the high dimensionality and complex relationships of IoT network data, which make intrusion data processing and feature extraction difficult for traditional detection methods. To address this problem, the proposed solution revolves around a novel GCN-Ensemble fusion model, which leverages graph convolution neural network (GCN) and deep learning models to represent and classify attacks in IDS datasets effectively. The proposed methodology, depicted in Fig. 14, involves three stages: data pre-processing, feature extraction, and classification, with a focus on capturing semantic features in the IDS dataset through GCN for feature learning and addressing the issue of data imbalance to improve accuracy.

In terms of the datasets used, the paper evaluates the proposed model on four datasets: BoT-IoT, ToN-IoT, CIC-IDS 2018, and NF UQ NIDS. These datasets are selected for evaluation based on their relevance to IoT environments and the presence of diverse attacks on real-world IoT traffic. The authors emphasize the importance of evaluating the proposed model on widely used open-access datasets to demonstrate its effectiveness in real-world scenarios.

The experiment results reveal promising insights into the performance of the proposed model. The authors compare the proposed model with state-of-the-art models based on accuracy, precision, recall, and confusion matrix classification metrics. The results indicate that the proposed model consistently outperforms other models, including deep learning or ensemble methods, showcasing its effectiveness and superiority. The gained insights from the experiment results emphasize the potential of the GCN-Ensemble fusion model to augment IoT network security, with observed accuracy improvements over baseline and state-of-the-art ensemble models.

However, the paper also acknowledges certain limitations, including the absence of exploration across diverse datasets in some previous studies and the need to explore varying datasets further to address data imbalance issues and improve accuracy. Additionally, the authors acknowledge the computational complexity and memory requirements associated with certain approaches, highlighting the need for future research to address these challenges.

[15] discusses designing and evaluating a Modified Binary IDS (MBIDS) for detecting attacks on Z-Wave networks. The identified problem is the need to enhance the system's ability to detect and prevent attacks while reducing the cost of employing the system. The proposed solution, illustrated in Fig. 15, involves conducting

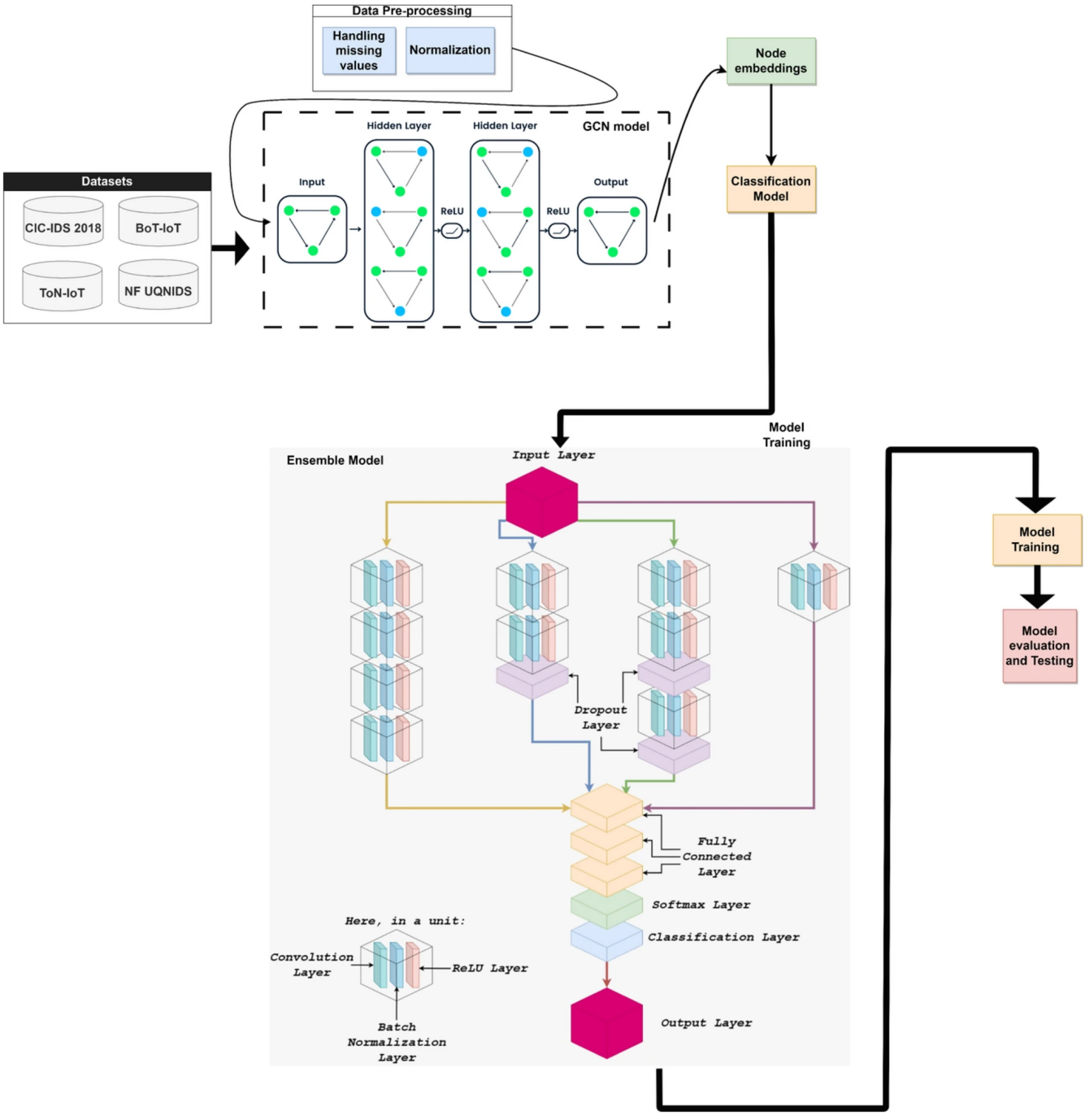


Figure 14: Proposed model by [14]

two experiments to evaluate the effectiveness of the MBIDS in detecting various types of attacks and to assess the impact of enhancements made to the system.

The experiments involve crafting and wirelessly transmitting packets to the Z-Wave network and evaluating the MBIDS' detection capabilities. The first experiment scenario aims to compare pre and post-enhancement results and includes seven tests totaling 2100 trials. On the other hand, the second experiment scenario evaluates the enhancement strategies and includes 2100 trials. The results of the experiments show that the MBIDS has a high mean detection rate for various types of attacks, such as manipulated packet injection, invalid payloads, and other misuse cases. The system effectively detects injected packets with both unsupported Command Classes (CmdCls) and invalid Commands (Cmd), with mean detection rates of 100%

Insights gained from the experiments include the effectiveness of the MBIDS in detecting and logging misuse cases and the impact of system enhancements on detection accuracy. Additionally, the paper highlights the challenges of accounting for all known-good Header types in Z-Wave transmissions and the limitations of fully integrating certain checks, which may reduce the system's overall effectiveness. The study's limitations include the inability to account for all possible combinations of certain Command Classes due to their variability and the infeasibility of integrating certain checks, which may impact the system's performance.

In [16], the authors address the issue of enhancing trust and informed decision-making in intrusion detection systems (IDS) within the Industrial Internet of Things (IIoT). The identified problem revolves around the need for explainable AI (XAI) approaches to improve the interpretability and trust of machine learning (ML) models among users, security experts, and stakeholders, particularly in the context of imbalanced attack class samples. The proposed solution, illustrated in Fig. 16, involves the analysis and experimentation of state-of-the-art algorithms for attack detection, as well as the application of post-hoc XAI techniques, such as SHapley Additive exPlanations (SHAP) and Local Interpretable Model-Agnostic Explanations (LIME), to evaluate predictive algorithms and assess trust and concordance levels in IDS models. The study utilizes the WUSTLIIoT dataset and various classifiers to evaluate the effectiveness of the proposed XAI architecture, focusing on local explanations and feature relevance to gain insights into the data features crucial for model predictions. The experiment results provide a comprehensive analysis of the performance of the XAI techniques, shedding light on the decision and confidence impact ratios, mean vote for model inference, and trust, thereby enabling informed decisions for cybersecurity experts and stakeholders. However, the study also acknowledges limitations, particularly regarding the comprehensibility of the XAI techniques to non-technical users, which highlights the need for further research in this area.

In the paper by [17], the authors address the challenge of enhancing security within the Internet of Medical Things (IoMT) network, which is susceptible to significant security vulnerabilities due to its reliance on communication protocols. The proposed solution, a novel Meta-Intrusion Detection System (Meta-IDS) illustrated in Fig. 17, employs a meta-learning approach to enhance the detection of known and zero-day intrusions. This approach integrates signature-based and anomaly-based detection techniques, incorporating privacy-preserving methods for handling sensitive IoMT data. The authors evaluated the methodology using three publicly available datasets (WUSTL-EHMS-2020, IoTID20, and WUSTL-IIOT-2021), demonstrating remarkable accuracy rates for signature-based and anomaly-based detection and impressively low misclassification rates. Through comparative analysis with the state-of-the-art E-GraphSAGE model, the authors affirm the performance and reliability of the Meta-IDS, highlighting its significant promise in bolstering cybersecurity within the IoMT network.

The choice of datasets, including WUSTL-EHMS-2020, IoTID20, and WUSTL-IIOT-2021, reflects a comprehensive evaluation of the proposed Intrusion Detection System (IDS) across diverse scenarios, encompassing healthcare

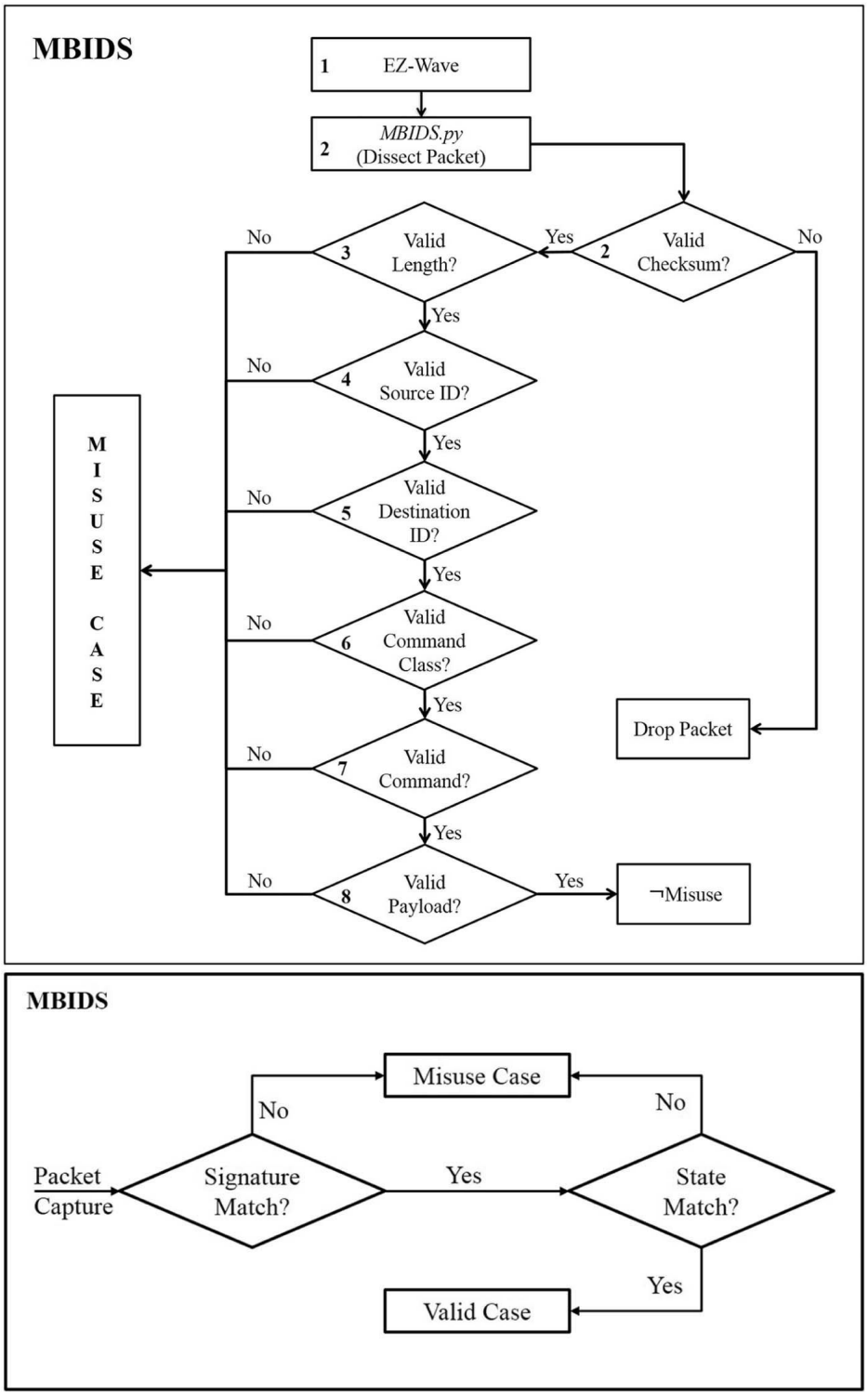


Figure 15: Proposed model by [15]

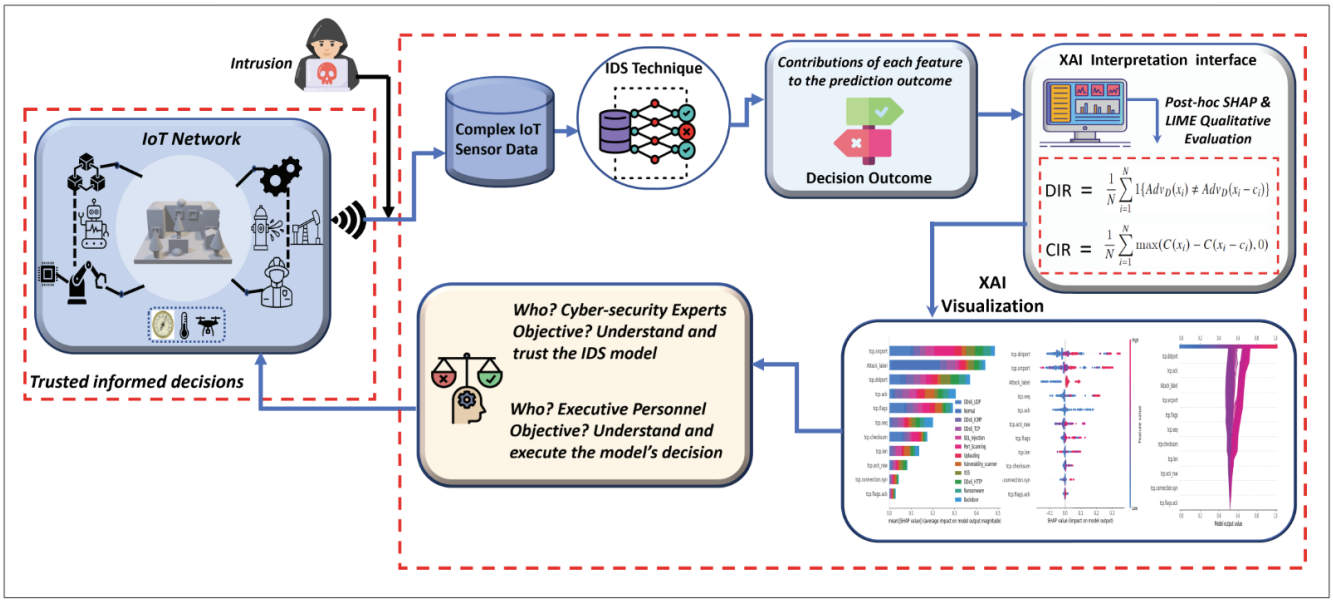


Figure 16: Proposed model by [16]

environments, IoT networks, and industrial IoT settings. The experiments and performance evaluations using these datasets yielded insights into the IDS's adaptability and effectiveness across varied domains and security requirements. The evaluation metrics included Accuracy, precision, recall, and F1-score, which were calculated to assess the model's efficacy.

Estimating known intrusions' performance involved hold-out and cross-validation methods, ensuring generalizability and mitigating overfitting risks. The model evaluation and dataset split procedures included allocating seventy percent of the data for training and reserving the remaining thirty percent for testing. Additionally, the model underwent evaluation on distinct subsets of the training dataset through 10-fold cross-validation, addressing concerns related to concept drift and overfitting. The paper also discusses the datasets' limitations, highlighting that some datasets contain only network traffic data, which may not be suitable for attacks using IoMT. Furthermore, the authors acknowledge the need to improve the results obtained from certain datasets, indicating areas for future research and development.

In [18], authors address the increasing security risks of deploying smart devices in home environments. The identified problem pertains to the vulnerability of smart devices to suspicious or malicious activities, necessitating the development of an intrusion detection and mitigation framework. The proposed solution, illustrated in Fig. 18 IoT-IDM, offers network-level protection by monitoring the network activities of smart devices and blocking intruders upon detection of an intrusion. The framework's modular design allows for the customization of machine learning techniques for detection based on learned signature patterns of known attacks. Leveraging software-defined networking technology and OpenFlow communication protocol, the authors developed a prototype of IoT-IDM. They conducted a case study using a smart light bulb to demonstrate its applicability and efficiency.

The experiment results demonstrate the feasibility and effectiveness of IoT-IDM, revealing limited computation and communication overhead. The insights gained from the experiment highlight the potential of the framework to provide network-level protection for smart home IoT devices, particularly in detecting and preventing unauthorized access and other security threats. However, the paper has limitations, such as the framework's applicability to specific technology designs and its focus on the design and development phases rather than the operational phase of attack identification and prevention. Additionally, due to their heterogeneous architectures, the proposed risk analysis may not be universally applicable to all smart home automation systems. Despite these limitations, the

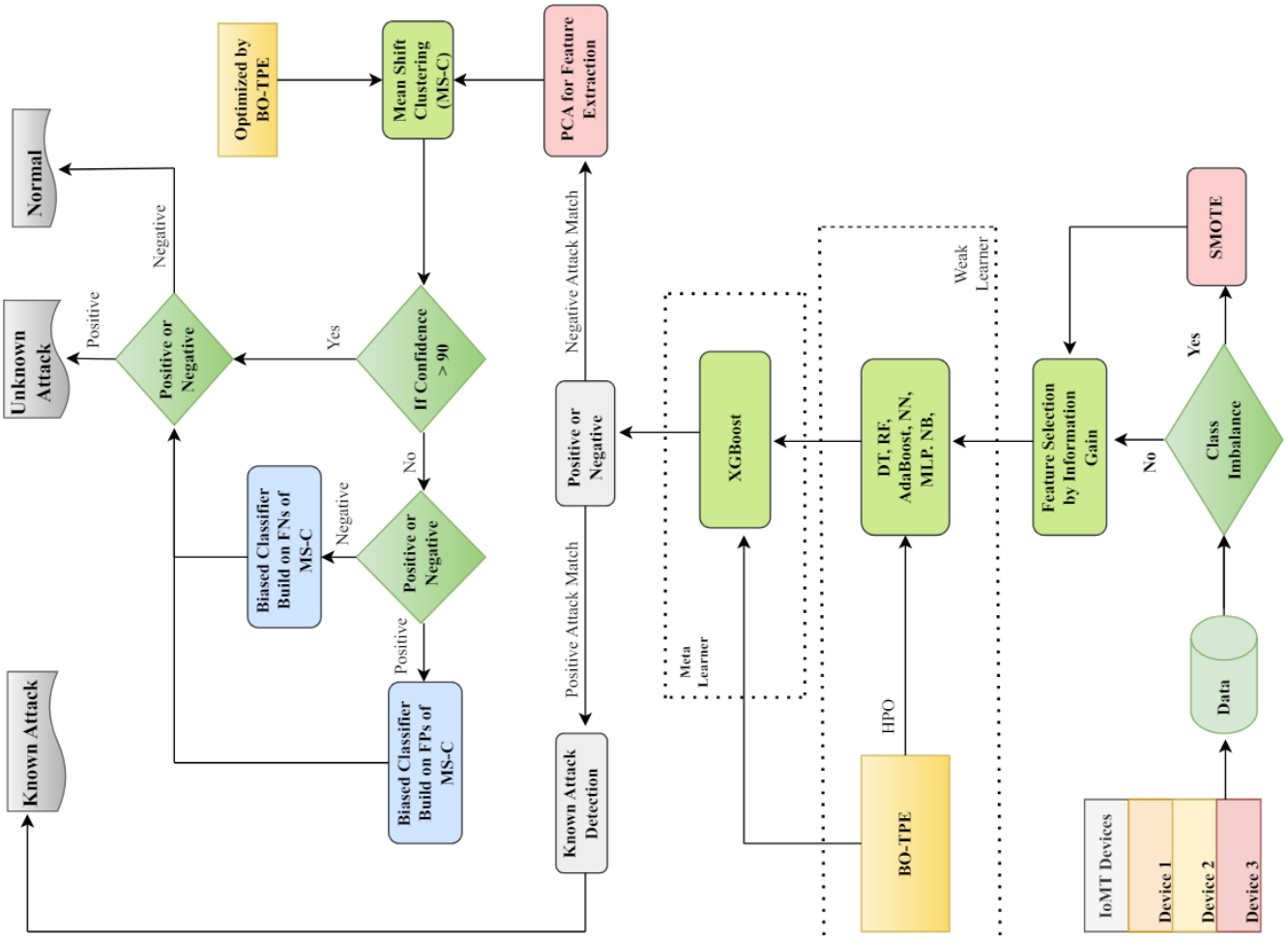


Figure 17: Proposed model by [17]

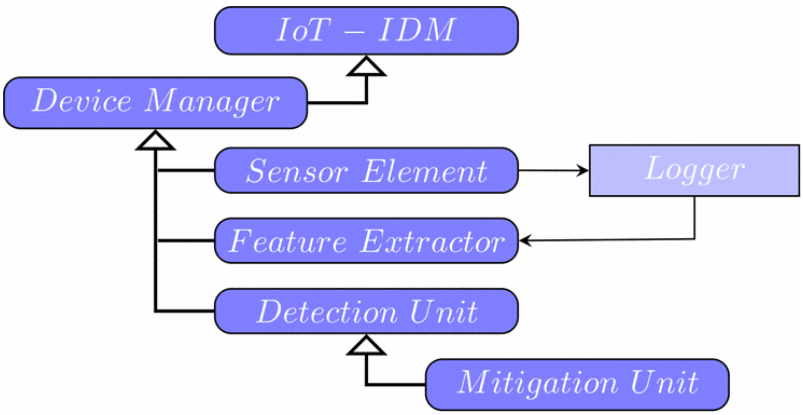


Figure 18: Proposed model by [18]

paper provides valuable insights into developing an intrusion detection and mitigation framework tailored to the unique security challenges of smart home IoT environments.

In [19], authors address the critical security challenges faced by smart cities and indoor wireless networks in the context of the rapidly growing IoT applications. The identified problem revolves around the vulnerabilities in IoT devices, which have increased the risk of injection attacks, such as ARP, ChopChop, and Fragmentation attacks. These attacks pose a significant threat to the security and privacy of billions of IoT smart devices, making intrusion detection a crucial aspect of mitigating these risks. The proposed solution, presented in Fig. 19 involves the development of an intrusion detection method that utilizes machine learning techniques, specifically employing two feature selection techniques, constant removal and recursive feature elimination, to detect injection attacks in IoT applications. The method was evaluated using the public dataset, AWID, and the results demonstrated that the decision tree classifier could detect injection attacks with an accuracy of 99% and an F1-Score of 90% using only eight features, which were selected using the proposed feature selection method.

However, it is important to acknowledge the limitations of the proposed method. One of the limitations is the need for further investigation into the convergence time and iteration of the proposed feature selection technique and its impact on detection accuracy. Additionally, while the proposed method demonstrated high accuracy in detecting injection attacks, it is essential to consider the scalability and adaptability of the method to diverse IoT environments and evolving attack vectors. Moreover, the practical implementation of the proposed intrusion detection method in real-world IoT applications, especially in large-scale smart city deployments, requires careful consideration of computational resources, energy efficiency, and real-time processing capabilities. These limitations highlight the need for continued research and development to enhance the robustness and applicability of intrusion detection methods for smart IoT environments.

In the work by [20], the authors address the prevalent issue of data imbalance and associated collection costs in Network Intrusion Detection Systems (NIDS). The authors propose a solution, illustrated in Fig. 20, that explores the potential of using 100% synthetic data generated via Generative Adversarial Networks (GAN) for training ML models in NIDS, thereby reducing the dependency on real-world data significantly. The study utilizes three datasets, namely UNSW-NB15, NSL-KDD, and BoT-IoT, to evaluate the performance of the proposed approach. The experimental results demonstrate high performance, with an accuracy of 90%, precision of 91%, recall of 90%, and an F1 score of 89% for the UNSW-NB15 dataset, 84% accuracy, 85% precision, 84% recall, and 84% F1 score for the NSL-KDD dataset, and perfect scores of 100% across all metrics for the BoT-IoT dataset. The insights

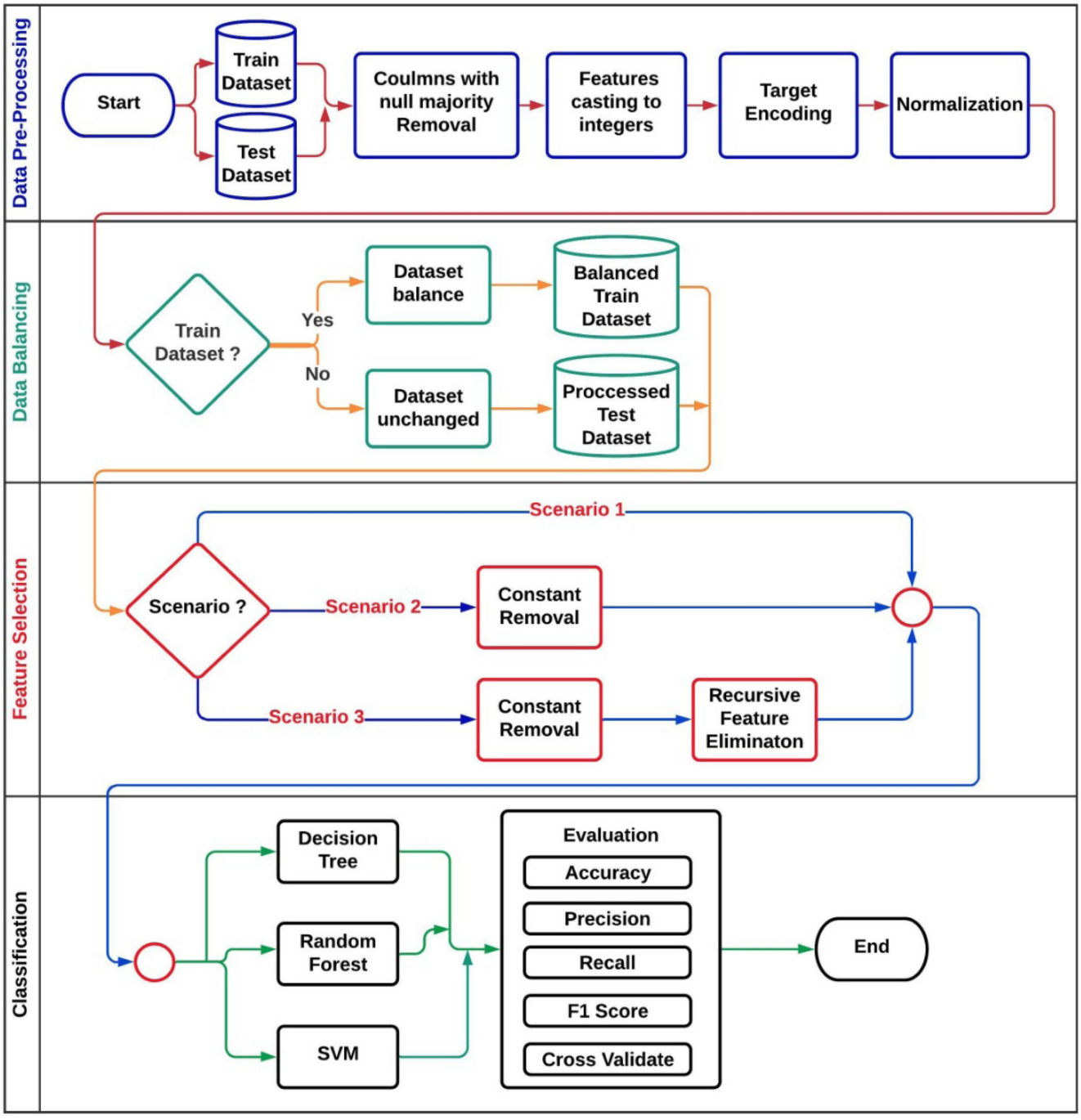


Figure 19: Proposed model by [19]

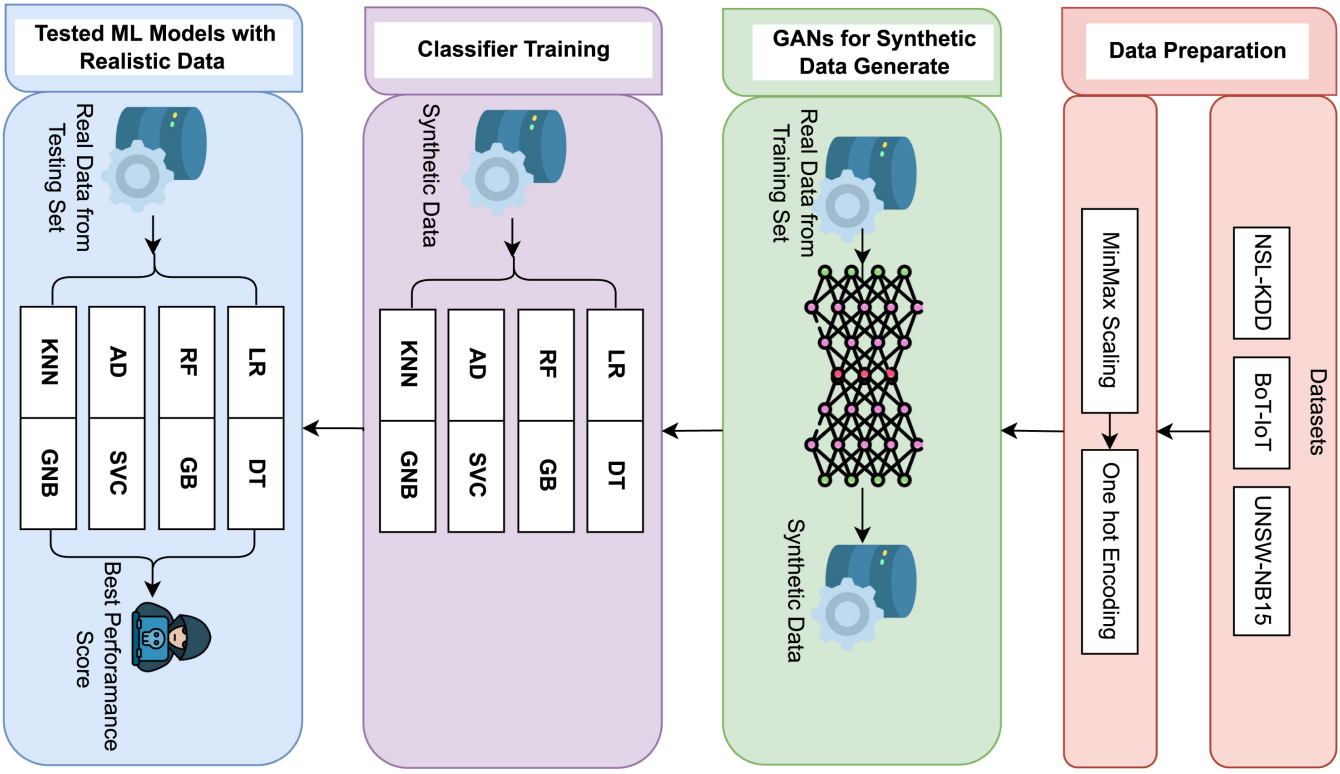


Figure 20: Proposed model by [20]

gained from the study indicate that the proposed GAN-based framework successfully replicates real-world network intrusion detection data, showing new opportunities for using generative data in cybersecurity. However, the study also acknowledges limitations, such as the absence of the Mean Time to Detect (MTTD) metric and the need for further exploration of adaptive GAN frameworks to address emerging attack techniques

In [21], authors address the issue of network security in the digital world, emphasizing the significance of Intrusion Detection Systems (IDS) in detecting and preventing network intrusions. The identified problem is the need for more efficient and effective intrusion detection systems, necessitating careful consideration in the present digital world. To address this problem, the authors propose a feature selection and majority vote-based intrusion detection method, successfully reducing the number of available features from 41 to 12 while maintaining detection accuracy. The proposed method, depicted in Fig. 21, was tested using the NSL-KDD benchmark dataset, and the experiment findings revealed an accuracy rate of 96.49%, indicating the method's worth in selecting the most relevant and instructive features for the classification operation. The insights gained from the experiment results contribute significantly to developing more efficient and effective intrusion detection systems by emphasizing the role of feature selection in improving classification model performance in detecting network security threats.

This study does not extensively discuss the potential challenges and drawbacks of the feature selection and majority vote-based intrusion detection method. Additionally, the research findings are based on the experiment conducted using the NSL-KDD benchmark dataset, which may limit the generalizability of the results to other datasets or real-world scenarios. Furthermore, the paper does not extensively delve into the computational complexity and scalability of the proposed method, which are crucial considerations in the real-world deployment of intrusion detection systems. Therefore, while the proposed method shows promise in enhancing cybersecurity, further research, and experimentation are necessary to address the identified limitations and validate the method's effectiveness in diverse network security environments.

In [22], authors address the security concerns arising from the proliferation of Internet of Things (IoT) devices in smart homes. The authors highlight the vulnerabilities introduced by smart home gateways, which serve as a

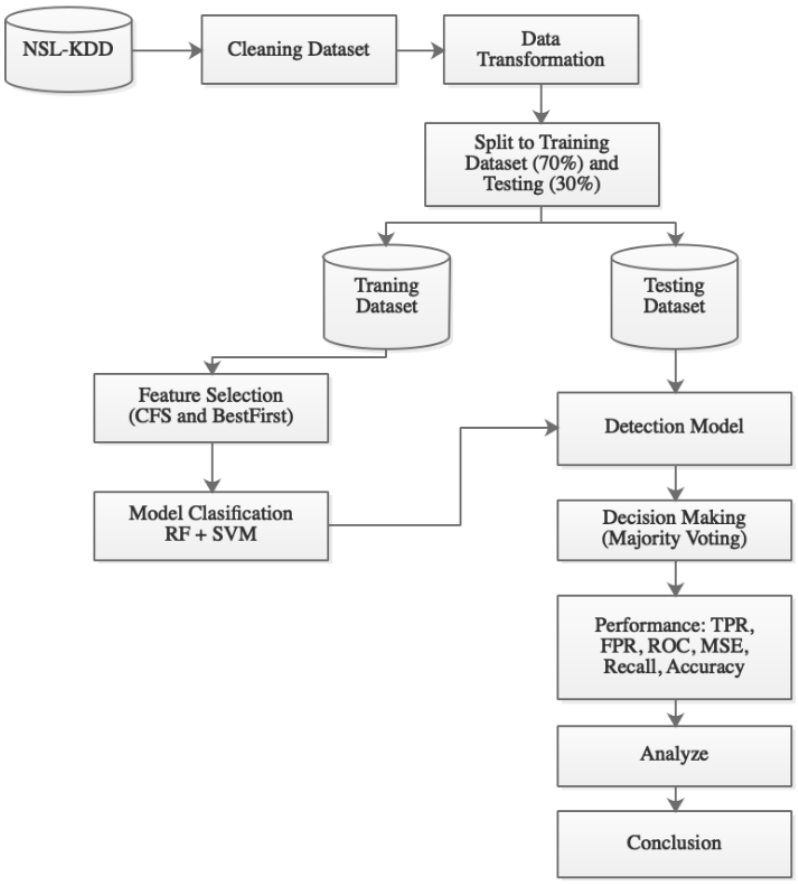


Figure 21: Proposed model by [21]

central point of communication between IoT devices and can potentially create a backdoor for hackers. To mitigate these security risks, the paper proposes an intrusion detection system (IDS), presented in Fig. 22, that utilizes a bidirectional long short-term memory (BiLSTM) and convolutional neural network (CNN) hybrid model to detect anomalies in smart home network traffic. The proposed model is designed to preserve learned information through time using the BiLSTM recurrent behavior, while the CNN is employed to extract data features. The study's experimental results demonstrate the effectiveness of the proposed BiLSTM-CNN hybrid model, achieving high accuracy, precision, recall, and F1 score. Insights gained from the experiments underscore the potential of the proposed model to be implemented in any smart home network gateway, offering improved security and the ability to detect various attack categories, including Mirai, DoS, MITM ARP, normal, and scan. However, the study acknowledges limitations, such as the need for further research to address the scalability and real-time applicability of the proposed model in large-scale smart home networks.

In [23], authors address the critical problem of securing smart homes against cyber threats, which have become increasingly prevalent due to the widespread adoption of IoT technology. The authors highlight the limitations of existing intrusion detection systems (IDSs) in smart homes, emphasizing the need for adaptive and self-configurable models to detect security breaches in dynamic smart home environments effectively. To tackle this challenge, the paper introduces MAGPIE, presented in Fig. 23, the first smart home IDS that leverages reinforcement learning to autonomously adjust its anomaly classification models based on changing smart home conditions, thereby achieving high accuracy by considering both cyber and physical data sources.

The proposed solution, MAGPIE, is experimentally evaluated to demonstrate its effectiveness in addressing the identified problem. The experiment results provide valuable insights into the prototype's performance, including

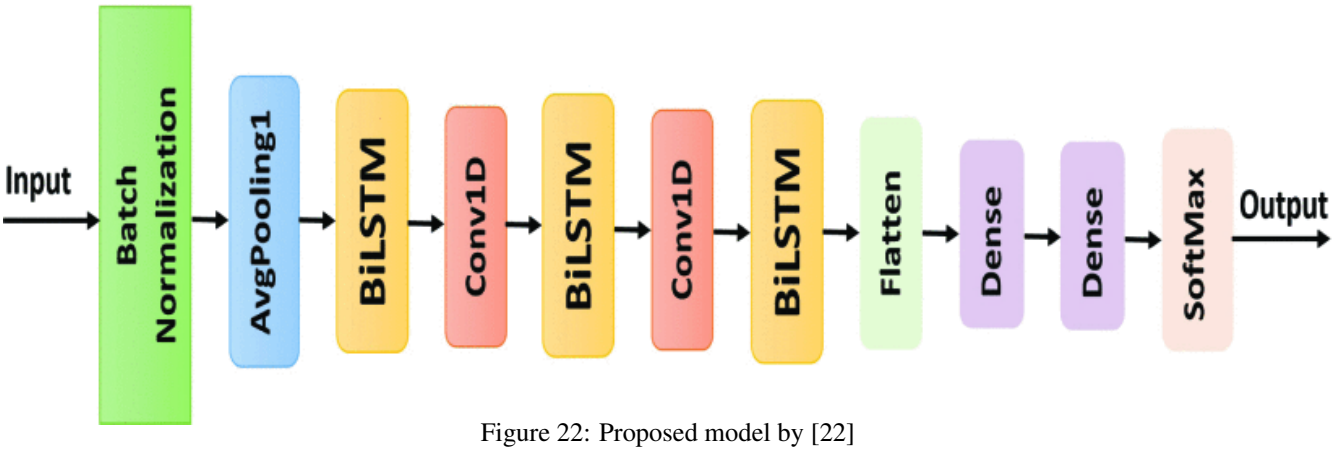


Figure 22: Proposed model by [22]

its ability to detect previously unseen attacks, adapt to changing conditions through reinforcement learning, and benefit from the use of both cyber and physical data sources. The experiment also sheds light on the impact of user presence in the smart home on threat detection performance and the latency associated with attack detection and end-to-end monitoring. Furthermore, the results highlight the significance of incorporating physical data sources in improving the IDS's performance, especially for attacks that are typically undetectable by systems monitoring only TCP/IP traffic.

While the experiment results provide valuable insights, the paper also acknowledges certain limitations. These limitations include the need for more expressive features to improve the detection of specific attacks, the influence of different settings on the reinforcement learning adaptation process, and the challenges associated with detecting attacks that blend in with the occupants' use of smart home devices. Moreover, the influence of different settings on the reinforcement learning adaptation process is highlighted, indicating the complexity of optimizing the system's performance under varying conditions.

In the paper [24], the authors address the challenges associated with developing an effective and scalable Intrusion Detection System (IDS) for the Internet of Things (IoT) in the face of dynamic and evolving cyber threats. The identified problem revolves around the need for a robust security framework to safeguard interconnected IoT systems in the wake of increasing cyberattacks. The proposed solution, shown in Fig. 24, introduces a horizontal federated learning (FL) model that combines Convolutional Neural Networks (CNN) and Bidirectional Long-Term Short Memory (BiLSTM) to enhance intrusion detection. This hybrid approach aims to overcome existing methods' limitations and improve intrusion detection's effectiveness in the context of FL for IoT.

The study utilizes two diverse and real-world traffic datasets, namely CICIDS2017 and Edge-IIoTset, to evaluate the proposed intrusion detection system rigorously. These datasets encompass a wide range of traffic, including benign and various malicious activities, ensuring a comprehensive assessment of the classifiers. The experiment results demonstrate the effectiveness of the proposed approach over centralized and federated deep learning-based systems. The insights gained from the experiments provide valuable information on the performance and efficacy of each classifier, offering a detailed comparative analysis of the proposed privacy-preserving DFL-based framework against traditional centralized learning approaches.

While the proposed approach presents several strengths, it is important to consider the limitations identified in the study. The limitations include communication latencies experienced during performance evaluation, the model's suitability for large-scale IoT networks, communication overhead, and an increased ratio of false positives. The model's training time is also considerably high, and end-to-end communication delays are increased. These limitations highlight areas for further improvement and optimization in developing intrusion detection systems for IoT.

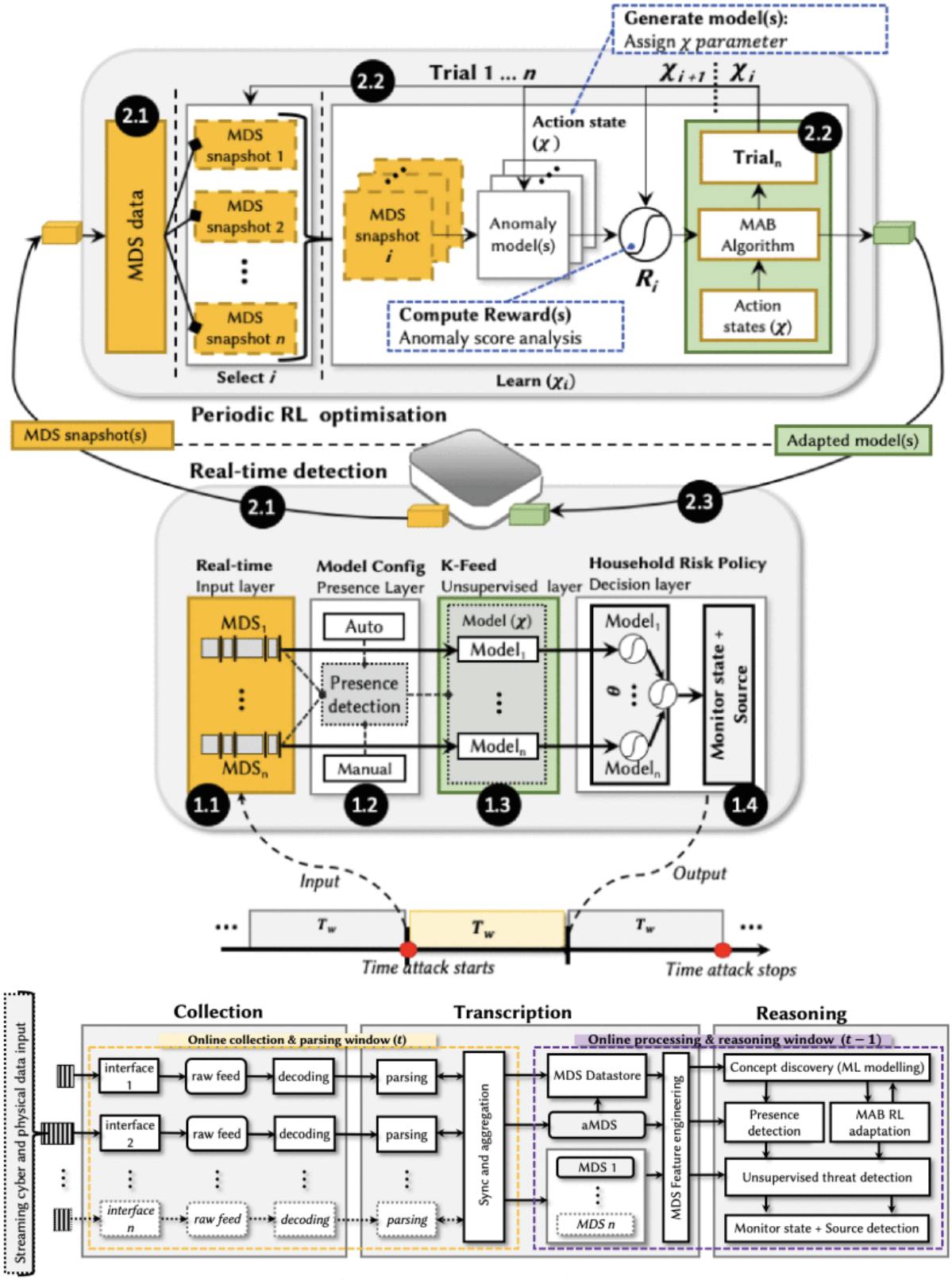


Figure 23: Proposed model by [23]

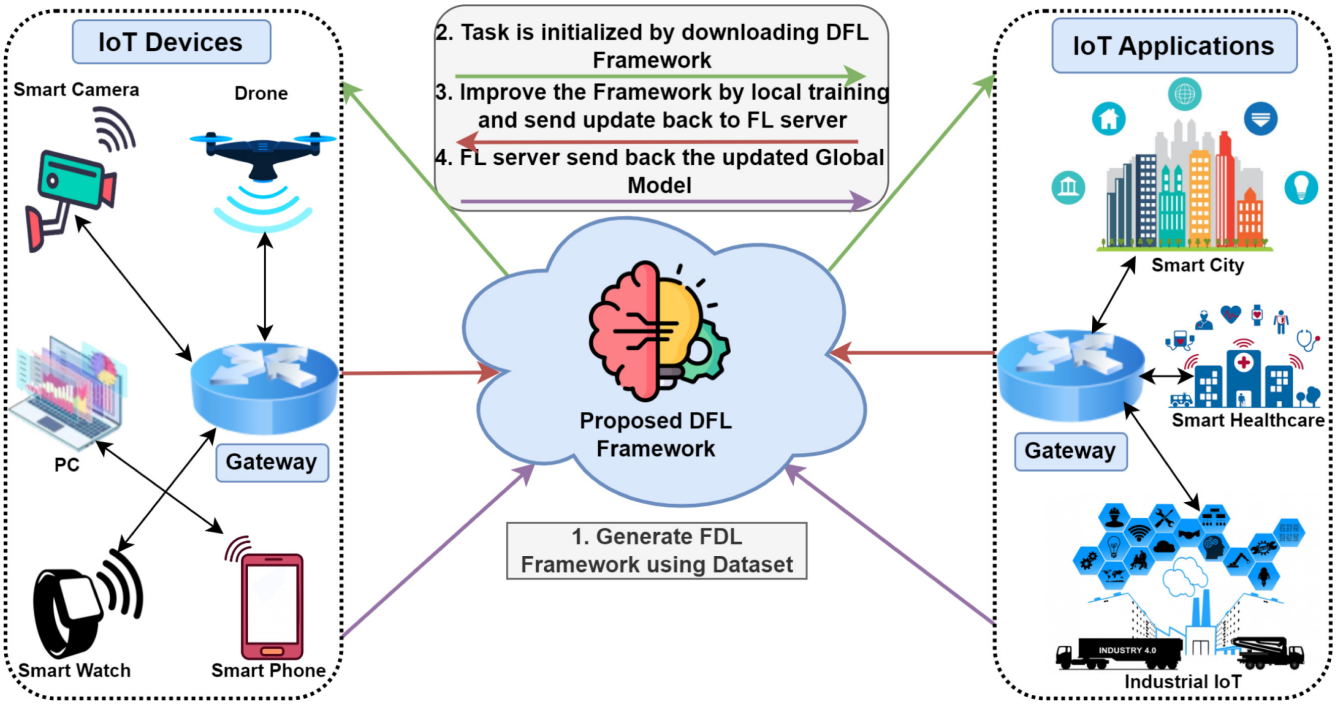


Figure 24: Proposed model by [24]

In another work, by [25], the authors address the security problem in IIoT networks, which are vulnerable to diverse cyber threats and attacks. The authors highlight that traditional attack detection methods are insufficient to protect privacy and security in IIoT networks. To tackle this issue, the paper introduces the novel Gradient Descent Scaling and Segmented Regression Fine-tuned Federated Learning (GDS-SRFFL) method, which aims to enhance the security of IIoT networks. The proposed methodology, shown in Fig. 25, involves applying Gradient Descent Scaling-based preprocessing to the raw dataset and using a Segmented Regression Fine-tuned Mini-batch Federated Learning model with the novelty of SoftMax Regression for intrusion detection. The experiment results demonstrate that the GDS-SRFFL method has significantly improved accuracy, precision, recall, specificity, and attack detection time compared to existing techniques, such as CNN + LSTM, Enhanced Deep and Ensemble learning, RNN, and other CNN methods.

The dataset used in the experimentation is a realistic testbed that comprises properties of Software-Defined Networking (SDN), Network Virtualization Function (NVF), and Service Orchestration (SO) to enable communications among the layers of edge, fog, and cloud in IIoT networks. The dataset includes profiles of IoT activities, such as IoT fridge and garage activities, and GPS tracker data, each with specific features relevant to IIoT network operations. The experiment results reveal that the GDS-SRFFL method outperforms existing techniques in terms of attack detection time, demonstrating its effectiveness in detecting IIoT attacks with minimal time consumption. The insights gained from the experiment results emphasize the potential of the GDS-SRFFL method in significantly improving the precision, recall, accuracy, and specificity of IIoT attack detection, thereby contributing to the overall security of IIoT networks. The paper also acknowledges certain limitations, such as the need for further research to integrate advanced swarm optimization techniques with deep learning to detect more advanced cyber-attacks and enhance the security of IIoT networks in real-time environments. Additionally, the authors emphasize the importance of addressing the limitations of the proposed methodology to ensure its applicability in diverse IIoT network scenarios.

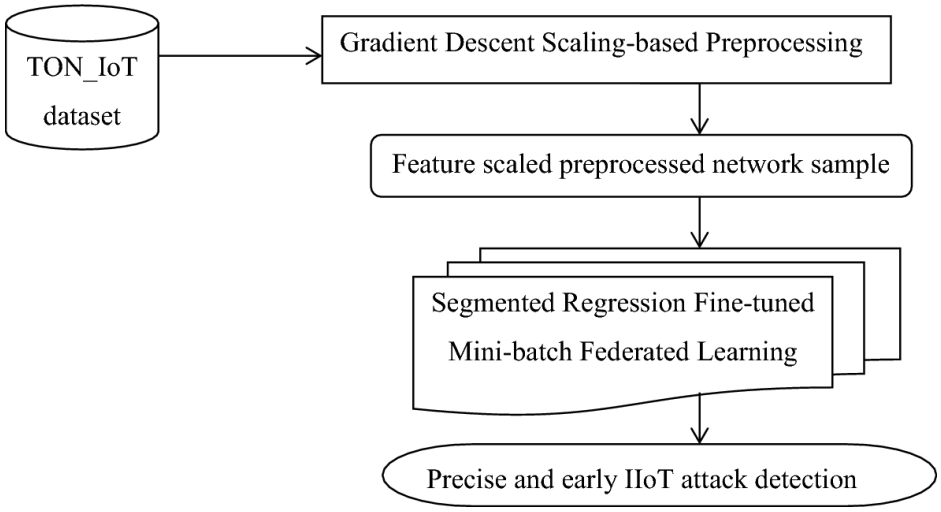


Figure 25: Proposed model by [25]

2.2.4 Zero-day Detection Works

Zero-day attacks exploit unknown vulnerabilities, posing critical challenges for IoT networks. With the growing prevalence of IoT devices, researchers have developed various detection strategies, leveraging machine learning, federated learning, and deep learning architectures to address these threats. This section explores key advancements in zero-day attack detection, focusing on methods that enhance accuracy, efficiency, and privacy while addressing constraints like limited data and real-time processing. Highlighting strengths, limitations, and future directions, it provides an overview of the state-of-the-art approaches aimed at securing IoT environments.

In the work by [67], a multi-step attack prediction system designed for resource-constrained IoT environments, such as smart homes. Addressing the limitations of existing techniques in detecting complex multi-step attacks, the proposed architecture incorporates several innovative features. It leverages a variational auto-encoder (VAE) for identifying both known and unknown threats and introduces a dual-domain defense strategy for enhanced detection capabilities. To enable real-time prediction, the system combines the hidden Markov model (HMM) with VAE and employs an aggregated HMM (AHMM) for low-latency attack prediction. The results demonstrate the system's ability to model and track multi-step attacks, including zero-day threats, and provide early warnings. Despite promising outcomes, the authors acknowledge limitations, such as the restricted scope of experimental scenarios, suggesting future work to extend the approach to more complex attack chains.

In another work by [68], they introduce a Federated Deep Learning (FDL) approach for detecting zero-day botnet attacks in IoT-edge devices, prioritizing user data privacy. Unlike centralized deep learning (CDL), which risks privacy breaches, the FDL method enables decentralized training across IoT-edge devices using an optimal deep neural network (DNN) architecture. A model parameter server coordinates local DNN training, with updates aggregated via the Federated Averaging (FedAvg) algorithm to produce a global model. Experiments using Bot-IoT and N-BaIoT datasets demonstrate the FDL model's advantages in classification performance, data privacy, low communication overhead, memory efficiency, and reduced network latency. The study shows that FDL outperforms CDL, localized deep learning (LDL), and distributed deep learning (DDL) models, although it has longer training times due to iterative communication rounds.

In a similar approach by [69], they present a federated learning framework for detecting zero-day botnet attacks in IoT networks, addressing limitations of centralized deep learning (DL) approaches. Traditional DL methods rely on centralized data sharing, which risks user privacy and performs sub-optimally against zero-day attacks. The

proposed framework introduces a novel K-greedy aggregation algorithm, which leverages uncertainty assessment to enhance global model convergence while preserving data privacy. Evaluations using the N-BaIoT dataset show that the FL-based framework with K-greedy aggregation outperforms baseline FL aggregation methods, achieving effective detection of zero-day botnet attacks.

In a different work by [70], they investigate the reliability of Convolutional Neural Networks (CNNs) in detecting zero-day botnet attacks in IoT networks, focusing on enhancing Intrusion Detection Systems. Botnets, particularly through attacks like DoS and DDoS, pose critical threats to IoT networks by overloading resources and causing service disruptions. The study evaluates CNN classifiers using regularization techniques (L1 and L2) to mitigate overfitting and improve detection accuracy for unseen attacks. Experimental results reveal that CNNs outperform classical machine learning methods, especially in zero-day scenarios. Using nine selected features from the Bot-IoT dataset, the CNN model with L2 regularization achieves the highest performance, with approximately 91% accuracy and a 94% ROC score in detecting zero-day attacks. The study demonstrates the potential of regularized CNNs in addressing unseen IoT attack scenarios.

In another work by [71], they address the critical threat of zero-day vulnerabilities and attacks in IoT networks, emphasizing their potential to exploit network defenses and disrupt device functionality. To counter these threats, the authors propose a consensus framework for detection and mitigation, leveraging the contextual behavior of IoT devices. The framework integrates an alert message protocol and a critical data-sharing protocol to ensure reliable communication during attack mitigation. Numerical analysis demonstrates the framework's effectiveness in detecting and eliminating zero-day attacks without compromising network performance. Comparative results highlight its advantages in latency and overhead, supporting its suitability for maintaining high-performance IoT networks even under attack conditions.

In an interesting work by [72], they focus on the challenge of detecting zero-day attacks in IoT and Internet of Battle Things (IoBT) applications. These attacks exploit unknown vulnerabilities, making them difficult to identify and mitigate effectively. The study introduces novel network flow-based features engineered from raw packet data to improve detection rates. The authors tested six traditional machine learning models (DT, SVM, KNN, LR, NB, and RF) in two settings: one with raw features and another with the proposed complex flow-based features. Results showed that models trained on raw features failed to detect certain zero-day attacks, with only the Gaussian Naive Bayes (NB) model successfully identifying the MQTT-Brute Force attack. In contrast, the flow-based features yielded significantly higher performance, particularly for the Random Forest (RF) model, which achieved an average F1 score of 0.999 and a 100% detection rate across scenarios. The study highlights the potential of the proposed flow-based features for real-time IoT and IoBT applications, as flows were limited to 16 packets for faster detection. Future research will validate these findings with additional datasets and evaluate their generalizability through split-at-scenario cross-validation.

Finally, in [73] the authors introduce NERO, a deep learning pipeline designed for detecting zero-day attacks (ZDAs) in IoT environments. The method emphasizes data efficiency by leveraging Neural Algorithmic Reasoning (NAR) and Prototypical meta-learning frameworks, enabling the pipeline to perform few-shot learning. This is particularly relevant in IoT scenarios, where data availability can be constrained due to privacy concerns. The NERO architecture adopts an encode-process-decode structure, inspired by neural algorithmic reasoning, to generalize the detection and classification of ZDAs. It models high-level abstract reasoning, decoupling its behavior from use-case-specific data distributions. This abstraction allows it to distinguish between type_A and type_B attacks with balanced binary classification accuracy. Unlike neuro-symbolic AI approaches, NERO is designed to operate

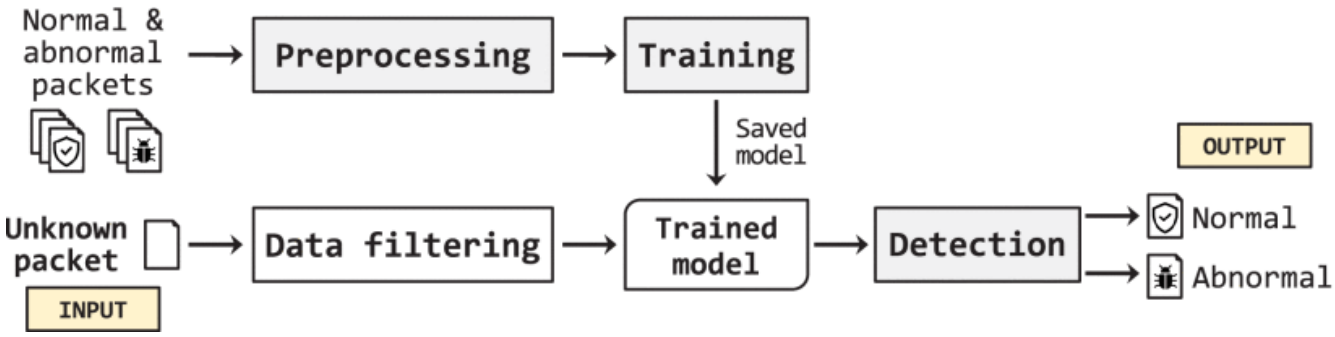


Figure 26: Proposed model by [26]

with low computational and network overhead, making it suitable for IoT deployments. The authors suggest that the NERO pipeline represents an implicit mapping of conditional discriminative criteria used in other frameworks. Future research will focus on deploying NERO in real IoT environments for online learning and enhancing its open-set classification accuracy to better handle unseen attacks.

2.2.5 Anomaly Detection Works

Anomaly detection plays a crucial role in fortifying the security of IoT-enabled smart homes by identifying deviations from normal behavior patterns. Here, we delve into studies focused on anomaly detection techniques tailored to the unique characteristics of smart home environments. Our exploration encompasses diverse approaches aimed at effectively detecting and mitigating anomalous activities.

In [26], the identified problem revolves around the security vulnerabilities that have emerged in the Z-Wave protocol due to manufacturers prioritizing device functionality over security. The proposed solution presented in Fig. 26 is the ZMAD system, a lightweight model-based approach for detecting anomalies in the Z-Wave protocol.

The experiment results highlight the success of ZMAD in detecting abnormal behaviors within Z-Wave networks with accuracy rates greater than 97%, showcasing its potential to improve the security posture of smart homes significantly. The insights gained from this research emphasize the importance of tailored anomaly detection methods that consider the unique characteristics of the Z-Wave protocol, underscoring the necessity for proactive security measures in IoT environments. By focusing on packet formalization and centralized learning techniques, ZMAD enhances anomaly detection accuracy and expands the coverage of Z-Wave Command Classes, providing a more comprehensive approach to security monitoring.

However, despite its strengths, the ZMAD system is not without limitations. The scalability of ZMAD to larger and more complex smart home networks may pose challenges, requiring further optimization and refinement for broader deployment. Additionally, the reliance on specific packet fields for anomaly detection could limit the system's effectiveness in scenarios where attackers manipulate other packet fields not considered by ZMAD, highlighting the need for ongoing research and development. These limitations underscore the continuous evolution and refinement required in anomaly detection systems to combat emerging security threats in smart home automation environments effectively.

In [74], authors highlight the potential security risks associated with smart devices, such as unauthorized access and misuse of cyber attacks, and propose the application of big data and machine learning to identify anomalous activities in smart home environments. The proposed solution involves training a Hidden Markov Model (HMM) on network-level sensor data collected from a test bed with multiple sensors and smart devices.

The experiment results demonstrated an accuracy of 97% in identifying potential anomalies that indicate attacks, demonstrating the approach's effectiveness in detecting abnormal behaviors in smart home environments. Further-

more, the paper discusses the data collection process, which involves aggregating data from various IoT sensors deployed in a smart home environment. The data was categorized into behavioral and network data, providing a comprehensive understanding of the smart home environment and enabling the modeling of typical behaviors using HMM models.

However, this study also suffers from some limitations. The study primarily focuses on the application of HMM and does not extensively explore other anomaly detection techniques available in the literature. Additionally, the experiment was conducted in a controlled testbed environment, and the real-world applicability of the proposed solution in diverse smart home settings remains to be fully validated. Furthermore, the paper does not extensively discuss the computational and resource requirements for implementing the HMM-based anomaly detection model in real-world smart home environments, which could be a potential limitation in practical deployment. Moreover, the paper does not delve into the potential challenges or considerations related to privacy and data security when implementing anomaly detection models in smart home environments.

In [27], authors address the critical issue of detecting anomalies in Appified smart homes, where anomalies can result from attacks or device malfunctions, leading to severe consequences. The authors highlight that existing anomaly detection systems utilizing data mining techniques suffer from high false alarm rates and miss many real anomalies due to the lack of consideration of each event's semantics, which can be acquired from smart apps, device types, and functionalities. The paper proposes HAWatcher, a semantics-aware anomaly detection system that models a smart home's normal behaviors based on event logs and semantics, illustrated in Fig. 27. HAWatcher generates hypothetical correlations according to semantic information, such as apps, device types, relations, and installation locations, and verifies them with event logs. The system refines the mined correlations using correlations extracted from the installed smart apps and uses a Shadow Execution engine to simulate the smart home's normal behaviors. During runtime, anomalies between devices' real-world and simulated states are reported as anomalies. The authors evaluate the prototype on the SmartThings platform in four real-world testbeds and test it against 62 different anomaly cases, demonstrating that HAWatcher achieves high accuracy, significantly outperforming prior approaches.

The experiment results presented in the paper provide valuable insights into the effectiveness of HAWatcher in detecting anomalies in Appified smart homes. The authors demonstrate the system's high accuracy and ability to outperform prior approaches, highlighting the potential of semantics-aware anomaly detection in addressing the limitations of existing data mining-based methods. The insights gained from the experiment results emphasize the importance of incorporating semantic information, such as automation logic, device types, relations, and installation locations, to improve anomaly detection accuracy in smart homes.

The paper also discusses the limitations of the proposed solution, shedding light on the challenges that need to be addressed for the practical implementation and real-world deployment of HAWatcher. These limitations include the need to represent diverse semantic information in the form of event logs, the identification and resolution of conflicts between system behavior patterns derived from smart apps and those mined from event logs, and the lack of effective methods to update the system profiling when smart apps change.

In [28], authors address the growing concern of security threats targeting smart building automation systems, particularly in the context of Home Area Networks (HANs). The identified problem centers on the vulnerability of smart building automation systems to hacking attacks, necessitating robust anomaly detection mechanisms to protect both the HAN and the network resources of service providers. The proposed solution, presented in Fig. 28, introduces a novel strategy for anomaly detection that leverages machine learning techniques to classify monitoring

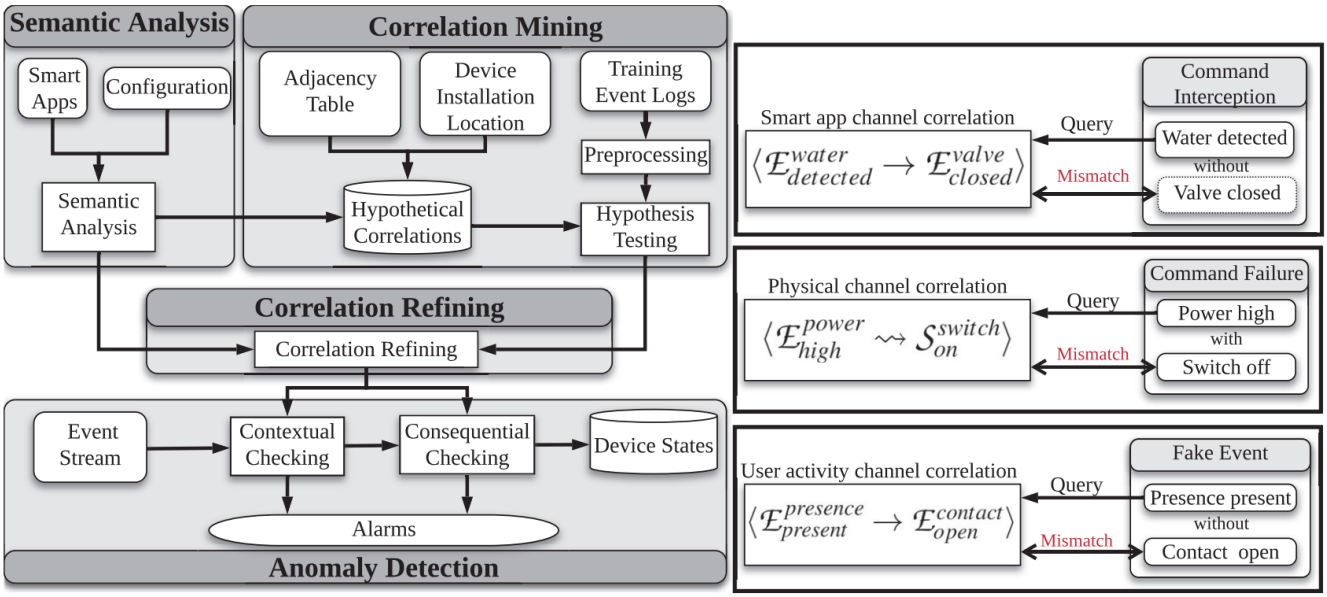


Figure 27: Proposed model by [27]

data and correlate anomalies across network resources, thereby establishing a shared responsibility between service clients and network providers for security.

The experiment results demonstrate the effectiveness of the anomaly detection process, as evidenced by the successful identification and correlation of anomalies in simulated subnetwork types, such as Zigbee and WiFi networks. The insights gained from the experiment underscore the importance of anomaly detection as a collaborative effort between service clients and network providers, with the potential to protect both HANs and network resources. However, the study's limitations include the need for further real-world experimentation and validation of the proposed anomaly detection strategy in diverse smart home automation IoT systems.

In [29], authors highlight the vulnerability of smart home IoT devices to cyber-attacks due to the lack of substantial security measures. To combat this, the paper introduces the Graph-based Outlier Detection in the IoT (GODIT) approach, which represents smart home IoT traffic as a real-time graph stream and efficiently processes graph data to detect DoS attacks in real-time. The proposed solution, as depicted in Fig. 29, is underpinned by a shingling-based graph sketching technique, enabling higher dimensional graph data representation into a memory-efficient fixed-sized vector.

The experiment results presented in the paper underscore the efficacy of the GODIT approach in detecting DoS attacks in smart home IoT devices. The experimental setup involved the use of real-world smart home IoT traffic data. The data encompassed various attacks, including ARP Spoofing, Ping of Death, and Smurf Attacks, with most of the attacks being DoS-related. The GODIT approach effectively detected anomalous traffic patterns, showcasing its superiority over traditional machine learning approaches. Furthermore, the experimental results demonstrated the feasibility of using the proposed sketching approach to classify normal and DoS attack graphs in static and streaming settings. The insights gained from the experiments underscore the potential of the GODIT approach in enhancing anomaly detection in graph streams, thereby contributing to the advancement of security measures for smart home IoT devices.

One limitation highlighted in the paper is the need for further scalability and robustness testing of the GODIT approach in streaming scenarios. Moreover, the paper underscores the need for future work to enhance the proposed methodology for graph sketching and address potential challenges in real-time anomaly detection.

In the work by [30], the authors address the critical issue of detecting and classifying malware attacks in the context

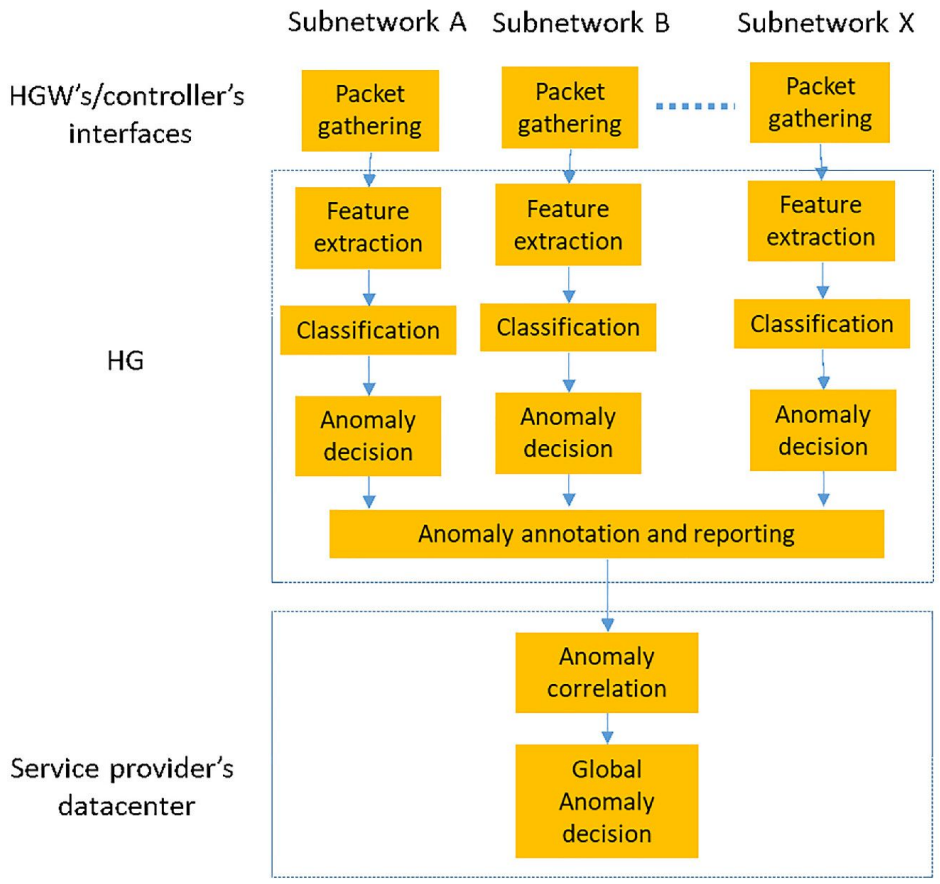


Figure 28: Proposed model by [28]

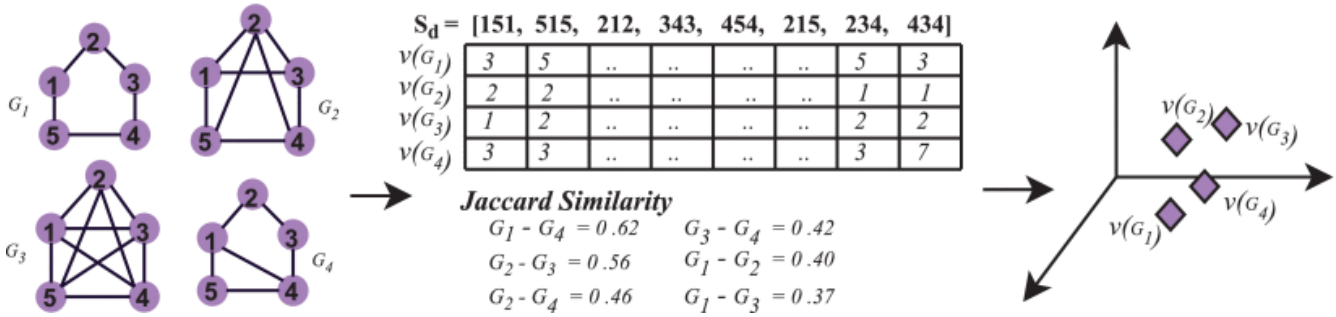


Figure 29: Proposed model by [29]

of the Industrial Internet of Things (IIoT). The authors highlight the challenges posed by the proliferation of interconnected devices in IIoT, which have created a new attack surface for threat actors, necessitating the development of advanced malware detection systems. To tackle this problem, the paper proposes a novel image-based malware detection system, illustrated in Fig. 30, that leverages software-defined networking (SDN) honeypots, convolution neural networks (CNN), and a two-level autoencoder. The proposed system transforms binary programs into grayscale images and extracts textural features using deep CNN architectures via transfer learning, followed by dimension reduction using a custom two-level autoencoder. The experimental results demonstrate the superiority of the proposed methodology, achieving high test accuracy, precision, recall, and f1-score, as well as fast detection capabilities.

The research utilizes the publicly benchmarked MalImg dataset, which consists of 9339 total malware samples belonging to 25 different malware classes, to measure the effectiveness of the proposed methodology in malware detection and classification. The dataset is divided into training and testing sets, and the experiments are strategically designed to optimize hyperparameters and evaluate different feature extractor methods used in the proposed architecture. The experimental results showcase the efficiency of the proposed model, with faster detection times and improved performance compared to existing methods, demonstrating the practical applicability of the developed system in real-world IIoT environments.

The insights gained from the experimental results underscore the potential of the proposed image-based malware detection system to offer a scalable and industry-suitable solution for detecting malware in IIoT environments without relying on intensive feature engineering and minimizing the need for domain expertise. However, the paper also acknowledges certain limitations, such as the lack of consideration for the training time of the model, which could impact the comparative analysis with existing methods, and the need for further exploration of the proposed methodology in diverse IIoT environments. Despite these limitations, the proposed image-based malware detection system offers a promising approach to addressing the evolving threat landscape in IIoT environments, contributing to advancing cybersecurity measures in Industry 4.0.

2.2.6 Bot Detection Works

Bots pose significant security risks in smart home networks, ranging from data breaches to orchestrated attacks. This subsection examines research efforts to detect and mitigate bot presence within IoT ecosystems. We evaluate strategies and methodologies devised to identify and neutralize bot-related threats effectively.

In [31], authors address the critical issue of classifying highly imbalanced network traffic data in Smart Home Networks (SHN) that are vulnerable to complex botnet attacks. The proposed solution, presented in Fig. 31, involves the use of a Stacked Recurrent Neural Network (SRNN) model, which leverages multiple layers of RNN to learn hierarchical representations of the imbalanced network traffic data with varying levels of abstraction. The experiment results demonstrate the superior performance of the SRNN model compared to traditional RNN models, showcasing its ability to effectively handle over-fitting and achieve better generalization in detecting network traffic samples of the minority classes. The insights gained from the experiments highlight the robustness and discriminating feature learning capabilities of the SRNN model, emphasizing its potential for enhancing botnet detection in SHNs. However, the paper also acknowledges certain limitations, such as the computational expense of developing separate ML/DL models for each botnet attack type and the need for further research to address these challenges comprehensively.

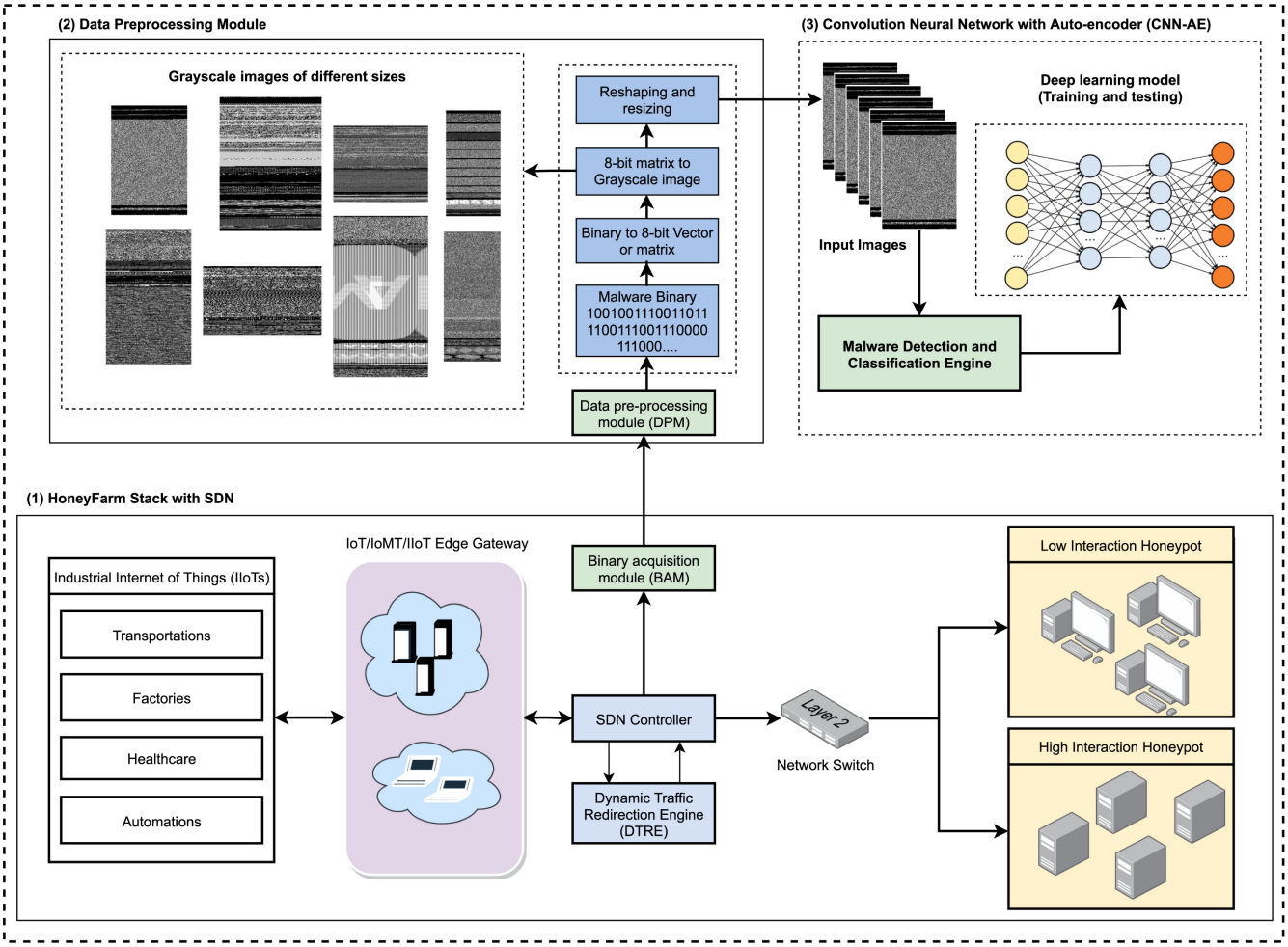


Figure 30: Proposed model by [30]

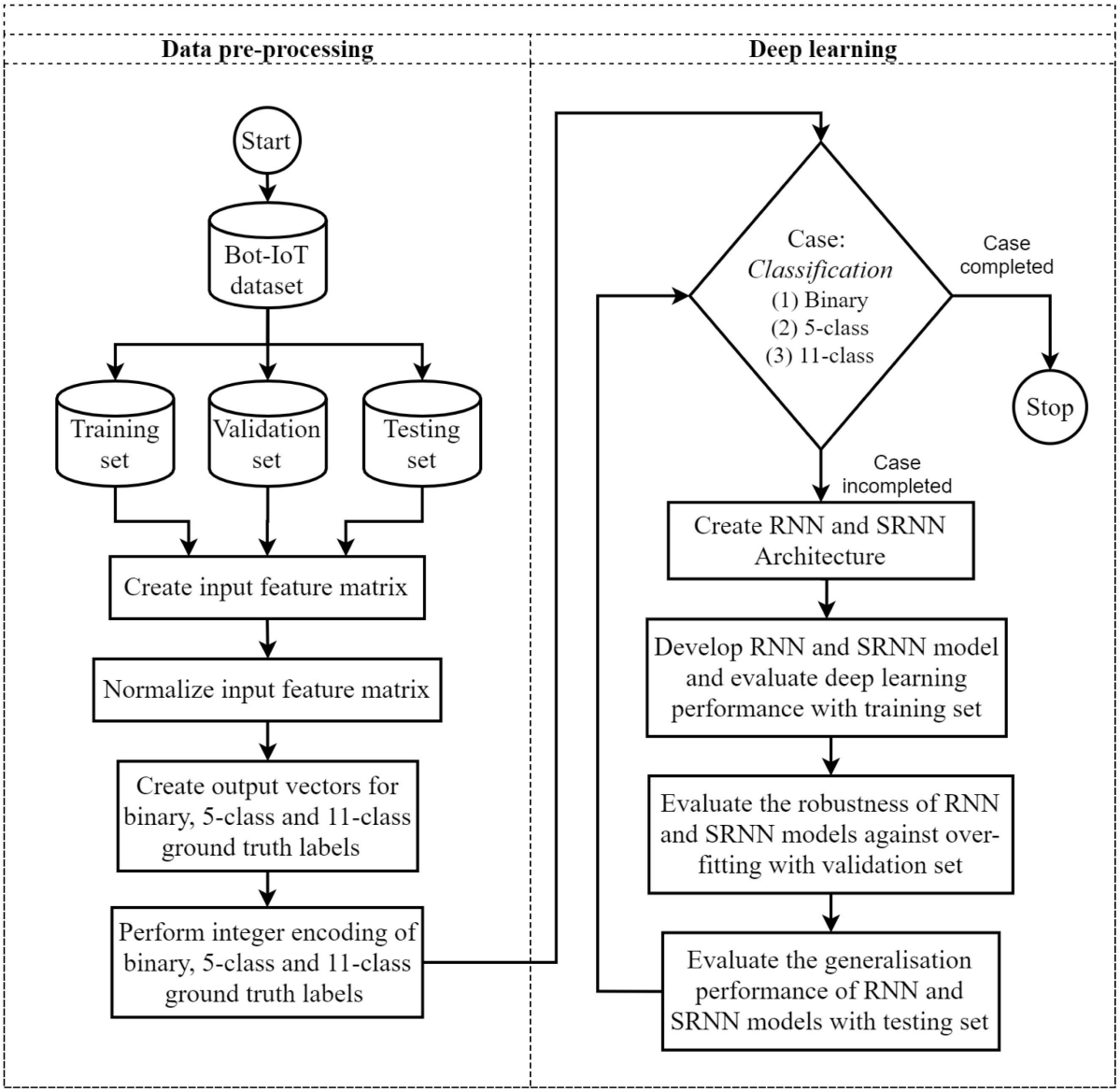


Figure 31: Proposed model by [31]

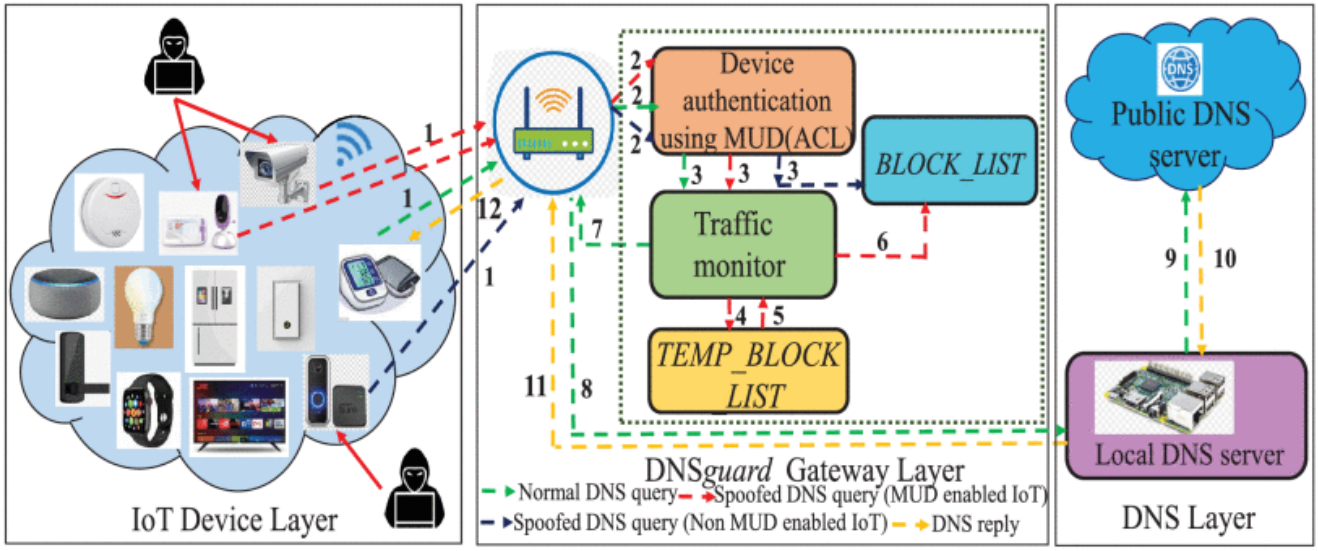


Figure 32: Proposed model by [32]

2.2.7 DNS-related Attack Detection Works

Domain Name System (DNS)-related attacks pose significant risks to the integrity and availability of smart home networks. This subsection explores research efforts dedicated to detecting and mitigating DNS-related attacks targeting IoT devices. We examine methodologies and strategies to fortify DNS infrastructure and safeguard against various DNS-based threats.

In [32], authors address the critical issue of DNS flooding attacks in IoT networks, which significantly threaten the quality of service (QoS) in smart home infrastructure. As presented in Fig. 32, the proposed solution, DNSguard, leverages Manufacturer Usage Description (MUD) to authenticate IoT devices and monitor DNS traffic, thereby mitigating DNS flooding attacks at the local DNS server. The experiment results demonstrate a substantial decrease in DNS response time to 67.2% and a reduction in CPU utilization to 7%. These findings provide valuable insights into the efficacy of the DNSguard framework in enhancing the security and resilience of IoT networks against DNS flooding attacks. However, the proposed solution suffers from issues such as scalability and potential overhead associated with MUD authentication in large-scale IoT deployments.

2.2.8 User Behavior Profiling Works

Understanding user behavior is essential for enhancing the usability and security of smart home systems. This subsection delves into research endeavors focused on user behavior profiling within IoT environments. We explore methodologies and frameworks to analyze and model user interactions with smart home devices, enabling personalized services and effective security measures.

In [33], authors address the issue of detecting cyberattacks on home IoT devices by leveraging user behavior and home conditions. The proposed solution, illustrated in Fig. 33, involves a method that learns sequences of user behaviors based on conditions such as time of day, temperature, and humidity. When an operation command is received, the method compares the current sequence with learned sequences for the current condition, flagging any discrepancies as anomalous operations. The experiment involved constructing a network of home IoT devices in a laboratory setting and allowing four subjects to operate the devices for three months. The experiment's results

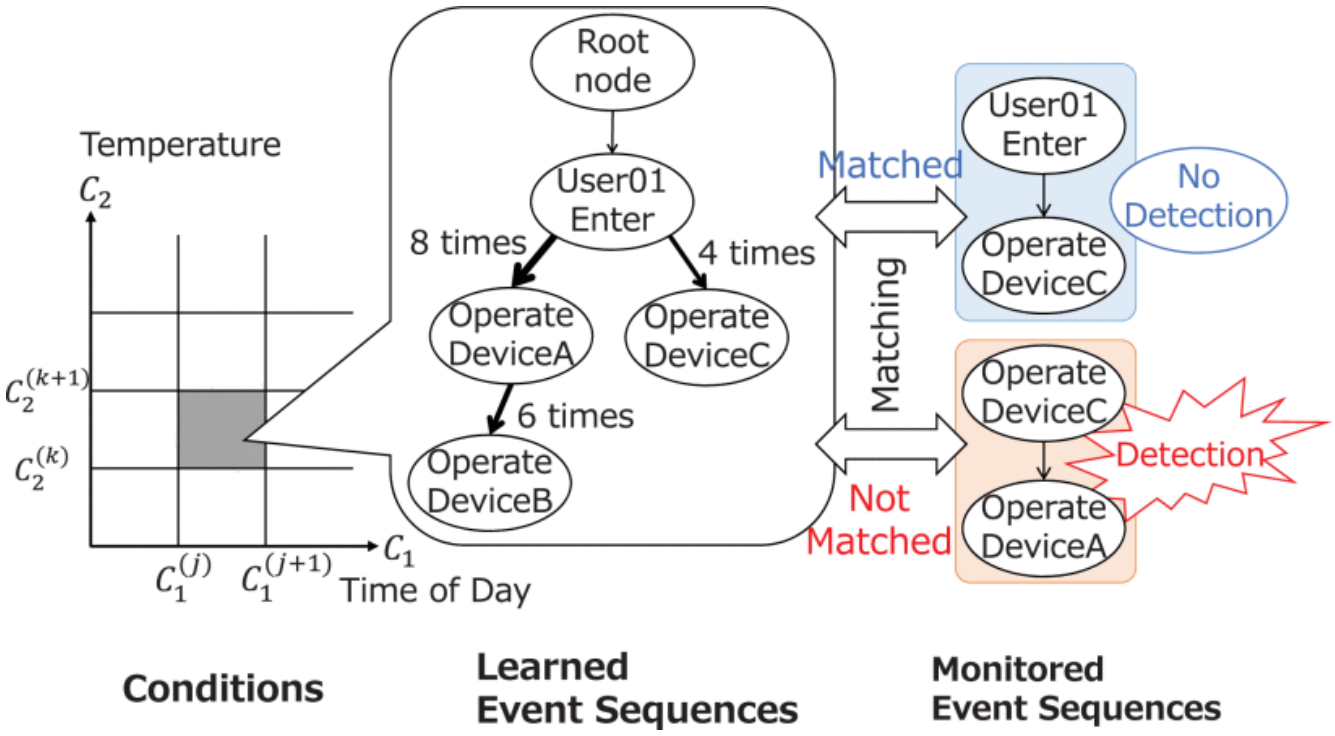


Figure 33: Proposed model by [33]

demonstrated a detection ratio exceeding 90% for anomalous operations, with less than 10% misdetections. The insights gained from the experiment highlight the effectiveness of considering condition information in detecting anomalous operations while shedding light on misdetections' impact. However, the study also acknowledges the limitations of the method, particularly in distinguishing legitimate operations of the coffee maker from anomalous operations, and the need to improve detection accuracy as part of future work.

In conclusion, this section has provided a comprehensive overview of the current state of research in smart home profiling, organized according to prevalent threat models outlined in the existing literature. Through meticulous analysis of seminal works, we have explored various aspects of device profiling, detection of Denial of Service (DoS) attacks, intrusion detection mechanisms, anomaly detection techniques, bot detection strategies, spam detection methods, DNS-related attack countermeasures, and user behavior profiling frameworks tailored for IoT-enabled smart homes. By systematically examining the problem formulations, methodologies, experimental findings, insights, and limitations of each reviewed study, this section contributes to a deeper understanding of the evolving landscape of smart home security. It lays a foundation for future research endeavors in this domain.

2.3 Synthesis

While previous works have significantly contributed to the advancement of Smart Home security research, it is crucial to acknowledge and address their inherent limitations. Recognizing these shortcomings underscores the importance of developing new models and datasets to overcome existing challenges and propel the field forward. By learning from the successes and pitfalls of previous works, researchers can inform the design of more realistic and comprehensive datasets, ultimately fostering the development of more effective and adaptable detection methods. Below, we have compiled a list highlighting the shortcomings identified in the previous works:

1. ***Lack of Real-time or Early Detection:*** The absence of real-time or early detection capabilities in previous models undermines the timeliness of threat response. Delays in identifying and responding to intrusions can

lead to significant security breaches and compromise network integrity.

2. ***Limited Ability to Detect Unknown or Zero-day Attacks:*** The limited capability to detect unknown or zero-day attacks in prior models reflects a gap in adaptive threat detection. Without mechanisms to identify novel attack patterns, intrusion detection systems may fail to respond effectively to emerging threats.

3. ***Lack of Consideration of Z-wave or Zigbee Traffic Packets:*** Many existing models overlook the unique characteristics of IoT protocols such as Z-wave or Zigbee, instead focusing solely on TCP/IP traffic. This oversight results in incomplete threat detection capabilities, as attacks leveraging these protocols may go undetected.

4. ***Lack of Realistic Attack Scenarios:*** Previous studies often fail to present realistic attack scenarios reflective of the dynamic cybersecurity landscape. Failure to incorporate contemporary threat scenarios hampers the efficacy of detection systems, leaving smart home networks vulnerable to modern attack methodologies.

5. ***Low Accuracy and High False Positive Rate:*** Shortcomings in accuracy and a high false positive rate in prior methods indicate a need for improvement. These issues can result in ineffective threat detection and unnecessary alerts, straining network resources and impeding the reliability of intrusion detection systems.

6. ***Lack of Comprehensive Feature Set:*** Insufficient feature sets in previous models limit the depth of analysis and understanding of network activities. The absence of a comprehensive set of features hampers the ability to capture nuanced patterns and behaviors, compromising the efficacy of intrusion detection.

7. ***Absence of Multi-Modal Data:*** The absence of multi-modal data integration in previous works limits the depth of analysis and understanding of smart home network behaviors. Failure to incorporate diverse data sources hampers the accuracy and effectiveness of detection systems in identifying and responding to malicious activities.

8. ***Lack of Attacks Related Specifically to IoT Protocols:*** Previous models neglect attacks targeting IoT protocols such as Z-wave or Zigbee, focusing instead on generic network-based threats. This oversight compromises the ability to detect and mitigate attacks tailored to exploit vulnerabilities inherent in IoT communication protocols.

9. ***Limited Diversity of Attacks:*** Many existing approaches suffer from a narrow focus on specific types of attacks, failing to account for the diverse range of threats that smart home networks may encounter.

10. ***Outdated Threat Scenarios:*** Prior models often rely on outdated scenarios, overlooking cybersecurity threats' dynamic nature.

11. ***Lack of Specific IoT Devices in the Smart Home Architecture:*** The absence of specific IoT devices within smart home architectures poses a significant limitation in previous works. Failure to incorporate a comprehensive range of devices limits the ability to detect attacks targeting specific device types, leaving smart home networks susceptible to exploitation.

12. ***Lack of Consideration of Different Types of Homes and Users:*** Previous models often fail to account for the diverse range of smart home architectures and user behaviors. This lack of consideration limits the

applicability and effectiveness of intrusion detection systems across varying smart home environments and user demographics.

13. ***Lack of Privacy Concerns:*** Privacy concerns within smart home environments are often overlooked in previous works, leading to insufficient protections for sensitive user data. Failure to address privacy considerations compromises the integrity of detection systems and undermines user trust in smart home technologies.

14. ***Lack of Effective Behavioral Profiling Model and Visualization:*** Many existing approaches lack a comprehensive behavioral profiling model, hindering accurate threat detection. Additionally, inadequate visualization methods limit the interpretability of network activities, impeding effective analysis and decision-making.

15. ***Necessity of Prior Knowledge to Update Each Attack:*** The requirement for prior knowledge to update each attack in previous models poses a significant limitation. This dependency on pre-existing information hampers the adaptability of intrusion detection systems, making them less effective in scenarios where emerging threats lack historical precedent. The need for constant manual updates also introduces challenges in maintaining the relevance and accuracy of the detection mechanisms.

16. ***Prone to Attacker Evasion with System Knowledge:*** Vulnerability to evasion by attackers with system knowledge highlights a critical weakness. Without robust countermeasures, intrusion detection systems may be circumvented, allowing knowledgeable attackers to exploit vulnerabilities undetected.

Recognizing the importance of overcoming these limitations, our current work aims to address the first 15 challenges outlined in this thorough analysis.

3 Proposed Profiling Model

This section presents the core contribution of this thesis, a novel intrusion detection system specifically designed to address critical security gaps within the IoT ecosystem. The section begins by outlining the rationale and motivation for developing this model, highlighting the limitations of existing approaches and the unique challenges posed by IoT environments. Following this, a comprehensive description of the model's architecture, methodology, and underlying techniques is provided. Finally, the chapter concludes with an in-depth discussion of how the proposed model not only overcomes the limitations identified in the Literature Review but also integrates the key concepts and objectives introduced in the motivation, positioning it as a robust solution for enhancing IoT security.

3.1 Motivation

The growing reliance on IoT devices, particularly within smart home environments, underscores an urgent need for effective and adaptable security measures. As IoT ecosystems expand, they increasingly attract various cyber-attacks, ranging from Denial of Service (DoS) and Distributed Denial of Service (DDoS) to more advanced threats like privilege escalation, man-in-the-middle attacks, malware injection, and exploitation of protocol-specific vulnerabilities. These attacks are often disruptive and challenging to detect with conventional security mechanisms, which are generally not optimized for the specific characteristics of IoT networks. The dynamic nature of IoT environments, characterized by devices with limited processing capabilities, diverse protocols, and high degrees of interconnectedness, demands a specialized detection model capable of effectively addressing these challenges. Thus, a tailored approach is necessary to ensure reliable security within such unique and varied conditions.

IoT networks are vulnerable to a diverse array of security threats. While traditional attacks like DoS remain significant, other forms of intrusion, such as protocol-specific vulnerabilities and exploitation of device-level weaknesses, are becoming increasingly common due to the diversity of IoT communication standards and device types. Therefore, an effective detection system for IoT must be comprehensive and capable of identifying a broad range of attack types, particularly those most prevalent in IoT systems. This level of adaptability is critical for preserving the safety and integrity of IoT networks, which often operate in sensitive environments, including residential, healthcare, and industrial settings. The ability to respond to a broad spectrum of threats is essential for maintaining the resilience of IoT systems in these critical domains.

For this thesis, the focus is narrowed specifically to smart home environments. Several factors drive this choice. First, the smart home sector represents one of the fastest-growing segments within the IoT market, with millions of households worldwide adopting connected devices for security, convenience, and energy management. The increasing deployment of smart home devices—such as thermostats, lighting systems, surveillance cameras, and voice-activated assistants presents vast opportunities but also introduces considerable security risks. These devices often lack robust security features and may be particularly susceptible to exploitation, making smart homes a valuable focal area for developing and testing an effective detection model.

Additionally, smart homes present a particularly challenging environment for intrusion detection due to the variety of devices and communication protocols. Each household may employ different combinations of technologies, ranging from Wi-Fi and Zigbee to Z-Wave and Bluetooth, resulting in a complex and heterogeneous network landscape. Thus, the proposed detection model must be protocol-agnostic and function seamlessly across all types of smart home configurations regardless of the specific devices or communication standards employed. This flexibility is vital, as it ensures the model's applicability across various IoT deployments within and beyond the context

of smart homes. Such versatility in design is essential to ensure that the model can be widely implemented and remain effective in diverse IoT environments.

Another motivating factor behind the proposed detection model is the issue of device malfunction, which may arise from hardware failures, software bugs, or malicious interference. Malfunctioning devices can exhibit abnormal behavior that could either indicate an attack or stem from system failures, and distinguishing between these scenarios is crucial for maintaining the functionality of the smart home ecosystem. Thus, the detection model must focus on identifying direct security threats and incorporate profiling mechanisms to detect anomalies that may suggest underlying problems. By accounting for both malicious intrusions and benign malfunctions, the model will enhance the resilience and reliability of smart home networks, promoting their secure and uninterrupted operation.

In conclusion, the motivation for developing this detection model stems from the need to address the unique security challenges posed by IoT environments. The model is intended to detect various attacks, operate across diverse smart home configurations, and handle malicious activities and device malfunctions. While the primary focus of this thesis is on smart homes, the underlying framework is designed to be adaptable to any IoT environment, ensuring its relevance and effectiveness across various contexts. This adaptability, combined with a focus on comprehensive threat detection, positions the proposed model as a critical advancement in securing the increasingly interconnected world of IoT.

3.2 Proposed Solution

We introduce a profiling model designed for detecting and identifying malicious activities to address the complex security challenges presented by smart home environments. Considering the unique behaviors and interactions of devices in a smart home, this model represents a departure from traditional anomaly detection methods. The foundation of our approach lies in the idea that a general device profiling strategy is insufficient for malicious activity detection. Device behavior can vary significantly based on factors such as device location, home architecture, and user habits. For instance, a smart lamp in a kitchen will exhibit different patterns than one in a garage. This variability highlights the need for individualized profiling within each smart home context rather than applying a universal detection model.

Traditional intrusion detection systems are often limited by training on a narrow range of malicious activities or specific environments. This makes them ineffective in detecting diverse or zero-day attacks in varied home setups. This limitation arises because existing models are typically tailored to specific homes and device configurations, reducing their applicability to new or different contexts. Thus, we argue that any effective IDS for smart homes should be tailored to each unique home environment, accounting for its devices, architectural nuances, and user behaviors. Creating a one-size-fits-all IDS is impractical, given the diverse range of home layouts and user interactions.

Our proposed model introduces the overarching detection concept in accessible terms before delving into mathematical frameworks, AI architecture, and other technical details. We believe that a robust smart home detection model should adapt to local data, classify distinct behavior types, and precisely identify the nature of each malicious activity. Different malicious actions require tailored responses. For example, a brute-force attack attempting to access the hub necessitates a different approach than a Distributed Denial of Service (DDoS) attack aimed at draining a device's battery by sending repeated packets to an unreachable hub. In such cases, identifying the type and source of the attack allows for more effective response strategies, as making a wrong decision might not only fail to stop the attack but also worsen the situation by compromising the entire home environment. Consequently,

our model incorporates a multi-class classification mechanism that enables targeted decision-making, minimizing potential harm from incorrect responses.

Additionally, a comprehensive IDS for smart homes must analyze individual device behavior and interaction among devices, as these relationships can reveal valuable insights. For example, a typical scenario might involve a motion sensor and door lock, where motion detection precedes door unlocking. If the door unlocks without any motion detected, it could indicate unauthorized remote access to the lock. Such inter-device correlations are critical to detecting atypical behaviors that could signify security breaches.

The proposed model must also consider temporal patterns in device behaviors. Devices in a smart home exhibit different usage patterns throughout the day and across seasons. For instance, door locks may be used more frequently during the day, while winter conditions may increase energy consumption or alter temperature sensor readings. These environmental factors, including daylight variations (such as the significant seasonal daylight difference in some areas), must be integrated into the IDS, as they directly influence device and network behaviors.

Our proposed model observes individual and relational device behaviors within specific daily and seasonal time windows to address these challenges. The general process involves converting smart home conditions for each time window into a graph structure, aggregating these graphs, and learning the home's behavior patterns. To learn the behavior effectively, we extract relevant features from the created graph in a tabular format that learning algorithms can process. We then employ a multi-layered framework to detect and identify attacks, with models selected and trained based on data characteristics and classification targets. We refer to this entire approach as Internal Behavior Learning, which involves learning the inter-device communication behavior within the smart home environment.

Internal Behavior Learning enhances explainability by considering each device's interactions within the smart home in different time windows. These interactions include the individual and relation behavior of each device. This enables differentiation between devices of the same type in varying contexts (e.g., two door sensors on separate doors). Since data exchange between devices is typically encrypted, our model analyzes transmission characteristics like packet size and frequency within specified time windows. This approach provides insight into each device's network behavior relative to different times of day.

Internal data is typically transmitted over protocols specific to IoT devices like Z-Wave, Zigbee, and Wi-Fi. However, to comprehensively detect malicious activities that target the smart home, we must also monitor external data—internet-based interactions between the smart home hub and external systems. External data, in contrast to internal data, uses IP networking to connect with third-party applications or user devices over the internet for monitoring and control functions. Thus, the proposed model further incorporates External Behavior Learning, focusing on interactions between the smart home hub and the internet. Since most smart home devices lack direct internet access due to limited processing capabilities, attacks targeting the hub can disrupt the entire smart home network. Our solution, therefore, tracks these internet-based communication patterns, monitoring for atypical behaviors by analyzing hub activity in different time windows throughout the day. For instance, the model detects unusual user interactions with the hub, such as an unexpected device connection late at night, which could signal potential malicious activity.

Both data types serve critical roles; internal data captures localized interactions within the home network, while external data tracks internet-based activity that could introduce broader network vulnerabilities. Considering both, our model achieves a holistic view of the smart home's security, recognizing that threats may arise within the home network and from internet-based entry points.

For that reason, the proposed model is designed to account for two primary behavioral categories, internal and

external behaviors. Each can independently detect a range of malicious activities. Although some overlap exists between the types of malicious activities each can identify, their combined analysis ensures comprehensive coverage of all potential threats to a smart home environment. Consequently, the final decision of our system is derived from integrating the results of these two models, thereby achieving a more complete and accurate intrusion detection system.

To effectively merge these behaviors into a unified representation of smart home activity, we employ a learning structure that leverages backpropagation to optimize each classifier's weight in intrusion detection. This approach considers data from internal and external communications, which is analyzed across different time windows to capture variations in behavior over a day. Internal data is processed through the internal behavior detection model, while the external behavior detection model evaluates external data. Each model then outputs a probability value corresponding to the likelihood of a specific state or activity in that time window. These probability values are multiplied by their associated weights, and the model with the highest resulting probability is selected as the final classification outcome. The weights are calculated based on a backpropagation approach to find the optimum weight of each model. This approach ensures that each behavior contributes proportionally to the overall decision.

Given the extensive number of classes (approximately 100), we implement a hierarchical classification structure to improve accuracy and efficiency. In the first layer, the model identifies the primary category of activity (the main class), while the second layer determines the precise activity within that main class. This hierarchical approach utilizes the One-vs-Rest (OvR) technique, where binary classifiers are trained for each class, allowing the model to handle many classifications effectively. Each classifier outputs a probability value indicating the likelihood that the input data corresponds to a specific class. The OvR approach is particularly beneficial in this context, as it simplifies the classification task into manageable binary decisions, enhancing the robustness and interpretability of the overall system.

In the final stage, the probabilities from each classifier are assessed, and the system selects the class with the highest probability as the final output. If the probability score remains below a certain threshold, the input is flagged as a potential zero-day attack, warranting further investigation. This structure ensures the model remains adaptable to new, unknown threats while maintaining accuracy across known categories.

The general procedure is depicted in Fig. 34. In the following sections, we detail each component of our proposed solution, delving into the mathematical foundations and technical aspects that support the model's efficacy in identifying and responding to diverse malicious activities within smart home ecosystems.

3.2.1 Graph Creation

The graph creation phase forms the foundation of our intrusion detection model by providing a structured representation of the smart home environment as a dynamic, weighted directed graph $G(V,E,T_i)$ for each time window T_i. The graph G captures both the internal topology and the interaction patterns of the smart home devices, with vertices V representing individual devices and directed edges E denoting packet transfers between them. The concept of a time window T_i is crucial for the graph construction process, as it defines the period within which device communications are aggregated, thereby allowing for a time-segmented analysis of the smart home's network activity.

To model each device interaction, each directed edge $e \in E$ between vertices v_i and v_j carries a set of packet-level attributes, denoted as $P(e) = \{p_1, p_2, \ldots, p_n\}$. These attributes include essential metrics such as the timestamp $t(e)$, packet size $s(e)$, header size $h(e)$, and payload size $\rho(e)$, which provide an in-depth view of the communication

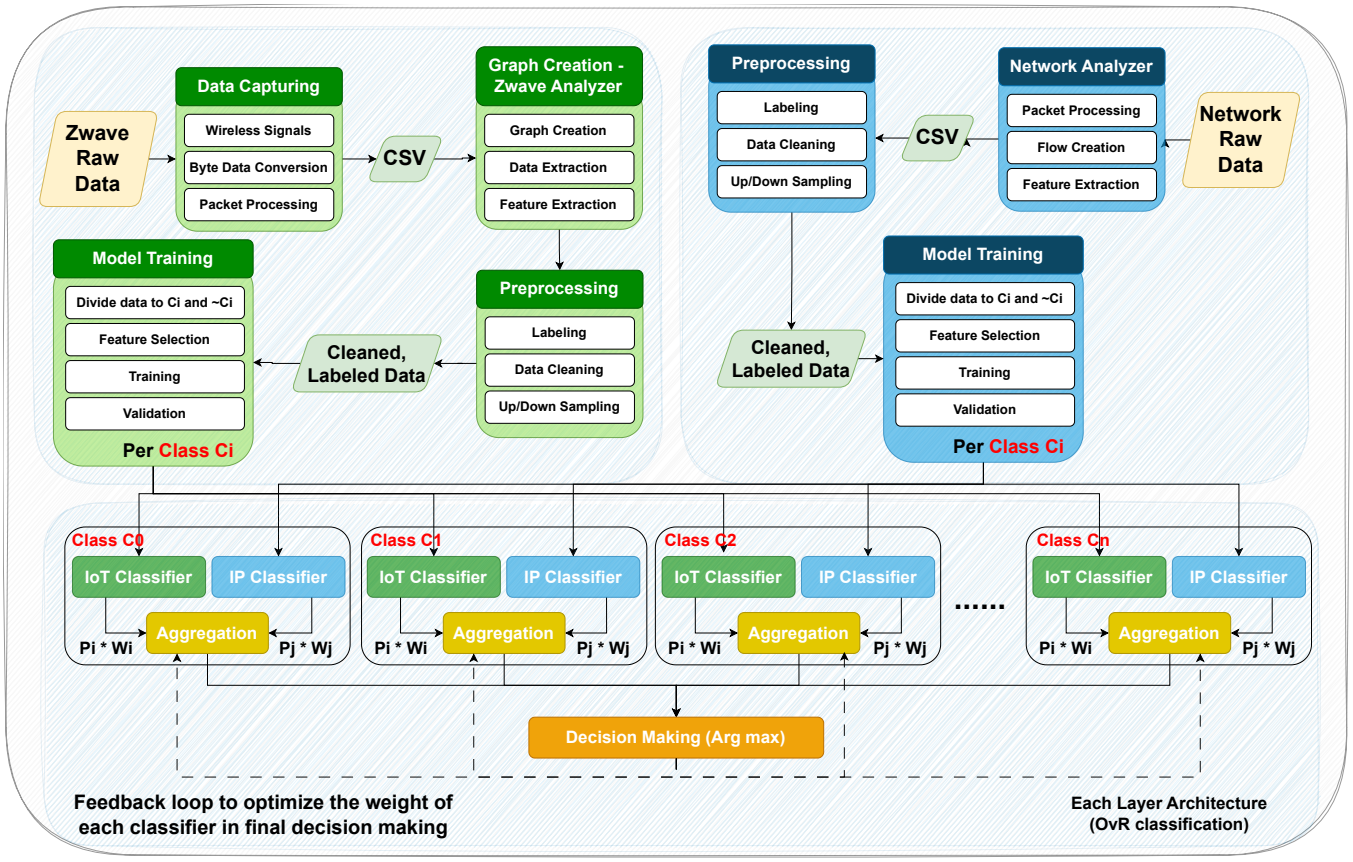


Figure 34: General architecture of the proposed model

taking place within the smart home. For clarity, we can express each edge mathematically as:

$$e_{ij} = (v_i, v_j, P(e)), \quad P(e) = \{t(e), s(e), h(e), \rho(e)\} \tag{1}$$

In addition, to ensure user privacy, the packet data remains encrypted, represented by an encryption indicator function $\varepsilon(e)$, where $\varepsilon(e) = 1$ if the packet data is encrypted and $\varepsilon(e) = 0$ otherwise. This design guarantees confidentiality without compromising the detection capabilities of the model.

To illustrate this construction, consider the scenario in Fig. 35, where a resident arrives home (the "come-to-home" scenario). This event activates a sequence of interactions across the smart home network, initiated by the Motion Sensor (M) detecting movement. As the system responds, devices such as the Smart Bulb (S), Door Lock (D), and Heater (H) communicate with the central Gateway/Hub (G) to manage system responses, including lighting adjustments, security protocol verification, and climate control. The dynamic network of interactions, represented within G, provides the system with a real-time map of device connectivity and packet flow, enabling deeper analysis of behavior patterns in the smart home.

The graph creation stage concludes with the establishment of G as a fully functional data structure capturing all necessary interactions and packet-level details within each time window T_i. With its encrypted attributes and comprehensive packet information, this graph sets the stage for subsequent data extraction and analysis in the intrusion detection process.

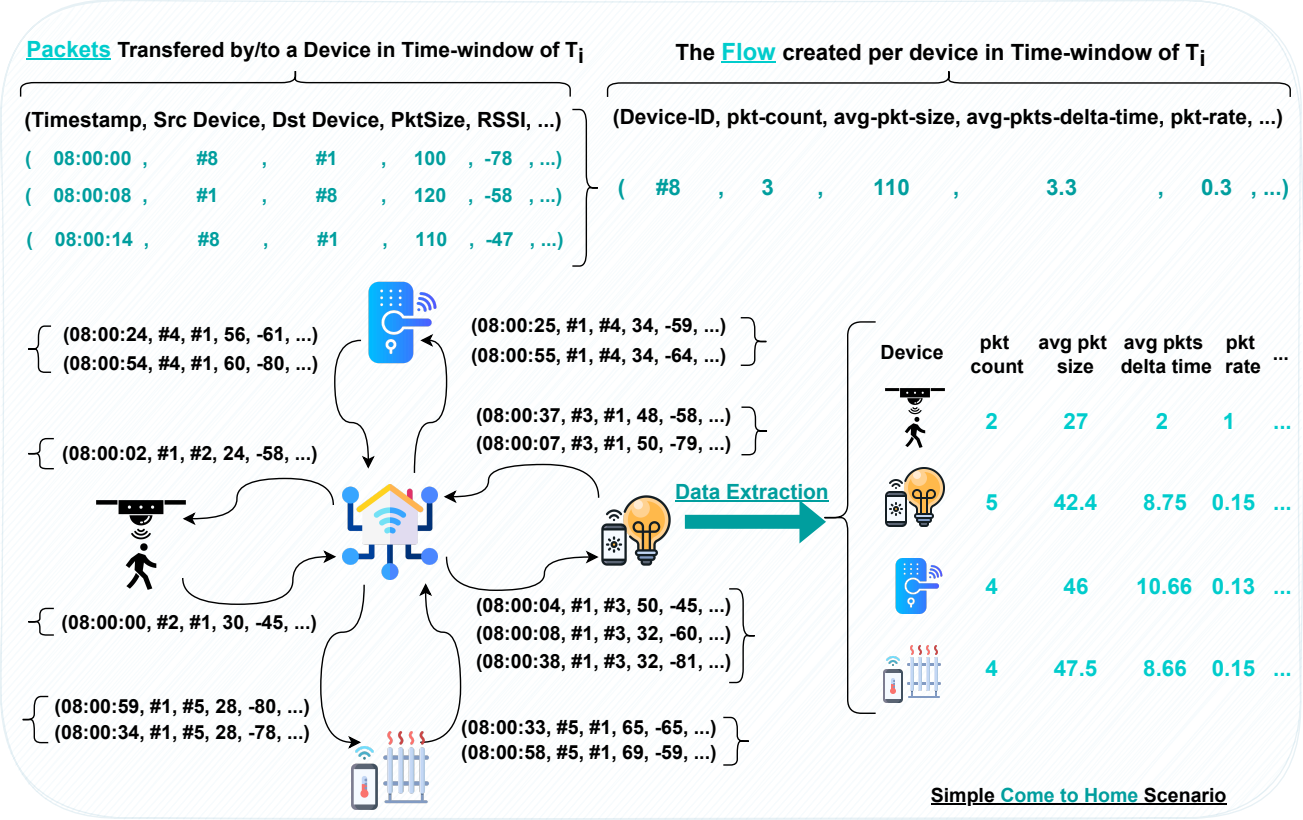


Figure 35: Example of IoT Smart Home graph

3.2.2 Data Extraction

The data extraction phase builds upon the established graph $G(V,E,T_i)$ by extracting relevant communication attributes for each device interaction, thereby enabling detailed insight into device behavior within the smart home. This step is critical to identifying potential malicious activities and patterns in network activity. The extraction mechanism is designed to capture data exchanged by each device and is aggregated within a list termed ***IoTFlow***, formulated in the following equation:

$$IoTFlow_{T_i} = [Flow_1, Flow_2, \ldots, Flow_n] \tag{2}$$

The list IoTFlow represents the collection of packet flow information for each device over a defined time window T_i, where each Flow_k is given by:

$$Flow_k = \{Device_k, f_1, f_2, \ldots, f_m\} \tag{3}$$

with $\mathbf{f}_k = (f_1, f_2, \ldots, f_m)$ representing the feature vector of $Flow_k$, where each $f_i \in \mathbb{R}$ corresponds to features such as packet size, header size, and arrival time. This formulation allows for a detailed and structured analysis of each device's communication behavior, providing insight into its interaction within the broader network context.

For each device d in G, the set of features $F(d)$ is derived by analyzing the incoming and outgoing edges over T_i. Let S_d denote the sequence of packets associated with device d in T_i, and let $|S_d|$ be the total number of packets. For every $e \in S_d$, the key statistics are extracted as follows:

$$f_1 = \sum_{e \in S_d} s(e), \quad f_2 = \sum_{e \in S_d} h(e), \quad f_3 = \frac{\sum_{e \in S_d} t(e)}{|S_d|} \tag{4}$$

The output IoTFlow for each device consolidates these features into a compact representation, capturing the flow characteristics of each device's interaction profile within T_i.

In the "come-to-home" scenario discussed earlier, the data extraction framework is applied to compile IoTFlow details representing each device's response. These values provide a continuous and time-relevant perspective on device interactions, offering a critical view of the smart home's operating state. The IoTFlow structure captures each device's distinct communication pattern, enabling anomaly detection algorithms to assess deviations from expected behavior, thus signaling potential intrusions or malfunctions.

The data extraction stage is crucial for linking the graph structure G with real-time behavioral analysis, setting the stage for effective intrusion detection. This phase ensures that each device's communication footprint is documented comprehensively, creating a robust dataset that feeds into the detection and classification processes of the overall intrusion detection model.

3.3 Smart Home Intrusion Detection Architecture

This section presents the theoretical basis and structured design of our two-layer intrusion detection system, which is tailored to classify network activity as benign or indicative of multiple specific attack types. Our system employs an ensemble-based method to enhance classification accuracy across distinct attack sub-categories.

The principal objective is to effectively combine two data sources, namely external internet network data (denoted as $\mathcal{D}_{\text{ip}}$) and internal IoT network data (denoted as $\mathcal{D}_{\text{iot}}$), to improve detection performance across varying attack scenarios. The system is structured to perform classification in two layers. The first layer categorizes each instance into one of several primary classes, where each class represents either benign traffic or a main attack category. In the second layer, the system refines the classification by identifying specific attack sub-categories within the detected attack type.

We propose a dual-tier model in which each data source undergoes independent classification using One-vs-Rest (OvR) classifiers to accomplish this. Let $\mathcal{C} = \{c_0, c_1, \ldots, c_k\}$ represent the set of $k+1$ primary classes, where c_0 denotes benign traffic and $\{c_1, c_2, \ldots, c_k\}$ represent k distinct attack types. In the first layer, two classifiers are constructed per class c_i, one classifier $f_{c_i}^{\text{ip}}$ using $\mathcal{D}_{\text{ip}}$ and another classifier $f_{c_i}^{\text{iot}}$ using $\mathcal{D}_{\text{iot}}$. Each classifier outputs a probability score $P(c_i \mid \mathcal{D}_{\text{ip}})$ or $P(c_i \mid \mathcal{D}_{\text{iot}})$, quantifying the likelihood that the input belongs to class c_i.

The final probability for each primary class c_i is determined by a weighted combination of these probabilities from the two data sources:

$$P(c_i) = w_{\text{ip}} \cdot P(c_i \mid \mathcal{D}_{\text{ip}}) + w_{\text{iot}} \cdot P(c_i \mid \mathcal{D}_{\text{iot}}) \tag{5}$$

where w_{ip} and w_{iot} are weights that balance the contributions of $\mathcal{D}_{\text{ip}}$ and $\mathcal{D}_{\text{iot}}$, respectively. The final prediction for the input instance is assigned to the class c with the maximum combined probability:

$$c = \arg\max_{c_i \in \mathcal{C}} P(c_i) \tag{6}$$

For example, when $|\mathcal{C}| = 5$, the system identifies one benign class c_0 and four distinct attack classes $\{c_1, c_2, c_3, c_4\}$. Each class employs OvR classification for each data type, ensuring that the strengths of each data source are utilized

optimally while minimizing the risk of over-reliance on any single classifier.

3.3.1 Data Structure and Preprocessing

To build a robust and layered detection system, it is crucial to define a consistent and well-structured dataset carefully segmented by data type and class. This section outlines the data structure and preprocessing steps, including independent features extracted from Internet Protocol (IP) and Internet of Things (IoT) sources. These are subsequently processed to optimize the classification process across multiple categories and sub-categories.
We denote the IP and IoT datasets as $\mathcal{D}^{\text{ip}} = \{(\mathbf{x}_i^{\text{ip}}, y_i)\}_{i=1}^{N^{\text{ip}}}$ and $\mathcal{D}^{\text{iot}} = \{(\mathbf{x}_i^{\text{iot}}, y_i)\}_{i=1}^{N^{\text{iot}}}$, where each element $(\mathbf{x}_i, y_i)$ consists of a feature vector and a corresponding class label. The classes are organized hierarchically: main categories C_j (e.g., benign, primary attack types) for $j \in \{1,2,\ldots,m\}$ represent overarching classes, while each main category C_j further branches into sub-categories S_{jk} for $k \in \{1,2,\ldots,n_j\}$, allowing classification of specific attack types or benign variants within each main category.

$$\mathcal{D}^{\text{ip}} = \{(\mathbf{x}_i^{\text{ip}}, y_i)\}_{i=1}^{N^{\text{ip}}} \quad \text{(IP data)}, \tag{7}$$

$$\mathcal{D}^{\text{iot}} = \{(\mathbf{x}_i^{\text{iot}}, y_i)\}_{i=1}^{N^{\text{iot}}} \quad \text{(IoT data)}. \tag{8}$$

Feature vectors $\mathbf{x}_i^{\text{ip}}$ and $\mathbf{x}_i^{\text{iot}}$ represent the extracted data points within each dataset, structured as follows:

$$\mathbf{x}_i^{\text{ip}} = [x_{i1}^{\text{ip}}, x_{i2}^{\text{ip}}, \ldots, x_{id}^{\text{ip}}] \in \mathbb{R}^d, \tag{9}$$

$$\mathbf{x}_i^{\text{iot}} = [x_{i1}^{\text{iot}}, x_{i2}^{\text{iot}}, \ldots, x_{iq}^{\text{iot}}] \in \mathbb{R}^q. \tag{10}$$

To facilitate synchronized data analysis across these sources, we segment both datasets into non-overlapping time windows of length Δt, ensuring that classification processes benefit from an aligned temporal frame across IP and IoT data streams. This approach enables detailed and time-sensitive classification, ensuring that temporal behaviors and anomalies are accurately captured within each time frame.
Preprocessing: A preprocessing pipeline is essential to prepare the data for classification. Firstly, normalization is applied to standardize feature values across datasets. Each feature vector $\mathbf{x}_i$ is transformed into a normalized vector $\mathbf{x}_i^{\text{norm}}$ defined as:

$$\mathbf{x}_i^{\text{norm}} = \frac{\mathbf{x}_i - \mu}{\sigma}, \tag{11}$$

where μ and σ are the mean and standard deviation, respectively, calculated over the entire dataset:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i, \tag{12}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}_i - \mu)^2}. \tag{13}$$

This normalization process ensures consistency in feature magnitudes, thereby enhancing the performance of clas-

sification algorithms by aligning features from IP and IoT data sources onto a common scale.
Furthermore, categorical features within the datasets are encoded using One-Hot Encoding to transform discrete variables into binary vectors. For each categorical feature x_{ij}, the encoding is defined as follows:

$$x_{ij}^{\text{encoded}} = \begin{cases} 1, & \text{if feature } j \text{ is associated with category } i, \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

This encoding enables the incorporation of categorical information within the numeric-based classification framework, thereby enhancing the classifier's ability to discern between categorical features within each sub-category.
Finally, to address potential class imbalances common in intrusion detection scenarios, we utilize the Synthetic Minority Over-sampling Technique (SMOTE). This technique balances the dataset by generating synthetic samples for underrepresented classes, enriching the dataset with minority-class samples. The balanced dataset $\mathcal{D}_{\text{balanced}}$ is formally represented as:

$$\mathcal{D}_{\text{balanced}} = \mathcal{D}^{\text{ip}} \cup \mathcal{D}^{\text{iot}} \cup \{(\mathbf{x}_j^{\text{new}}, y_j) \mid y_j \text{ belongs to a minority class}\}, \tag{15}$$

where synthetic instances $\mathbf{x}_j^{\text{new}}$ are generated by linearly interpolating between a sample $\mathbf{x}_i$ and its nearest neighbor $\mathbf{x}_k$, using a random scalar δ:

$$\mathbf{x}_j^{\text{new}} = \mathbf{x}_i + \delta \cdot (\mathbf{x}_i - \mathbf{x}_k), \quad \delta \sim U(0,1). \tag{16}$$

Through these preprocessing steps, we create a structured, balanced, and harmonized dataset that optimally supports the dual-layered classification architecture. The resulting dataset ensures that each category and sub-category is represented, enhancing the classifier's ability to identify benign and malicious traffic with granularity across primary and secondary classification stages.

3.3.2 Layer 1 - Main Category Classification

A One-vs-Rest classification approach is implemented in the first layer of the intrusion detection system to classify data into the main categories effectively. Given the complexity of distinguishing between regular and malicious activity within IP and IoT traffic, distinct classifiers for each data type, which aim to capitalize on their unique characteristics, are utilized. Each main category C_j is represented by two dedicated Random Forest (RF) classifiers, one for IP data and one for IoT data, noted as h_j^{ip} and h_j^{iot}, respectively. The binary indicator y_j^{m} for each classifier reflects whether an input instance belongs to a specific category, establishing a reliable initial categorization that enables refined analysis in subsequent layers.
Here, the RF algorithm has been chosen for several key reasons that align with the intrusion detection requirements and the data characteristics. First, RF models excel in handling high-dimensional datasets and can inherently capture complex non-linear relationships due to the ensemble approach of combining multiple decision trees. In this system, IP and IoT data often exhibit distinct statistical patterns; RF's decision tree ensemble structure allows it to adapt flexibly to these differences without explicit feature engineering. Additionally, RF classifiers resist overfitting, essential in network traffic analysis where anomalous patterns may occur infrequently but must still be detected accurately. The bootstrapping technique and feature randomness in RF training further enhance its robustness by reducing variance and yielding reliable predictions even in heterogeneous data environments. Another

advantage of RF is its ability to provide probabilistic class outputs, which we leverage for ensemble probability calculations. The combined strengths of RF classifiers create a robust foundation for the classification task in the first layer, supporting the intricate requirements of intrusion detection across varied data types.

For each time window $\mathcal{T}$, the RF classifiers operate on separate datasets for IP and IoT data, denoted as $\mathcal{T}^{\text{ip}} = \{(\mathbf{x}_i^{\text{ip}}, y_i)\}_{i=1}^{N^{\text{ip}}}$ and $\mathcal{T}^{\text{iot}} = \{(\mathbf{x}_i^{\text{iot}}, y_i)\}_{i=1}^{N^{\text{iot}}}$. Each classifier computes a class probability $P(C_j|\mathbf{x}_i^m)$ for each main category C_j, producing an estimate of the likelihood that an instance belongs to that category. This probability-based output enables a structured approach to further classification, wherein initial results contribute to a probabilistic foundation for deeper categorization in subsequent layers.

The classification results from the IP and IoT classifiers are integrated by combining their probabilistic outputs through a weighted ensemble model. Let $P(C_j|\mathcal{T}^{\text{ip}})$ and $P(C_j|\mathcal{T}^{\text{iot}})$ represent the probabilities assigned by the IP and IoT classifiers, respectively. The final probability for each main category C_j, denoted as $P_{\text{ensemble}}(C_j|\mathcal{T}_k)$, is then calculated as a weighted combination of these two probabilities:

$$P_{\text{ensemble}}(C_j|\mathcal{T}_k) = \alpha_j^{\text{ip}} \cdot P(C_j|\mathcal{T}^{\text{ip}}) + \alpha_j^{\text{iot}} \cdot P(C_j|\mathcal{T}^{\text{iot}}), \tag{17}$$

where α_j^{ip} and α_j^{iot} are weight parameters optimized to reflect the importance of IP and IoT data for each category. The weighting parameters are determined through an optimization procedure that iteratively adjusts them based on classification performance, ensuring that each data source contributes optimally to the overall ensemble probability.

3.3.3 Optimization of Weight Parameters

An Advanced Composite Weighted Quadratic Cross-Entropy Loss with Gradient Penalty and Elastic Net Regularization is employed to refine the ensemble model's weight parameters. This intricate loss function, labeled $\mathcal{L}_{\text{advanced}}$, is formulated to achieve optimal predictive accuracy and robust generalization by blending multiple components, including cross-entropy loss, quadratic penalties, gradient penalties, and Elastic Net regularization. These components work harmoniously to balance the ensemble model's outputs' accuracy, smoothness, and interpretability.

The advanced loss function $\mathcal{L}_{\text{advanced}}$ is defined as:

$$\mathcal{L}_{\text{advanced}}(y, \hat{y}) = -\sum_{k=1}^{K} \left(y_k \cdot \log(\hat{y}_k) + \gamma_k \cdot (y_k - \hat{y}_k)^2 + \eta_k \cdot \nabla \hat{y}_k \right) + \lambda_1 \sum_{j=1}^{m} |\alpha_j^{\text{ip}} - \alpha_j^{\text{iot}}| + \lambda_2 \sum_{j=1}^{m} (\alpha_j^{\text{ip}} - \alpha_j^{\text{iot}})^2, \tag{18}$$

where y_k and $\hat{y}_k$ represent the true and predicted probabilities for class C_k, respectively. Each term within the loss function plays a specific role in improving the ensemble model's performance. The parameter γ_k scales the quadratic loss term, intensifying penalties for larger deviations between y_k and $\hat{y}_k$. Meanwhile, η_k adjusts the gradient penalty term $\nabla \hat{y}_k$, encouraging smooth transitions in predictions by penalizing substantial gradient variations. The Elastic Net regularization terms, governed by λ_1 and λ_2, balance sparsity and smoothness in weight distribution.

Each component of the advanced loss function is designed to address specific prediction and regularization needs. Expanding the Quadratic Loss Term $\gamma_k \cdot (y_k - \hat{y}_k)^2$, we obtain:

$$\gamma_k \cdot (y_k - \hat{y}_k)^2 = \gamma_k \cdot \left(y_k^2 - 2 y_k \hat{y}_k + \hat{y}_k^2 \right). \tag{19}$$

This decomposition reveals individual contributions from the true class probability y_k, the predicted class proba-

bility $\hat{y}_k$, and their interaction term. The quadratic expansion effectively enhances model robustness, particularly against outliers, by allocating a proportionally higher penalty to larger errors, thereby helping to stabilize predictions across varied data points.

Similarly, the Gradient Penalty Term $\eta_k \cdot \nabla \hat{y}_k$ penalizes abrupt prediction changes:

$$\eta_k \cdot \nabla \hat{y}_k = \eta_k \cdot \frac{d\hat{y}_k}{dx}, \tag{20}$$

where $\nabla \hat{y}_k$ represents the gradient of the predicted probability $\hat{y}_k$ with respect to input x. By summing this term across all classes K, we introduce a smoothing effect that regularizes against sharp transitions in predicted values, aiding in continuous and smooth output probabilities. The gradient penalty reduces model overfitting, especially in high-variance or noisy data distributions.

Lastly, Elastic Net Regularization blends L1 and L2 regularization to stabilize the model's complexity:

$$\lambda_1 \sum_{j=1}^{m} |\alpha_j^{\text{ip}} - \alpha_j^{\text{iot}}| + \lambda_2 \sum_{j=1}^{m} (\alpha_j^{\text{ip}} - \alpha_j^{\text{iot}})^2. \tag{21}$$

The L1 term $\lambda_1 \sum_{j=1}^{m} |\alpha_j^{\text{ip}} - \alpha_j^{\text{iot}}|$ introduces sparsity by encouraging the model to set certain weights to zero, favoring simpler models with fewer non-zero parameters. In contrast, the L2 term $\lambda_2 \sum_{j=1}^{m} (\alpha_j^{\text{ip}} - \alpha_j^{\text{iot}})^2$ promotes weight continuity which allows for gradual changes across similar classes or features. Together, these terms offer a balance between sparsity and continuity, enhancing the model's generalization capacity.

The optimal ensemble weights α_j^{ip} and α_j^{iot} are determined by minimizing $\mathcal{L}_{\text{advanced}}$ under the constraint $\alpha_j^{\text{ip}} + \alpha_j^{\text{iot}} = 1$. The optimization problem, formulated over the training distribution $\mathcal{D}$, is expressed as:

$$(\alpha_j^{\text{ip}}, \alpha_j^{\text{iot}}) = \arg \min_{\alpha_j^{\text{ip}} + \alpha_j^{\text{iot}} = 1} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathcal{L}_{\text{advanced}} \left(y, \alpha_j^{\text{ip}} \cdot h_j^{\text{ip}}(x) + \alpha_j^{\text{iot}} \cdot h_j^{\text{iot}}(x) \right) \right]. \tag{22}$$

To compute the gradients for optimization, we derive partial derivatives of $\mathcal{L}_{\text{advanced}}$ concerning α_j^{ip} and α_j^{iot}. For α_j^{ip}, the partial derivative is:

$$\frac{\partial \mathcal{L}_{\text{advanced}}}{\partial \alpha_j^{\text{ip}}} = \sum_{k=1}^{K} \left(h_j^{\text{ip}}(x) - h_j^{\text{iot}}(x) \right) \left(\gamma_k \cdot (y_k - \hat{y}_k) + \eta_k \cdot \nabla h_j^{\text{ip}}(x) \right) + \lambda_1 \cdot \text{sign}(\alpha_j^{\text{ip}} - \alpha_j^{\text{iot}}) + 2\lambda_2 (\alpha_j^{\text{ip}} - \alpha_j^{\text{iot}}). \tag{23}$$

which simplifies to:

$$= \sum_{k=1}^{K} \left[\gamma_k \left(h_j^{\text{ip}}(x) - h_j^{\text{iot}}(x) \right) (y_k - \hat{y}_k) + \eta_k \left(h_j^{\text{ip}}(x) - h_j^{\text{iot}}(x) \right) \nabla h_j^{\text{ip}}(x) \right] + \lambda_1 \cdot \text{sign}(\alpha_j^{\text{ip}} - \alpha_j^{\text{iot}}) + 2\lambda_2 \cdot (\alpha_j^{\text{ip}} - \alpha_j^{\text{iot}}) \tag{24}$$

and finally it would be:

$$= \sum_{k=1}^{K} \left[\gamma_k \left(h_j^{\text{ip}}(x) - h_j^{\text{iot}}(x) \right) (y_k - \hat{y}_k) \right] + \sum_{k=1}^{K} \left[\eta_k \left(h_j^{\text{ip}}(x) - h_j^{\text{iot}}(x) \right) \nabla h_j^{\text{ip}}(x) \right] + \lambda_1 \cdot \text{sign}(\alpha_j^{\text{ip}} - \alpha_j^{\text{iot}}) + 2\lambda_2 \cdot (\alpha_j^{\text{ip}} - \alpha_j^{\text{iot}}) \tag{25}$$

Each term here incorporates contributions from the predictions $h_j^{\text{ip}}(x)$ and $h_j^{\text{iot}}(x)$, the misclassification penalty

scaled by γ_k, the gradient penalty, and the Elastic Net regularization terms. The $\text{sign}(\alpha_j^{\text{ip}} - \alpha_j^{\text{iot}})$ term ensures sparsity by directing the gradient descent path, while $2\lambda_2(\alpha_j^{\text{ip}} - \alpha_j^{\text{iot}})$ promotes smooth weight adjustments for generalization.

For α_j^{iot}, the partial derivative similarly becomes:

$$\frac{\partial \mathcal{L}_{\text{advanced}}}{\partial \alpha_j^{\text{iot}}} = \sum_{k=1}^{K} \left(h_j^{\text{iot}}(x) - h_j^{\text{ip}}(x)\right) \left(\gamma_k \cdot (y_k - \hat{y}_k) + \eta_k \cdot \nabla h_j^{\text{iot}}(x)\right) + \lambda_1 \cdot \text{sign}(\alpha_j^{\text{iot}} - \alpha_j^{\text{ip}}) + 2\lambda_2(\alpha_j^{\text{iot}} - \alpha_j^{\text{ip}}). \quad (26)$$

which simplifies to:

$$= \sum_{k=1}^{K} \left[\gamma_k \left(h_j^{\text{iot}}(x) - h_j^{\text{ip}}(x)\right) (y_k - \hat{y}_k) + \eta_k \left(h_j^{\text{iot}}(x) - h_j^{\text{ip}}(x)\right) \nabla h_j^{\text{iot}}(x)\right] + \lambda_1 \cdot \text{sign}(\alpha_j^{\text{iot}} - \alpha_j^{\text{ip}}) + 2\lambda_2 \cdot (\alpha_j^{\text{iot}} - \alpha_j^{\text{ip}}) \quad (27)$$

and ultimately it would be:

$$= \sum_{k=1}^{K} \left[\gamma_k \left(h_j^{\text{iot}}(x) - h_j^{\text{ip}}(x)\right) (y_k - \hat{y}_k)\right] + \sum_{k=1}^{K} \left[\eta_k \left(h_j^{\text{iot}}(x) - h_j^{\text{ip}}(x)\right) \nabla h_j^{\text{iot}}(x)\right] + \lambda_1 \cdot \text{sign}(\alpha_j^{\text{iot}} - \alpha_j^{\text{ip}}) + 2\lambda_2 \cdot (\alpha_j^{\text{iot}} - \alpha_j^{\text{ip}}) \quad (28)$$

This derivative similarly accounts for model differences, penalized misclassification, gradient adjustments, and regularization effects, ensuring stability and optimal balance in the ensemble model's output through a fine-grained adjustment of weights.

3.3.4 Multi-Dimensional Threshold-Based Zero-Day Detection

In advanced intrusion detection systems (IDS), zero-day detection requires highly adaptive and precise mechanisms. A multi-dimensional thresholding mechanism that incorporates two types of thresholds, static and dynamic, is introduced to detect known and novel attack patterns. These two thresholds are designed to complement each other. The static threshold is pre-determined during training, while the dynamic threshold adjusts based on the observed divergence between predicted and prior distributions during inference. This dual-threshold approach ensures the system can accurately classify known attacks and flag new and unseen threats as zero-day anomalies. The interaction of these two thresholds is crucial for maintaining a balance between sensitivity to unknown threats and the stability of known class boundaries.

Static Threshold: The static threshold is integral to the training phase. During training, the model learns to classify observations into known classes (e.g., legitimate network traffic or known attack types). The static threshold is used to define the decision boundary for these classes. Importantly, zero-day anomalies are not explicitly included in the static training data; rather, the static threshold is learned based on the class distributions from the labeled training set, which only includes known categories.

To formalize this, let's denote the static threshold as a vector $\theta = [\theta_1, \theta_2, \ldots, \theta_m]$, where θ_j corresponds to the threshold for the predicted class C_j. The goal is to learn these thresholds so that the model can accurately classify observations into known categories while maintaining a margin of safety against misclassifications.

We approach this by training the model with a cross-entropy loss function, which measures the divergence between the predicted class probabilities and the true class labels. The cross-entropy loss function can be written as:

$$\mathcal{L}_{\text{static}} = -\sum_{j=1}^{m} P_{\text{true}}(C_j) \log(P_{\text{model}}(C_j)) + \lambda \sum_{j=1}^{m} \theta_j^2 \tag{29}$$

where $P_{\text{true}}(C_j)$ is the true probability distribution over the classes, $P_{\text{model}}(C_j)$ is the predicted probability for each class, and λ is a regularization parameter. This loss function is optimized to minimize the error between predicted and true class labels while penalizing excessively large threshold values to prevent overfitting.

The learning process aims to ensure that the thresholds θ_j for each class are fine-tuned to reflect the boundaries of known attack categories. Importantly, this training does not explicitly include zero-day anomalies in the training set. Therefore, during inference, if an observation's predicted probability for any class C_j is below its corresponding threshold θ_j, the model classifies the observation as a Zero-Day anomaly, signaling that it does not belong to any known category.

The final decision rule based on the static threshold can be written as:

$$y_{\text{main}} = \begin{cases} C_j, & \text{if } \max_j P_{\text{ensemble}}(C_j|\mathcal{T}_k) \geq \theta_j \\ \text{Zero-Day}, & \text{otherwise} \end{cases} \tag{30}$$

Here, $P_{\text{ensemble}}(C_j|\mathcal{T}_k)$ is the probability of class C_j for the input observation $\mathcal{T}_k$. The observation is classified as Zero-Day if the maximum predicted probability falls below the corresponding threshold θ_j for every class.

Dynamic Threshold: In addition to the static threshold, the dynamic threshold is introduced to handle the uncertainty and variability of the predictions, especially when the model encounters observations that deviate from the learned class distributions. The dynamic threshold dynamically adjusts based on how much the model's prediction diverges from its expectations. This is particularly useful for detecting zero-day anomalies, which are not present in the training data but exhibit behavior that significantly differs from known classes.

To measure this divergence, we utilize the Kullback-Leibler (KL) divergence, a measure of the difference between two probability distributions [75]. In this context, we compute the KL divergence between the predicted probability distribution $P_{\text{ensemble}}(C_j|\mathcal{T}_k)$ for an observation $\mathcal{T}_k$ and the prior class distribution $P_{\text{model}}(C_j)$, which represents the model's expectations from historical data. The dynamic threshold is determined by minimizing this divergence, which can be expressed as:

$$\theta_{\text{KL}} = \arg\min_{\theta_{\text{KL}}} D_{KL}(P_{\text{ensemble}}(C_j|\mathcal{T}_k) \parallel P_{\text{model}}(C_j)) \tag{31}$$

where the KL divergence is given by:

$$D_{KL}(P_{\text{ensemble}}(C_j|\mathcal{T}_k) \parallel P_{\text{model}}(C_j)) = \sum_{k=1}^{K} P_{\text{ensemble}}(C_j|\mathcal{T}_k) \log\left(\frac{P_{\text{ensemble}}(C_j|\mathcal{T}_k)}{P_{\text{model}}(C_j)}\right) \tag{32}$$

The prior class distribution $P_{\text{model}}(C_j)$ is estimated based on the historical data, and it reflects the expected probability of each class occurring before observing the current data. The KL divergence quantifies the deviation between the predicted class probabilities and this prior distribution. A large divergence suggests that the observation is inconsistent with the model's expectations, which signals a potential Zero-Day anomaly.

Minimizing this divergence, we adaptively adjust the decision boundary to account for significant deviations from known behavior, which is crucial for detecting novel attack patterns. The observation is flagged as an anomaly if the KL divergence exceeds a certain threshold.

Once both thresholds are calculated, the system must combine them to make the final classification decision. The dynamic threshold is unnecessary if the static threshold indicates a zero-day anomaly (i.e., the maximum predicted probability falls below the static threshold for all classes). However, if the static threshold does not indicate a zero-day anomaly, the dynamic threshold is used to evaluate the divergence from the prior distribution further. The final decision rule is:

$$y_{\text{final}} = \begin{cases} C_j, & \text{if } \max_j P_{\text{ensemble}}(C_j|\mathcal{T}_k) \geq \theta_j \text{ and } D_{KL} \leq \gamma \\ \text{Zero-Day}, & \text{if either of the conditions above is not met} \end{cases} \tag{33}$$

Where γ is a threshold for the KL divergence, determining the acceptable level of deviation. If the KL divergence exceeds this threshold or the static threshold flags a zero-day anomaly, the observation is classified as a Zero-Day attack.

In summary, the two-tier thresholding approach effectively combines static and dynamic thresholds to ensure robust and adaptable detection of zero-day anomalies. The static threshold is calculated during training using a cross-entropy loss function, which ensures that known attack classes are properly classified. This threshold provides a fixed decision boundary for classifying known patterns. Conversely, the dynamic threshold adapts to the observed divergence between predicted and prior distributions, allowing the system to detect previously unseen attacks by flagging significant deviations. Together, these thresholds enable the system to accurately classify known threats while maintaining high sensitivity to novel, zero-day attacks, thus providing an efficient and robust defense mechanism.

3.3.5 Layer 2 - Sub-Category Classification

The second layer of the intrusion detection framework extends the analysis of the main categories identified in Layer 1, refining the classification into specific sub-categories S_{jk} within each main category C_j. A set of dedicated RF classifiers is applied in this layer, targeting these sub-categories to provide a more granular breakdown of detected anomalies. Each main category C_j has its corresponding sub-categories classified through RF models, each tailored to the characteristics of both IP and IoT data and therefore ensuring a detailed evaluation across different types of traffic.

For each main category C_j, we define the subset of data points, denoted as $\mathcal{D}_j^{\text{sub}}$, which includes the features $\mathbf{x}_i^{\text{ip}}$ and $\mathbf{x}_i^{\text{iot}}$ and corresponding sub-category labels y_{ik}, where i belongs to both IP and IoT datasets ($\mathcal{T}^{\text{ip}}$ and $\mathcal{T}^{\text{iot}}$) and aligns with the specific sub-category S_{jk} under evaluation. Thus, $\mathcal{D}_j^{\text{sub}} = \{(\mathbf{x}_i^{\text{ip}}, y_{ik})\}_{i \in \mathcal{T}^{\text{ip}} \cap \mathcal{T}^{\text{iot}}}$, ensuring that each RF classifier operates on data instances specifically filtered for sub-category classification within C_j.

A weighted probability model is utilized to compute the likelihood of an instance falling into a sub-category. The probability for each sub-category S_{jk} is calculated as follows:

$$P(S_{jk}|\mathcal{T}) = \beta_{jk}^{\text{ip}} \cdot P(S_{jk}|\mathcal{T}^{\text{ip}}) + \beta_{jk}^{\text{iot}} \cdot P(S_{jk}|\mathcal{T}^{\text{iot}}) \tag{34}$$

where β_{jk}^{ip} and β_{jk}^{iot} are weighting parameters that balance the contributions from the IP and IoT classifiers, respectively, for sub-category S_{jk}. These parameters are tuned through a learning approach similar to the first layer that optimizes the cross-entropy loss function and thereby finding the ideal balance of IP and IoT data contributions specific to each sub-category. Additionally, zero-day attack detection in this layer follows a similar approach to Layer 1, wherein anomalies that deviate from sub-category profiles are flagged for further analysis.

3.4 Final Decision-Making Process

The final decision-making process is structured to synthesize the outputs from both classification layers, yielding a single, cohesive output that specifies the main threat category or sub-category or flags a zero-day anomaly if detected. This layered approach ensures the system makes accurate and contextually aware decisions, facilitating effective threat classification and anomaly detection across various network types.

At the core of this decision-making framework lies a hierarchical evaluation that first identifies the main threat category and, if recognized as a known category, subsequently evaluates the corresponding sub-category. Formally, let $P_{\text{ensemble}}(C_j|\mathcal{T}_k)$ represent the probability of each main category C_j in the first layer, and $P(S_{jk}|\mathcal{T}_k)$ denote the probability of each sub-category S_{jk} under the identified main category C_j in the second layer. By sequentially analyzing both probabilities, the model can deliver an output that reflects the full scope of category recognition.

The model's decision rule y_{final} integrates both layers by considering three potential cases: (1) known main and sub-category classifications, (2) known main category with an unknown (zero-day) sub-category, and (3) complete zero-day detection where no known main category is identified. This structured approach is represented mathematically as follows:

$$y_{\text{final}} = \begin{cases} (C_j, S_{jk}), & \text{if } \max_j P_{\text{ensemble}}(C_j|\mathcal{T}_k) \geq \theta_j \text{ and } \max_k P(S_{jk}|\mathcal{T}_k) \geq \delta_{jk} \\ (C_j, \text{Zero-Day Sub-Category}), & \text{if } \max_j P_{\text{ensemble}}(C_j|\mathcal{T}_k) \geq \theta_j \text{ and } \max_k P(S_{jk}|\mathcal{T}_k) < \delta_{jk} \\ \text{Zero-Day Main Category}, & \text{if } \max_j P_{\text{ensemble}}(C_j|\mathcal{T}_k) < \theta_j \end{cases} \tag{35}$$

where θ_j is the threshold for each main category, and δ_{jk} is the threshold within each sub-category. Through this unified framework, the system accurately captures the known threat categories, identifies partial zero-day anomalies when sub-categories are unrecognized, and detects complete zero-day threats when main categories are unknown. The overall result is a highly versatile classification model capable of effectively addressing known and emerging threats.

In conclusion, integrating these classification layers through a single, hierarchical decision-making process enables precise threat recognition across mixed-network data. The model's design captures the distinct features of each threat type, allowing it to categorize a wide range of threat behaviors. Consequently, the final decision-making framework enhances the model's applicability to dynamic network environments, providing robust security measures for traditional and IoT network traffic.

3.5 Loss of the Multi-Layer System

The performance of a complex multi-layer classification system hinges on an intricate interplay between its layers. This hierarchical classification introduces an inherent dependency between the layers. Thus the loss function must account for not only the error of each layer independently but also the cascading effect of errors from the first layer to the second. Furthermore, the system must handle the detection of zero-day anomalies, which require a separate loss term to measure the failure to classify a novel input.

This section provides a mathematically rigorous framework for the complete loss function, emphasizing the conditional nature of errors between the two layers. We will derive an elaborate loss function that integrates classification losses, conditional dependencies, regularization terms, and the response to unseen anomalies.

To begin, let us define the layers of the model. The first layer classifies an input $\mathbf{x}$ into one of the known main categories $C_j \in \{C_1, C_2, \ldots, C_m\}$ or a zero-day class, denoted $C_{\text{zero-day}}$. Mathematically, this can be represented as:

$$y_{\text{main}} = \text{argmax}_{C_j \in \{C_1, C_2, \ldots, C_m, C_{\text{zero-day}}\}} P(C_j|\mathbf{x}) \tag{36}$$

The second layer, conditioned on the correct classification of the first layer, refines the classification by determining a sub-category S_{jk} under the identified main category C_j. If the first layer misclassifies the main category, the second layer is expected to make a less reliable decision. Let y_{sub} represent the final sub-category prediction, conditional on C_j. We can express the dependency between the layers using a conditional probability:

$$y_{\text{sub}} = \text{argmax}_{S_{jk} \in \{S_{j1}, S_{j2}, \ldots, S_{jK_j}\}} P(S_{jk}|C_j, \mathbf{x}) \tag{37}$$

The zero-day classification is modeled independently but can affect the subsequent layers if the main category is misclassified as zero-day. Therefore, the loss function must account for the interaction between the main classification and the sub-category and the zero-day classification.

Let $\mathcal{L}$ represent the complete loss function. The system loss is a combination of three main terms: the loss from the first layer $\mathcal{L}_{\text{main}}$, the loss from the second layer $\mathcal{L}_{\text{sub}}$, and the loss for zero-day detection $\mathcal{L}_{\text{zero-day}}$. The first layer loss is computed based on cross-entropy, the second layer loss is computed conditionally based on the outcome of the first layer, and the zero-day loss applies when the first layer detects an anomaly.

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{\text{main}} + \beta \cdot \mathcal{L}_{\text{sub}} \cdot P_{\text{correct}}(y_{\text{main}}) + \gamma \cdot \mathcal{L}_{\text{zero-day}} \cdot P_{\text{correct}}(y_{\text{main}}) \tag{38}$$

The first layer's loss is computed using the cross-entropy function. We denote the predicted probability for category C_j as $P_{\text{main}}(C_j|\mathbf{x})$. The target probability is represented by the one-hot encoded vector $y_{\text{main}}(C_j)$, where $y_{\text{main}}(C_j) = 1$ if the true class is C_j, and $y_{\text{main}}(C_j) = 0$ otherwise. The loss for the first layer, $\mathcal{L}_{\text{main}}$, is then:

$$\mathcal{L}_{\text{main}} = -\sum_{j=1}^{m} y_{\text{main}}(C_j) \cdot \log P_{\text{main}}(C_j|\mathbf{x}) - y_{\text{main}}(C_{\text{zero-day}}) \cdot \log P_{\text{main}}(C_{\text{zero-day}}|\mathbf{x}) \tag{39}$$

For each known class C_j, the loss penalizes the negative log of the predicted probability, and for the zero-day class, it similarly penalizes the incorrect classification probability. However, to incorporate the dependency structure, we introduce a conditional loss correction term λ_{main}, which adjusts the impact of the first layer's loss on the second layer's performance.

$$\mathcal{L}_{\text{main}}^{\text{adjusted}} = \mathcal{L}_{\text{main}} + \lambda_{\text{main}} \cdot \sum_{j=1}^{m} P_{\text{main}}(C_j|\mathbf{x}) \cdot (1 - P_{\text{main}}(C_j|\mathbf{x})) \tag{40}$$

This term accounts for the uncertainty and adjusts for confidence in the first layer's prediction, introducing a correction that scales the main category loss by the uncertainty.

The second layer's loss $\mathcal{L}_{\text{sub}}$ measures the difference between the predicted sub-category S_{jk} and the true sub-category. Since the second layer is conditional on the first layer's correct classification, we express it as:

$$\mathcal{L}_{\text{sub}} = -\sum_{k=1}^{K_j} y_{\text{sub}}(S_{jk}) \cdot \log P_{\text{sub}}(S_{jk}|C_j, \mathbf{x}) \cdot P_{\text{correct}}(y_{\text{main}}) \tag{41}$$

The factor $P_{\text{correct}}(y_{\text{main}})$ is a weight that reduces the influence of the second layer's loss when the first layer's prediction is uncertain or incorrect. This term ensures that the second layer only contributes significantly to the loss when the main category classification is correct. The second layer's output also needs a regularization term

λ_{sub}, which penalizes overfitting and encourages smoothness in the sub-category predictions:

$$\mathcal{L}_{\text{sub}}^{\text{regularized}} = \mathcal{L}_{\text{sub}} + \lambda_{\text{sub}} \cdot \sum_{k=1}^{K_j} P_{\text{sub}}(S_{jk}|\mathbf{x}) \cdot \left(1 - P_{\text{sub}}(S_{jk}|\mathbf{x})\right) \tag{42}$$

The zero-day loss function is structured similarly to the first layer's loss function but emphasizes the model's ability to detect previously unseen categories. For zero-day detection, the loss is computed as:

$$\mathcal{L}_{\text{zero-day}} = -y_{\text{zero-day}} \cdot \log P_{\text{main}}(C_{\text{zero-day}}|\mathbf{x}) - (1 - y_{\text{zero-day}}) \cdot \log(1 - P_{\text{main}}(C_{\text{zero-day}}|\mathbf{x})) \tag{43}$$

To incorporate the cascading error structure, we add a penalty term $\lambda_{\text{zero-day}}$ that increases the loss when the first layer misclassifies a zero-day anomaly as a known category:

$$\mathcal{L}_{\text{zero-day}}^{\text{adjusted}} = \mathcal{L}_{\text{zero-day}} + \lambda_{\text{zero-day}} \cdot \sum_{j=1}^{m} P_{\text{main}}(C_j|\mathbf{x}) \cdot \left(1 - P_{\text{main}}(C_{\text{zero-day}}|\mathbf{x})\right) \tag{44}$$

To further refine the loss function, we introduce higher-order regularization terms. These terms control overfitting by imposing penalties on high-order activations or weights, ensuring that the model does not over-rely on certain features. The regularization terms are:

$$\mathcal{L}_{\text{reg}} = \lambda_{\text{weight}} \cdot \sum_{i=1}^{N} \|W_i\|^2 + \lambda_{\text{activation}} \cdot \sum_{i=1}^{N} \|A_i\|^2 \tag{45}$$

Where W_i and A_i represent the weight and activation vectors, respectively. These terms are then combined with the original loss function:

$$\mathcal{L}_{\text{final}} = \mathcal{L} + \mathcal{L}_{\text{reg}} \tag{46}$$

Thus, the final complete loss function is the weighted sum of all components, which can be expressed as:

$$\mathcal{L}_{\text{final}} = \alpha \cdot \mathcal{L}_{\text{main}}^{\text{adjusted}} + \beta \cdot \mathcal{L}_{\text{sub}}^{\text{regularized}} + \gamma \cdot \mathcal{L}_{\text{zero-day}}^{\text{adjusted}} + \lambda_{\text{weight}} \cdot \sum_{i=1}^{N} \|W_i\|^2 + \lambda_{\text{activation}} \cdot \sum_{i=1}^{N} \|A_i\|^2 \tag{47}$$

This loss function now incorporates the hierarchical nature of the classification task, the dependency between the layers, the conditional terms, regularization, and the zero-day detection component, which all together create a highly complex and interdependent model.

3.6 Concluding Remarks

This section primarily contributes to enhancing smart home intrusion detection by proposing a novel multi-tier detection system. The system leverages both internal IoT device-to-device communications and external network traffic to provide accurate and comprehensive detection and identification of malicious activities. By considering the unique characteristics of smart home IoT networks, this model addresses the shortcomings of traditional intrusion detection systems and represents a significant advancement in IoT security methodologies.

4 Creating the Training and Testing Dataset

This section outlines the comprehensive process involved in creating the new dataset. It begins with evaluating existing datasets from various perspectives, establishing a scoring system, and identifying their current limitations. Following this assessment, we describe the sequential steps to develop the new dataset from the ground up, focusing on addressing the identified shortcomings. Finally, the new dataset is compared against previously available datasets to highlight improvements and advancements.

4.1 Available Datasets

This subsection examines key datasets pertinent to the Smart Home and Smart Office IoT domains. Each dataset in this review comprises at least five real IoT devices to ensure comprehensive field coverage. Summaries of these datasets are provided in Table 2, highlighting their core attributes and contributions. Subsequently, we outline a set of evaluation criteria, assigning importance factors to each, to facilitate a detailed comparative analysis (score column in Table 2). This analysis reveals existing gaps in the datasets, thereby emphasizing the urgent need for a more encompassing dataset to address these shortcomings effectively.

1. ***IoT SENTINEL [76]***:

   IoT SENTINEL is a comprehensive collection of network traffic captures from various IoT devices. It includes fingerprints of 31 smart home IoT devices representing 27 different types. Each device's setup was repeated at least 20 times, generating multiple captures to ensure robustness in the dataset. The traffic captures focus on the initial setup phase of the devices. The features extracted from the traffic include protocol types, packet sizes, IP addresses, and ports used. The dataset is publicly available for research purposes and can be accessed through Aalto University's research portal [76, 77].

2. ***N-BaIoT [78]***:

   The N-BaIoT dataset is a comprehensive collection designed to facilitate studying network-based anomaly detection in IoT environments, particularly focusing on detecting botnet attacks. The dataset was created by infecting nine different commercial IoT devices with two well-known IoT-based botnets, Mirai [79] and BASHLITE [80], and collecting the resulting network traffic. The devices used include security cameras, baby monitors, and smart lights, reflecting a realistic range of IoT devices commonly found in domestic and enterprise environments. The dataset comprises raw network traffic captures (in pcap format), including benign traffic and traffic generated during various stages of botnet attacks, such as scanning, flooding, and command execution [78].

3. ***UNSW benign (IEEE TMC 2018) [81]***:

   This dataset was compiled over six months from a testbed containing 28 different IoT devices, which included cameras, lights, plugs, motion sensors, appliances, and health monitors. The data capture covered various network traffic characteristics, including autonomous device communications and user-interaction-driven traffic. Key attributes extracted from the traffic data include flow volume, flow duration, average flow rate, and device sleep time, as well as more nuanced signaling patterns such as DNS queries, NTP requests, and cipher suite exchanges [81].

4. ***UNSW attack (ACM SOSR 2019) [82]***:

   This dataset was created within a controlled laboratory environment using a diverse set of consumer IoT devices, including smart plugs, cameras, motion sensors, and various smart home devices. The primary objective was to capture benign traffic, various volumetric attack scenarios, and various reflective attacks. The dataset includes detailed annotations of the attack scenarios, specifying the type of attack, the duration, the attack rate, and the targeted IoT devices. This structured labeling is crucial for developing and benchmarking machine learning models to detect and mitigate IoT-specific volumetric attacks. The authors also provide metadata that helps understand the network behavior under normal and attack conditions, including features like packet counts, byte counts, and flow statistics [82].

5. ***UNSW MUD (IEEE TDSC 2020) [83]***:

   This dataset was created to facilitate the verification and monitoring of IoT devices' network behavior using Manufacturer Usage Description (MUD) profiles. This dataset comprises network traffic data from 28 consumer IoT devices collected over six months. The devices include various common smart home devices such as cameras, sensors, and smart plugs. The traffic captures benign and potentially malicious interactions, providing a robust basis for developing and testing MUD profiles.

   The dataset was generated using a tool called MUDgee, which automatically creates MUD profiles from the traffic traces of IoT devices. The tool captures all possible benign states by combining autonomous and interactive approaches. For autonomous data capture, user interactions with the IoT devices were recorded using a touch replay tool on a tablet. Additionally, direct user interactions were captured in a controlled lab environment [83].

6. ***IoT Information Exposure [84]***:

   This dataset is a comprehensive and detailed collection designed to investigate the extent and nature of information exposure from consumer IoT devices. It includes network traffic data from 81 IoT devices tested in controlled laboratory environments in the United States and the United Kingdom. The dataset encompasses data from 34,586 experiments, both automated and manual, to capture a broad spectrum of interactions and behaviors from various IoT devices such as smart cameras, speakers, TVs, and other smart home appliances.

   The data collection methodology involved setting up these devices in environments that simulate typical home settings and capturing all network traffic using tcpdump. The experiments were designed to cover a range of scenarios including power-on, idle, local and remote interactions, and specific user-triggered actions. This comprehensive approach ensures that the dataset captures not only the typical operational data but any potential unexpected or privacy-compromising behaviors exhibited by the devices.

   The dataset focuses on multiple dimensions of information exposure: the destinations of the Internet traffic generated by the devices, the encryption status of the communications, the types of data being transmitted, and any inferred device activities from the traffic patterns [84].

7. ***Bot-IoT [85]***:

   The Bot-IoT dataset is designed to support developing and evaluating machine-learning models for detecting botnet attacks. The comprehensive dataset contains approximately 73 million instances of network traffic data. The dataset comprises multiple subsets, each tailored for different stages of data analysis and machine

learning model training. The primary subsets include the Raw Set, the Full Set, the 5% Subset, and the 10-Best Subset.

The Raw Set comprises around 70 GB of packet capture (PCAP) files containing unprocessed network data. The Full Set is provided in CSV format and generated using the Argus network security tool. It contains about 73 million instances, each representing a network session. This set includes 26 independent features that capture various aspects of the network traffic. The 5% Subset is a more manageable version of the Full Set, containing roughly 3.6 million instances. This subset includes 43 independent features, combining the Argus network flow features and additional calculated features developed by the dataset authors. The 10-Best Subset is derived from the 5% Subset and contains the same number of instances but focuses on the 10 most relevant independent features. These features were selected based on their correlation coefficient and joint entropy, making this subset ideal for initial model training and feature selection studies [85].

8. ***MedBIoT [86]***:

   The MedBIoT dataset facilitates the detection and analysis of IoT botnet attacks within a medium-sized network environment. This dataset captures normal and malicious traffic from an IoT network comprising 83 devices. These devices include a mix of real and emulated devices, such as smart locks, fans, light bulbs, and switches, reflecting a realistic and diverse IoT ecosystem.

   The researchers deployed three botnet malware families to generate the dataset: Mirai, BashLite, and Torii. The network was set up in a controlled environment, ensuring a comprehensive capture of network traffic during various stages of botnet activity, including infection, propagation, and command and control (C&C) communication. The dataset includes approximately 17.8 million packets, with around 30% of the traffic labeled malicious. A significant advantage of the MedBIoT dataset is its inclusion of the Torii botnet, which had not been extensively studied or included in other publicly available datasets before [86].

9. ***UNSW IPFIX [87]***:

   This dataset is a collection of IP Flow Information Export (IPFIX) records aimed at inferring the types of IoT devices connected in residential ISP networks. Over three months, this dataset was collected from a testbed comprising 26 different IoT devices, including smart cameras, speakers, light bulbs, and sensors. The data collection process involved capturing network traffic from these devices using IPFIX, a standard flow-level telemetry protocol, to monitor and analyze their network behavior without requiring modifications to the home network setups.

   The dataset includes approximately nine million IPFIX records, capturing flow-level statistics and features. These records contain 28 flow features, including packet counts, flow duration, interarrival times, and byte counts. The data was collected in a controlled environment, simulating typical residential network conditions to ensure realistic and comprehensive coverage of IoT network traffic [87].

10. ***CIC IoT Enriched Dataset [88]***:

    This dataset addresses the limitations of two existing datasets: BoT-IoT and TON-IoT. It offers a more detailed and comprehensive resource for developing machine-learning models to detect and classify IoT-based attacks. It is constructed by merging several IoT datasets, each covering specific aspects of IoT network traffic and attack scenarios. This integration process, known as vertical enrichment, results in a unified dataset that captures various attack types, device behaviors, and network conditions.

A key innovation of the CIC Enriched IoT Dataset is the introduction of novel feature sets that enhance its analytical richness. These features are categorized into three main groups: connectivity, dynamic, and layered. Connectivity features focus on relationships between network packets and flows, capturing patterns that may indicate abnormal or malicious behavior. Dynamic features capture temporal aspects of traffic, such as changes in traffic patterns over time and variations in traffic volume, which are crucial for identifying time-based attacks. Layered features extract protocol-specific metrics from different network stack layers, providing insights into how attacks manifest across the communication layers. These enriched features enable more accurate and robust detection and classification of various attack types, including denial-of-service (DoS), man-in-the-middle (MitM), and port scanning.

11. ***TON_IoT* [89]**:

    This dataset is designed to support developing and evaluating intrusion detection systems specifically tailored to IoT and industrial IoT environments. This dataset was collected from a medium-scale network testbed at the Cyber Range and IoT Labs at UNSW Canberra, capturing data from multiple layers, including Edge, Fog, and Cloud, to simulate realistic IoT/IIoT network configurations.

    The dataset includes telemetry data from various IoT/IIoT services, operating system logs, and network traffic. The telemetry data encompasses a wide range of IoT devices, such as smart fridges, GPS trackers, motion lights, garage doors, Modbus, thermostats, and weather sensors, with seven distinct IoT/IoT sensors. The collected data reflects both normal operations and multiple types of cyber-attacks [89].

12. ***IoTLS [90]***:

    This dataset focuses on Transport Layer Security (TLS) usage in consumer IoT devices, aiming to understand how effectively these devices establish secure connections, validate certificates, and adapt to evolving security protocols over time. The dataset includes over two years of TLS network traffic data from 40 different IoT devices, covering categories such as cameras, smart hubs, home automation, TVs, audio devices, and appliances. The collected data spans approximately 17 million TLS connections, obtained through passive and active experiments.

    The passive experiments involved recording network traffic generated by the devices over a long period. This passive data collection allows researchers to observe the evolution of TLS usage and configurations in these devices without direct intervention. In contrast, the active experiments involved using tools like mitmproxy to intercept and analyze TLS connections actively, testing for vulnerabilities such as insecure protocol versions, ciphersuites, and improper certificate validation.

    The IoTLS dataset provides detailed TLS handshake data, including ClientHello and ServerHello messages, which are crucial for understanding the supported TLS versions and ciphersuites. Additionally, the dataset includes information on the presence and validation of root certificates, revealing that many devices still trust deprecated and potentially compromised certificates, posing significant security risks [90].

13. ***MQTTSet [91]***:

    This dataset is designed for the analysis and detection of cyber-attacks targeting the MQTT protocol in IoT environments. It was created using IoT-Flock, a network traffic generator tool capable of emulating

IoT devices and networks based on MQTT and CoAP protocols. The dataset includes legitimate and malicious traffic, providing a comprehensive foundation for training and validating machine learning models for anomaly detection in MQTT-based IoT networks.

The dataset consists of network traffic data from a simulated smart home environment with various IoT sensors, including temperature, light intensity, humidity, motion sensors, CO-gas detectors, smoke detectors, fan speed controllers, and door locks. It includes raw traffic data in PCAP format, captured over a one-week period, and represents approximately 11.9 million network packets.

The authors included several types of cyber-attacks, including DoS attacks, MQTT publish floods, SlowITe, malformed data attacks, and brute force authentication attacks. Key features extracted from the MQTTset dataset include various MQTT protocol-specific fields, such as connect flags, message identifiers, and quality of service (QoS) levels, as well as general network features like TCP flags, packet lengths, and inter-arrival times [91].

14. ***koleun et al (at TMA 2021 data) [92]***:

    This dataset, developed for IoT device identification, encompasses network traffic data collected over 27 weeks from a testbed comprising 41 IoT devices. These devices include a variety of smart home technologies such as cameras, smart TVs, smart speakers, hubs, and various home automation appliances like thermostats and smart plugs.

    The dataset captures detailed packet-level traffic data, stored in pcap format, with traffic filtered by the MAC addresses of each device. The data was processed to extract various features at different granularities, including one-hour windows, one-second windows, and individual TCP/UDP flows. Key features extracted include flow volume, flow duration, sleep time, DNS and NTP intervals, packet sizes, and inter-packet intervals [92].

15. ***27-Botnet [93]***:

    This dataset encompasses network communication logs from 260 botnet binaries spanning 27 distinct botnet families. These logs were from the VirusShare and VirusTotal repositories. To augment realism, benign network logs were collected over 18 days from various IoT devices, including motion sensors, smart cameras, plugs, bulbs, speakers, media streaming devices, and more. The dataset includes both training and test sets, with the training set containing logs from 25 botnet families and 8 IoT devices. In contrast, the test set includes additional zero-day scenarios featuring Mirai and Satori botnets, which were not part of the training data [93].

16. ***BLEBeacon [94]***:

    The BLEBeacon dataset is a collection of Bluetooth Low Energy (BLE) advertisement packet traces generated from BLE beacons carried by participants during their daily routines inside a university building. The data collection spanned one month and involved 46 participants, each carrying a BLE beacon. The experiment was conducted in a multi-floor facility, with 32 Raspberry Pi 3 devices deployed to continuously gather BLE advertisement packets and store them in a cloud-based environment.

    The BLEBeacon dataset includes two primary files: the RSSI Report file and the Check-In/Check-Out Report file. The RSSI Report file contains entries that include a unique identifier for each packet, the beacon ID, the Received Signal Strength Indicator (RSSI) in dB, a timestamp, and the ID of the Raspberry Pi that received

the packet. The Check-In/Check-Out Report file includes similar entries but also records the time a user enters and exits the vicinity of an RPi, marked by the first and last advertisement packets received [94].

17. ***cmu/zigbee-smarthome [95]***:

    This dataset is designed to analyze the security of Zigbee-enabled smart homes. The dataset includes Zigbee network traffic captured from various commercial Zigbee devices in a controlled testbed environment. It was created to investigate the potential security vulnerabilities in Zigbee networks, specifically focusing on passive inspection, jamming, and spoofing attacks.

    The dataset was generated using various commercial Zigbee devices, including smart bulbs, motion sensors, outlets, and door locks. These devices were configured to operate in a Zigbee network managed by a SmartThings hub. The network traffic was captured using software-defined radios (SDRs) and IEEE 802.15.4 USB adapters, ensuring a comprehensive collection of Zigbee packets. The captured packets were stored in PCAP format and included various Zigbee communication frames such as MAC beacons, MAC commands, NWK commands, APS commands, and ZDP commands [95].

18. ***GHOST-IoT [96]***:

    The GHOST-IoT dataset collects network traffic data captured from a real-world smart-home environment. The dataset includes traffic from multiple IoT devices interacting within a typical smart-home setup over ten days. The devices involved include motion sensors, door sensors, emergency buttons, weight scales, and blood pressure meters, utilizing protocols such as ZigBee, Bluetooth, WiFi, and RF869. During the initial nine days, the dataset reflects normal device usage. On the final day, seven distinct scenarios were introduced to simulate abnormal behaviors, such as device malfunctions and potential cyber-attacks, to enrich the dataset with anomalous traffic patterns. The captured data is processed through the Network and Data Flow Analysis (NDFA) component [96].

19. ***MQTT-IoT-IDS2020 [97]***:

    This dataset is designed to address the need for robust and comprehensive data for developing and evaluating intrusion detection systems, specifically for the MQTT protocol in IoT environments. It includes a mix of legitimate and malicious traffic collected from an emulated IoT network setup. The emulation involves a variety of IoT sensors, such as temperature, humidity, motion sensors, and actuators like door locks and fans, interacting with an MQTT broker.

    The dataset includes raw packet captures (PCAP) and processed CSV files. The PCAP files contain approximately 11.9 million network packets captured over one week. The processed dataset includes extracted features from the MQTT protocol, such as connection flags, message identifiers, quality of service (QoS) levels, and specific MQTT command types. MQTT-IoT-IDS2020 also incorporates various cyber-attack scenarios to simulate real-world threats. These scenarios include DoS attacks, MQTT publish floods, SlowITe, malformed data attacks, and brute force authentication attacks [97].

20. ***IoT-CIDDS [98]***:

    This dataset is developed as part of the IoT-Sentry project to address the need for realistic, cross-layer attack data in standardized IoT networks. The dataset includes traffic data generated from simulations of various

IoT attacks using the Cooja IoT simulator. It captures benign and malicious traffic across multiple layers of the IoT communication stack, specifically focusing on RPL, UDP, and ICMPv6 protocols.

The dataset encompasses five types of attacks: UDP Flooding, Blackhole, Selective Forwarding, Hello Flooding, and ICMPv6 Ping of Death. Each attack scenario was simulated in a network environment comprising 100 nodes. The dataset includes raw packet captures (PCAP files) and processed CSV files, which contain features extracted from the network traffic. These features include packet counts, transmission and reception rates, packet lengths, and various control message statistics [98].

21. ***DoS/DDoS-MQTT-IoT [99]***:

    This dataset is designed to evaluate intrusion detection systems in IoT networks using the MQTT protocol. It addresses the scarcity of real-world test data in MQTT-based IoT environments, particularly focusing on DoS and DDoS attack scenarios. The dataset is generated using a physical IoT testbed, which encompasses legitimate and malicious traffic. The physical testbed includes various sensors, such as temperature, humidity, and CO sensors, connected to Raspberry Pi devices configured as publishers. The dataset captures MQTT traffic across multiple attack scenarios, including CONNECT flooding, delayed CONNECT flooding, invalid subscription flooding, and TCP SYN flooding attacks. The dataset is presented in raw PCAP and processed CSV files, facilitating its use in various analysis and machine learning applications [99].

22. ***X-IIoTID [100]***:

    This dataset is designed specifically for Industrial Internet of Things environments. It is connectivity-agnostic and device-agnostic, addressing the challenges associated with the heterogeneity and interoperability of IIoT systems. It captures various attack behaviors and system activities across multiple connectivity protocols and IIoT devices, ensuring a realistic and thorough representation of IIoT network traffic and intrusions.

    The X-IIoTID dataset includes a diverse range of IIoT devices, such as programmable logic controllers (PLCs), sensors, actuators, edge gateways, and cloud services, configured within a testbed called the Brown-IIoTbed. This testbed simulates a real-world IIoT environment and covers three tiers of IIoT systems: edge, platform, and enterprise. The dataset captures data from multiple sources, including network traffic, host resources, logs, and security alerts.

    The attack traffic includes a variety of sophisticated and multi-stage attacks, such as reconnaissance, exploitation, lateral movement, command and control (C&C), exfiltration, and tampering. Key features extracted from the dataset include network flow characteristics, resource utilization metrics, and specific attack signatures [100].

23. ***Edge-IIoTset [101]***:

    This dataset is designed to support developing and evaluating intrusion detection systems in IoT and IIoT environments. It was created using a seven-layer testbed that includes various devices, sensors, protocols, and cloud/edge configurations. The testbed simulates a real-world IoT/IIoT environment. It encompasses several layers: Cloud Computing, Network Functions Virtualization (NFV), Blockchain Network, Fog Computing, Software-Defined Networking (SDN), Edge Computing, and IoT/IIoT Perception layers.

    The IoT data in the Edge-IIoTset dataset is generated from more than ten types of sensors, including temperature and humidity sensors, ultrasonic sensors, water level detection sensors, pH sensor meters, soil moisture

sensors, heart rate sensors, and flame sensors. The dataset includes fourteen types of attacks related to IoT and IIoT connectivity protocols, categorized into five main threat groups: DoS/DDoS attacks, information gathering attacks, man-in-the-middle (MitM) attacks, injection attacks, and malware attacks.

Key features of the dataset are extracted from various sources, including alerts, system resources, logs, and network traffic. The dataset includes 61 highly correlated features selected from an initial set of 1176 features. These features are extracted using network protocol analyzers such as Zeek and TShark and encompass attributes from protocols like IP, ARP, ICMP, HTTP, TCP, UDP, DNS, MQTT, and Modbus/TCP [101].

24. ***Kitsune [102]***:

    This dataset is designed to evaluate the performance of Kitsune, an online network intrusion detection system (NIDS) that utilizes an ensemble of autoencoders. Kitsune is tailored to detect anomalies in IoT networks by efficiently processing network traffic in real-time. The dataset captures network traffic from various scenarios involving normal and malicious activities.

    The dataset is generated from a testbed with multiple IoT devices, such as IP cameras, smart speakers, and various sensors, connected in a network that mirrors real-world conditions. The captured traffic encompasses several attacks, including OS scanning, fuzzing, man-in-the-middle attacks, ARP spoofing, active wiretaps, SSDP flooding, SYN DoS, SSL renegotiation attacks, and botnet malware like Mirai. The dataset includes the raw packet capture (PCAP) files and extracted features in CSV format. The features include network statistics like packet sizes, inter-arrival times, and protocol-specific details, which are critical for anomaly detection [102].

25. ***Plegma [103]***:

    This dataset is a collection of high-frequency electricity consumption measurements collected from 13 households in Greece over one year. The dataset includes aggregate household electricity usage and detailed appliance-level consumption recorded at 10-second intervals. The dataset also incorporates environmental data, including temperature and humidity, along with detailed metadata about the households, such as building characteristics, demographic information, and appliance usage practices. The dataset includes 218 million readings from 88 installed meters and sensors. It is structured only in CSV format. The data collection setup involves smart meters and sensors communicating with IoT gateways via the Z-Wave protocol, ensuring reliable data transmission and storage [103].

26. ***IoT-SH [7]***:

    This dataset is designed for the development and evaluation of supervised intrusion detection systems tailored for smart home IoT environments. This dataset comprises normal and malicious network traffic generated from various IoT devices. The normal traffic includes regular device communication patterns. In contrast, malicious traffic encompasses various attack types, such as Denial of Service, data exfiltration, and man-in-the-middle attacks. The dataset's creation involved a comprehensive setup of a smart home environment where devices like smart plugs, lights, cameras, and other IoT appliances were deployed. The dataset includes features extracted from the raw network traffic, such as packet size, inter-arrival times, and protocol-specific details [7].

27. ***CICIoT2022 [104]***:

This dataset is designed to support the development of advanced IoT identification, profiling, and intrusion detection systems. Developed by the Canadian Institute for Cybersecurity, this dataset captures detailed network traffic from 60 IoT devices, including WiFi, ZigBee, and Z-Wave devices, under various operational states such as powered on, idle, and active, as well as during interactions. The dataset provides a holistic view of device behavior by including normal and malicious traffic generated from flood denial-of-service (DoS) attacks and RTSP brute-force attacks.

Data collection was conducted in a lab environment and involved multiple stages. Devices were individually powered on to capture their initial communication patterns, left in an idle state over thirty-eight-hour periods to observe any unexpected communications, and actively interacted with through companion apps, voice assistants, and physical interactions to record typical usage traffic. Smart home scenarios were also simulated to capture inter-device interactions during common activities such as coming home, leaving home, and detecting intrusions. Specific attacks were performed to generate malicious traffic, providing valuable data for developing intrusion detection systems [104].

28. ***CICIoT2023 [105]***:

    This dataset is created to benchmark large-scale attacks in IoT environments. Developed by the Canadian Institute for Cybersecurity, it involves an extensive IoT topology comprising 105 devices, including smart home devices, sensors, cameras, and microcontrollers, configured to perform and capture various attacks. The dataset includes 33 distinct attacks categorized into seven classes: Distributed Denial of Service (DDoS), Denial of Service (DoS), reconnaissance (Recon), web-based attacks, brute force, spoofing, and Mirai botnet attacks. The captured traffic is stored in PCAP format and subsequently processed into CSV files, allowing for detailed feature extraction and analysis. Key features include packet sizes, inter-arrival times, and various protocol-specific details [105].

29. ***Zmad [26]***:

    The ZMAD dataset is introduced as part of a study on lightweight anomaly-based intrusion detection systems. It is specifically designed to use the Z-Wave protocol in smart home automation. The dataset includes traffic data collected from a testbed of 17 top-rated real-world Z-Wave smart home devices, including controllers, door locks, motion sensors, LED lights, plugs/outlets, and window contact sensors. This dataset was gathered to address the security vulnerabilities inherent in Z-Wave devices, both legacy and those using the newer S2 security framework.

    The dataset contains a mix of normal and abnormal traffic, capturing routine device operations and simulated attack scenarios. Normal traffic was generated by regularly interacting with the devices using a smartphone app (e.g., turning lights on/off, checking device status) and recording the resulting network packets over ten days. In total, 158,110 normal packets were collected. For the abnormal traffic, 5,171 packets were generated using known vulnerabilities and fuzzing techniques, which include a variety of attacks such as invalid route parameters, route table manipulation, remote code execution, and denial-of-service (DoS) attacks [26].

30. ***IoTID20 [106]***:

    This dataset is designed for detecting anomalous activities in IoT networks. It was created using a testbed that simulates a smart home environment, incorporating various IoT devices and interconnecting structures.

The primary devices used in the testbed are the SKT NGU smart home device and the EZVIZ Wi-Fi camera, which were selected to represent typical IoT devices in a smart home setup. These devices were connected to a smart home Wi-Fi router and other devices, such as laptops, tablets, and smartphones, which acted as attacking devices.

The normal traffic was generated through regular interactions with the IoT devices, capturing typical communication patterns. The anomalous traffic encompasses a variety of attack types, including DoS, Mirai, Man-in-the-Middle (MITM), and scanning attacks. These attacks were executed to simulate real-world cyber threats. The dataset comprises 83 network features and three label features (binary, category, and sub-category). Key features of the IoTID20 dataset include various network statistics such as packet sizes, inter-arrival times, and specific protocol details, which are crucial for identifying patterns and anomalies in network traffic [106].

4.1.1 Evaluation Criteria

This subsection presents the evaluation criteria, briefly explains each one and its importance, and describes how each criterion is calculated when evaluating a dataset. For each dataset, the evaluation metric is calculated using the following formula:

$$evaluation\_metric = \sum_{i=1}^{n} W_i \cdot E_i \tag{48}$$

where n is the number of evaluation criteria, W_i is the weight corresponding to each evaluation criterion, and E_i is the value of the related evaluation criterion, ranging from 0 to 10.

1. ***Release Date***: Being up-to-date is crucial as it ensures the dataset reflects the latest technological advancements and threat scenarios. Datasets from recent years likely use more modern devices and technologies in their testbeds, address issues from previous datasets, and cover newer threat scenarios. Moreover, newer datasets are more likely to include recent security patches and updates, making them more relevant for current research. We consider datasets from the last 10 years. The most recent datasets (2024) receive a score of 10, with one point deducted for each year prior. For example, a dataset from 2020 would score 6.

2. ***Number of Features***: Covering a wide range of perspectives is important because datasets with more features provide a more comprehensive analysis of raw data. This allows researchers from various domains to work on the data. For instance, some researchers might focus on time behavior, while others might analyze size behavior. Additionally, a dataset with diverse features can enable multi-dimensional analysis and cross-validation of results, enhancing the robustness of research findings. In our comparison, the dataset with the most features serves as the baseline. For example, if the highest number of features is 100, that dataset scores 10, while a dataset with 60 features scores 6.

3. ***Realistic Traffic***: Similarity to real-world scenarios is vital because realistic traffic is more applicable to real-world cases and industry use. Real traffic encompasses different types of traffic, unforeseen challenges, noise, and other factors that simulated traffic cannot replicate. Realistic datasets provide a more accurate representation of network behavior under various conditions, making them invaluable for developing and testing security measures. In our comparison, purely realistic traffic scores 10, simulated traffic scores 0, and

Table 2: IoT datasets comparison. (DP: Device Profiling)

Name	*Date*	*Main Proto.*	*Feat.*	*Real. Traf.*	*IoT Traf.*	*IP Traf.*	*Cls.*	*Dev.*	*Dur.*	*Meta Data*	*IP Att.*	*IoT Att.*	*Benign Records*	*Malicious Records*	*Ref.*	*Sc.*
IoT SENTINEL	2017	IP	23	✓	✗	✓	DP	31	—	✓	0	0	192,871	0	[76]	4
N-BaIoT	2018	IP	23-115	✓	✗	✓	11	9	—	✗	10	0	502,595	6,506,674	[78]	12
UNSW benign	2016	IP	9	✓	✗	✓	DP	28	20 d	✓	0	0	21,061,054	0	[81]	3
UNSW attack	2019	IP	42	✓	✗	✓	10	10	44 d	✓	9	0	570,506	1,970	[82]	13
UNSW MUD	2020	MUD	✗	✓	✗	✓	DP	28	6 mo	✗	0	0	✗	✗	[83]	8
Ren et al	2019	IP	33	✓	✗	✓	0	81	4.3 d	✓	0	0	450,177	0	[84]	4
Bot-IoT	2019	IP,MQTT	32-10	✗	✗	✓	7	5	—	✓	6	0	9,543	73,360,900	[85]	8
MedBIoT	2020	IP	100	Mix	✗	✓	4	7 - 89	1 w	✓	3	0	12,540,478	5,305,089	[86]	7
UNSW IPFIX	2020	IPFIX	28	✓	✗	✓	DP	26	3 mo	✗	0	0	12,853,547	0	[87]	5
TON_IoT	2021	IP	44	✗	✗	✓	9	10	—	✓	8	0	245,000	124,619	[89]	9
IoTLS	2020	IP, TLS	✗	✓	✗	✓	—	40	∼2 y	✓	9	0	✗	✗	[90]	16
MQTTSet	2020	IP,MQTT	33	✗	✗	✓	6	10 V	1 w	✓	5	5-MQTT	115,824	115,822	[91]	12
koleun	2021	IP	37	✓	✗	✓	DP	41	27 w	✓	0	0	136,180,819	0	[92]	9
27-Botnet	2021	IP	115	✓	✗	✓	28	10	18 d	✗	27	0	—	—	[93]	30
BLEBeacon	2018	BLE	6	✓	✓	✗	DP	46	33 d	✗	0	0	10,264,367 P	0	[94]	3
cmu/zigbee-smarthome	2020	Zigbee	✗	✓	✓	✗	DP	10	35 h	✓	0	0	✗	✗	[95]	3
GHOST-IoT	2019	IP, PPP, BT, RF Zigbee,	IP:34 IoT:60	✓	✓	✗	DP	14	10 d	✗	0	0	3,811,419 P, 546,954 F	0	[96]	2
MQTT-IoT-IDS2020	2020	MQTT	uni-F:19 bi-F:32 Pckt:31	✗	✓	✓	5	14	—	✓	4	1-MQTT	uF: 376,186 bi-F:188,378 P:2,317,158	uF: 119,339 bi-F:71,001 P:29,805,145	[97]	9
IoT-CIDDS	2021	IP,RPL, 6LoW	21	✗	✗	✓	6	100	1 d	✗	5	0	16,118	79,181	[98]	7
DoS/DDoS-MQTT-IoT	2023	MQTT	30	✓	✗	✓	11	88	—		10	10	∼66,800,000	∼92,100,000	[99]	23
X-IIoTID	2021	Modbus, CoAP, MQTT	67	✓	✗	✓	19	20	114 d	✗	18	0	421,417	399,417	[100]	22
Edge-IIoTset	2022	Modbus, MQTT	61	✓	✗	✓	15	20	51 d	✓	14	0	11,223,940 P	9,728,708 P	[101]	17
Kitsune	2018	IP	115	✗	✗	✓	10	13	—	✓	9	0	700,000	26,472,754	[102]	11
Plegma	2024	Z-wave, MQTT	0	✓	✗	✗	DP	88 (13 Ho)	1 y	✗	0	0	30,712,034	0	[103]	26
IoT-SH	2019	IP, WiFi	10	✗	✗	✓	13	8	5 w	✗	12	1	170,785	50	[7]	15
CICIoT2022	2022	IP, ZgB, Z-wave	48	✓	✓	✓	DP	48	545 h	✓	0	0	466,083	0	[104]	5
CICIoT2023	2023	IP	46	✓	✗	✓	34	105	28 h	✓	33	0	1,098,195	45,588,384	[105]	36
Zmad	2023	Z-wave	30	✓	✓	✗	2	17	10 d	✗	5	5	158,110	5,171	[26]	12
IoTID20	2020	IP	83	✓	✗	✓	9	5	—	✗	8	0	40,073	585,710	[106]	11
BCCC-IoT-Zwave-2024	2024	IP, Z-wave, MQTT, WBSoc, HTTP/S	350 IP 340 IoT	✓	✓	✓	100	110	143 d	✓	66	34	∼1.5 million	∼600 million	—	115

mixed traffic scores 5. If a dataset contains simulated traffic or devices, the scores for "Number of Devices", "Capturing Duration", and "Metadata Availability" features are halved.

4. ***IoT-Specific Traffic***: Detailed IoT analysis is essential as it allows for a more accurate assessment of the entire network and individual devices. IoT-specific traffic includes real packets transmitted to and from each device, such as Z-Wave or ZigBee packets. Most previous works focus on IP packets, which the Smart Home hub typically sends. However, not all device behaviors and communications are visible on the network. Capturing IoT-specific traffic is challenging due to different frequency ranges (e.g., Z-Wave, Bluetooth) and requires specialized devices, tools, and techniques. This traffic provides insights into device-specific behaviors, crucial for understanding and mitigating IoT-specific threats. In our comparison, datasets with IoT-specific traffic score 10; otherwise, they score 0.

5. ***IP-based traffic***: IP-based traffic is significant because it encompasses the common communication protocols used in IoT networks, enabling easier integration and analysis. Datasets with IP-based traffic provide a broad perspective on network behavior and are often more accessible for analysis. This type of traffic allows for the application of a wide range of network analysis tools and techniques, facilitating comprehensive security assessments and performance evaluations. Datasets with IP-based traffic score 10; otherwise, they score 0.

6. ***Number of labels***: The number of labels is important as it indicates the variety of classifications available within the dataset. More labels suggest a richer dataset with more detailed information, facilitating diverse research applications. A well-labeled dataset allows for developing more sophisticated machine-learning models and improves the accuracy of anomaly detection and classification tasks. In our comparison, the dataset with the most labels serves as the baseline. For example, if the highest number of labels is 18, that dataset scores 10, while a dataset with 9 features scores 5.

7. ***Number of real devices***: The number of real devices used in the dataset reflects the dataset's authenticity and applicability. Datasets with more real devices provide more realistic scenarios and diverse data, which is crucial for testing and validating IoT solutions in practical environments. Including various devices ensures that the dataset covers potential vulnerabilities and usage patterns. In our comparison, the dataset with the highest number of devices serves as the baseline. Furthermore, if the dataset includes simulated traffic, the score for this criterion will be halved.

8. ***Capturing Duration***: The capturing duration is crucial because longer durations provide a more comprehensive view of the network's behavior over time. Extended capturing periods help in understanding long-term trends, seasonal variations, and rare events that may not be visible in shorter datasets. This is particularly important for identifying persistent threats and assessing the stability and reliability of IoT systems. Datasets with longer capturing durations are more valuable and score higher. Our comparison uses the dataset with the longest duration as the baseline. Moreover, the score for this criterion will be diminished by 50% if the dataset contains simulated traffic.

9. ***Metadata Availability***: The availability of metadata and raw data is essential for in-depth analysis. Datasets that provide extensive metadata and raw data allow researchers to perform more detailed and varied analyses. Metadata can include information about the network environment, device configurations, and contextual details critical for replicating experiments and validating results. This transparency enhances the credibility

and utility of the dataset. Datasets with metadata availability score 10, otherwise 0. Additionally, if the dataset comprises simulated traffic, the score for this criterion will be reduced by half.

10. ***Number of Attacks/Malwares***: The number of attacks or malware instances in the dataset is important as it indicates the dataset's relevance for security research. Datasets with more attacks or malware instances provide richer data for developing and testing security solutions. A diverse range of attacks helps evaluate the robustness of defense mechanisms under different threat scenarios and improves the generalizability of research findings. Datasets with a higher number of attacks or malware instances score higher. In our comparison, the dataset with the most malicious labels is used as the baseline.

11. ***Number of IoT-Specific Attacks/Malwares***: The number of IoT-specific attacks or malware is particularly critical as these are harder to execute, have a higher impact on the network, and are more difficult to detect and mitigate. IoT-specific attacks target the unique vulnerabilities of IoT devices and protocols, making their detection and prevention more challenging. Datasets with more IoT-specific attacks provide valuable insights into these complex threat vectors and are essential for advancing IoT security research. Our comparison uses the dataset with the highest number of IoT-specific malicious labels as the baseline.

    In this analysis, the importance of IoT-specific attacks is twice that of non-IoT-specific attacks, reflecting their inherent complexity and greater potential impact on network security. For example, if two datasets contain four attacks—one consisting solely of IP-based attacks and the other comprising four IoT-specific attacks, the dataset with the IoT-specific attacks will receive a total score of eight. This score combines the four points allocated for the previous category, the total number of attacks, and an additional four points for the IoT-specific attacks category. In contrast, the dataset containing only IP-based attacks will receive a score of four (just for the previous category). This framework highlights the critical nature of IoT attacks in our comparative analysis, recognizing that they are significantly harder to execute, more challenging to detect and pose a greater threat to network integrity.

12. ***Data Distribution***: Balanced data distribution is important for ensuring the dataset is representative and unbiased. A well-balanced dataset ensures that all classes or categories are adequately represented, reducing the risk of biased analysis and improving the reliability of research outcomes. Balanced datasets are crucial for training machine learning models that generalize well to real-world data and for conducting fair and accurate evaluations. In our analysis, datasets with more balanced distributions score higher. For example, if a dataset contains 10 labels and 1 million total entries, an ideal distribution would allocate 100,000 entries per label. This perfect balance ensures that no bias exists when training learning algorithms.

    To assess balance, we calculate how much each label's distribution deviates from the ideal 10%. For example, if one label holds 15% of the data while another holds 5%, we calculate the sum of deviations and adjust for overrepresented and underrepresented labels. The total deviation is normalized and subtracted from a perfect score of 10. For a 5% imbalance (e.g., one label with 15% and another with 5%), the dataset would receive a score of 9.5, indicating a slight imbalance.

4.1.2 Synthesis

While valuable, the existing datasets in the Smart Home and Smart Office IoT domains have several limitations that hinder their effectiveness in addressing security and privacy challenges. These limitations include:

1. ***Limited Device Diversity***: Most datasets cover only a narrow range of device types, often excluding newer or less common IoT devices. This limited device diversity restricts the dataset's applicability for generalizing findings across a broad IoT ecosystem.

2. ***Inadequate Threat Coverage***: Many datasets concentrate on a limited selection of attack vectors, such as botnets or DDoS assaults, often neglecting other critical threats. This selective focus results in gaps in threat modeling, reducing the datasets' utility for training detection systems that must respond to a broader array of real-world vulnerabilities.

3. ***Absence of Real-World User Interaction***: Although some datasets simulate user interaction, the complexity and variety of real human interactions with IoT devices are rarely captured, impacting the accuracy of behavior-based intrusion detection systems.

4. ***Limited Accessibility to Raw Data***: Several datasets offer only flow-level or general statistics in CSV format without the availability of raw packet captures, which restricts the ability to conduct in-depth protocol analysis and detailed anomaly detection. This gap impedes researchers' capacity to understand network behavior and security vulnerabilities fully.

5. ***Imbalanced Attack Data***: Certain datasets exhibit insufficient instances of attack traffic, leading to an imbalanced representation that complicates the training of machine learning models for rare event detection. This imbalance can hinder the development of robust detection capabilities against less common attack types.

6. ***Synthetic Data Generation Bias***: Some datasets rely on synthetic data generation for specific traffic types, potentially introducing biases that diminish their applicability to real-world conditions. The resultant discrepancies may skew findings, limiting the validity of models developed using such data.

7. ***Reliance on Controlled Network Environments***: Some datasets are collected in highly controlled environments, which may not accurately represent the complexity of real-world network setups, such as those with varying security levels and background traffic and noises.

8. ***Traditional Network Overlap in IoT Research***: Many datasets in the IoT research field are based on traditional TCP/IP networks and their associated attacks, which have already been comprehensively explored within the broader computer network domain. This overlap not only diminishes the originality of the research but also limits its relevance to IoT-specific challenges. Many studies treat IoT networks as extensions of PC networks, often focusing primarily on their interactions with the Internet while neglecting scenarios where these networks operate autonomously.

   Furthermore, this approach overlooks the unique features of IoT systems, such as their limited computational resources, constrained communication channels, and the variety of device types involved. Consequently, such analyses fail to address the distinct security issues inherent to IoT environments, emphasizing the need for research methodologies that accurately reflect the complexities and specific requirements of IoT architectures. This gap calls for a dedicated exploration of IoT network dynamics, ensuring that security measures are effectively tailored to these systems' unique characteristics and challenges.

9. ***Insufficient Data Volume***: Some datasets do not provide a sufficiently large volume of data necessary for training robust machine learning models. The inadequacy in data quantity can significantly constrain the models' learning capabilities and overall performance.

10. ***Unrealistic Threat Scenarios***: Datasets often fail to replicate authentic attack environments, thereby undermining the efficacy of contemporary threat detection mechanisms. The lack of realism in simulated attacks can lead to inadequately trained models that are ill-prepared to handle actual security incidents.

11. ***Limited IoT-Specific Attacks***: Previous models overlook attacks targeting IoT protocols such as Z-Wave or Zigbee, favoring more generic network threats. This oversight significantly restricts the capacity to effectively identify and mitigate attacks exploiting the unique vulnerabilities of IoT communication protocols.

12. ***Outdated Threat Scenarios***: Dependence on outdated attack scenarios compromises the efficacy of detection models, rendering them less capable of identifying contemporary threats. This reliance can hinder the adaptive nature of security frameworks necessary to counter evolving attack vectors.

13. ***Lack of Comprehensive Feature Sets***: The inadequacy of feature sets in existing datasets significantly hampers the analytical depth and understanding of network behavior. A robust feature set is essential for researchers and intrusion detection systems, as it provides the necessary insights into the underlying dynamics of the data. By capturing diverse characteristics, including traffic patterns, device behaviors, and communication protocols, comprehensive feature sets enable the identification of meaningful correlations and anomalies.

14. ***Absence of Multi-Modal Data***: The failure to incorporate multi-modal data in prior research constrains the analysis of smart home network behaviors. Most previous studies have predominantly focused on network traffic data, neglecting other valuable data sources such as device logs and memory snapshots. Integrating multi-modal data would facilitate a more holistic understanding of network interactions, allowing for the detection of complex attack vectors and user behaviors that may not be evident from network data alone.

15. ***Labeling Inconsistencies***: Inconsistent or erroneous data labeling presents significant challenges for training and evaluating intrusion detection models. Accurate labeling is vital for supervised learning approaches, as it directly influences the model's ability to learn from data and generalize to new instances. When labels are incorrect or inconsistently applied, the resulting models may misinterpret normal behaviors as threats, leading to high false positive and false negative rates.

16. ***Metadata Deficiency***: The lack of adequate metadata accompanying datasets creates substantial barriers to understanding the context and limitations of the data. Metadata provides crucial information regarding the conditions under which the data was collected, including the environment, device configurations, and any pre-processing steps applied. Without this contextual information, researchers may misinterpret findings or overlook critical factors that could influence network behavior.

17. ***Lack of Support for Multi-Protocol Analysis***: Many datasets focus exclusively on specific communication protocols, neglecting other protocols widely utilized within IoT environments. This limited perspective restricts the ability to analyze and understand interactions in mixed-protocol settings, where devices may communicate using multiple protocols simultaneously. The absence of multi-protocol analysis can hinder the detection of vulnerabilities that arise from the complexities of these interactions, ultimately compromising the effectiveness of security strategies.

These limitations underscore the pressing need for more comprehensive and versatile datasets to meet the demands of contemporary IoT research. Recognizing this, this work aims to address all the challenges outlined in this analysis.

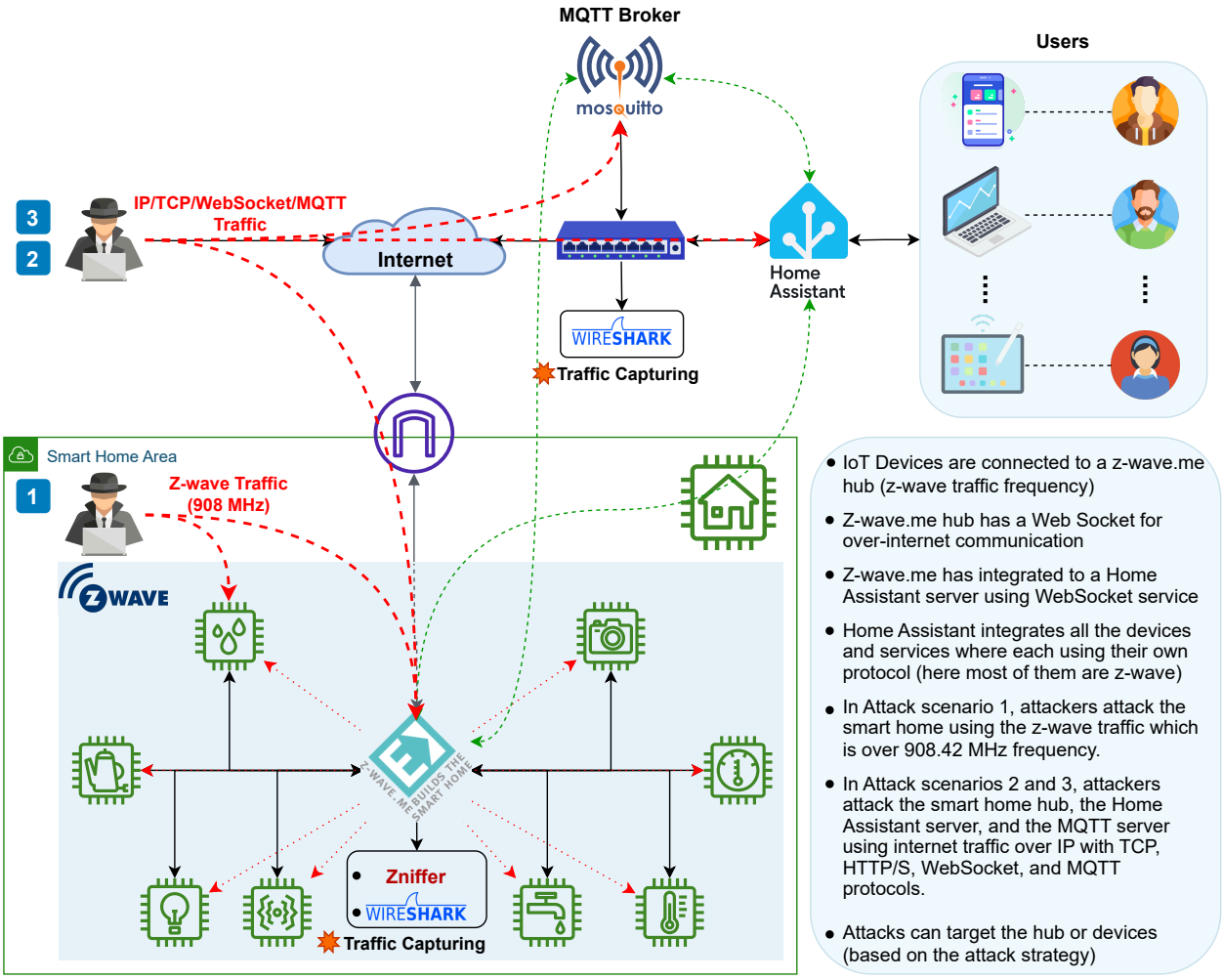


Figure 36: Testbed Architecture

4.2 Dataset Testbed Setup

This subsection presents a detailed description of the newly created dataset. This includes the Testbed Architecture, Threat Scenarios, Data Capturing, and CSV generation. The final created dataset is named "BCCC-IoT-Zwave-2025" and is publicly available on the BCCC website [107].

4.2.1 Architecture

The testbed architecture presents a comprehensive environment for smart home security analysis, designed to replicate real-world scenarios by integrating multiple layers of communication and devices. It incorporates both wireless and wired communication protocols, allowing for the simulation of various attack vectors and the evaluation of defense mechanisms. The testbed architecture is shown in Fig. 36. This architecture follows established standards seen in previous research, while also introducing additional capturing and analysis capabilities to provide deeper insights into system vulnerabilities.

At the core of the testbed is the network layer, designed to handle both wireless and internet-based communications. The architecture strategically links IoT devices using different protocols, allowing for the seamless transmission of data and control signals across multiple mediums. One of the standout features of this testbed is its ability to integrate disparate communication types, such as low-power wireless protocols (z-wave in this case) and internet protocols like TCP, MQTT, HTTP/S, and WebSockets. This hybrid network architecture mirrors the complexity of a real-world smart home. It is critical to understanding the attack surface in a typical smart home setup, where devices are often from different vendors and use different protocols yet must coexist and interact.

A key advantage of this testbed lies in its bidirectional data flow: IoT devices communicate with the central hub using z-wave protocol under the 908.42 MHz, while external services, users, and cloud applications interact with the hub over the internet. This setup is crucial for assessing the security implications of both local and external attacks. By capturing traffic at multiple points, including within the local network and at the gateway to the internet, the testbed offers comprehensive insights into how vulnerabilities can be exploited at various levels, from device-to-hub communication to remote attacks over the internet.

The capturing system is strategically placed to collect traffic from multiple points in the network, including local device-to-hub and hub-to-device interactions and hub-to-internet communications. This approach ensures a comprehensive dataset for evaluating not just individual device security but also how the system as a whole responds to different types of attacks.

Moreover, the testbed captures raw traffic data and metadata that provides context for device behavior. This allows for a deeper analysis of both legitimate and malicious activities. The testbed architecture is particularly effective at identifying patterns in anomalous behavior, thanks to the integration of advanced capturing mechanisms that monitor traffic across multiple layers. This supports the development of more sophisticated intrusion detection systems, enabling researchers to identify correlations between device activity and network traffic anomalies.

The testbed also integrates simulated attacks based on known vulnerabilities from various smart home devices and protocols. Attack simulations in our architecture can target different points in the system—whether device-level vulnerabilities, such as weak encryption in Z-Wave communication, or application-level threats, like exploiting flaws in cloud or external services APIs. By incorporating a wide range of attack vectors, the testbed supports a thorough investigation of smart home security issues, from man-in-the-middle (MITM) attacks to denial of service (DoS) attacks.

Lastly, the testbed is designed to be extensible. This modularity allows researchers to add or replace devices, protocols, or services as needed, ensuring that the testbed remains relevant in the face of new technologies and evolving security concerns. This adaptability is a core characteristic of advanced testbeds in the field, as maintaining flexibility is key to ensuring that the architecture can accommodate future research and development.

In summary, the testbed architecture is a multi-protocol, multi-layer system designed to emulate the complex interactions in smart home environments. Its diverse communication standards, comprehensive traffic capturing, attack simulations, and modular design make it a powerful tool for understanding and mitigating the security challenges inherent in modern smart homes. This architecture aligns with the trends and best practices observed in contemporary research testbed, while introducing unique features that enhance its analytical capabilities.

4.2.2 Devices Taxonomy

In addressing the fourth problem within our research, which pertains to data collection and diversity, we thoroughly analyzed prior works to identify the essential smart home devices required for a comprehensive testbed. The result is a taxonomy of smart home devices, as illustrated in Fig. 37, which serves as a structured framework for categorizing and understanding the roles of various devices within smart home environments.

Our taxonomy categorizes devices into functional groups critical for capturing the full spectrum of smart home activities. These categories encompass controllers, sensors, and actuators, each of which plays a pivotal role in ensuring a smart home's smooth operation and security. Controllers, such as hubs, gateways, and routers, are central nodes facilitating communication between devices and external services. Sensors monitor the environment and capture data, including motion detection, environmental conditions (temperature, humidity, and air quality),

and security metrics like door movement and smoke detection. Actuators, such as smart locks, bulbs, and switches, respond to sensor data by controlling physical devices in the environment.

One of the primary strengths of this taxonomy is its protocol-agnostic nature. By focusing on devices' functional roles rather than their communication protocols, we ensure that this taxonomy applies universally across a wide range of smart home architectures. This approach is particularly important in modern smart home ecosystems, where devices using protocols such as Z-Wave, Zigbee, Wi-Fi, and Bluetooth must coexist and interact seamlessly. This protocol-agnostic taxonomy allows for greater flexibility in device selection and system integration.

Our research emphasizes ensuring that at least one device from each taxonomy category is included in any smart home architecture. This comprehensive coverage is essential for capturing real-world smart homes' diverse interactions and behaviors. We can generate datasets that reflect the full scope of potential smart home operations by considering security-related devices (e.g., smart cameras, alarms, and door locks) and environmental sensors (e.g., air quality and water leak detectors). This diverse representation of device types is necessary for generating accurate and robust datasets, as it allows for analyzing various operational contexts and identifying potential security vulnerabilities across the ecosystem.

Furthermore, this taxonomy provides a foundation for future work in smart home and office research, including developing intrusion detection systems (IDS), behavior profiling models, and security frameworks. By mapping devices to specific functional categories, we can better understand the typical behavior of each device type and how deviations from this behavior might indicate malicious activity.

In summary, our taxonomy of smart home devices is a critical component of our research, serving as both a framework for organizing smart home devices and a tool for guiding dataset creation. Its protocol-agnostic design ensures broad applicability across different smart home architectures, and its inclusion of essential device categories guarantees comprehensive coverage of smart home functionalities. By leveraging this taxonomy, we ensure that our datasets represent real-world smart homes, enabling more accurate and meaningful analyses of smart home security and performance. This taxonomy will continue to serve as a valuable resource for researchers and practitioners seeking to advance the state of smart home technology and security.

4.2.3 Smart Home Protocol Selection

The selection of an appropriate communication protocol is critical for the design and efficacy of smart home systems, particularly when establishing a controlled testbed for research purposes. In this study, we have focused exclusively on the Z-Wave protocol. This decision is grounded in systematically evaluating the protocol's characteristics, security features, and applicability to the smart home environment. By concentrating solely on Z-Wave, we aim to facilitate a detailed exploration of its functionalities and vulnerabilities, ultimately contributing to a deeper understanding of smart home security dynamics.

The Z-Wave protocol is widely recognized in the smart home industry for several compelling reasons. First, it operates on a low-power, low-bandwidth model, making it particularly suitable for battery-operated devices commonly found in smart homes, such as sensors and locks. This focus on energy efficiency is crucial for ensuring the longevity of devices within a home environment, where frequent battery replacements can be impractical and costly. The protocol's mesh networking capabilities further enhance its utility; devices can communicate directly with one another, extending coverage and increasing reliability through a self-healing network structure. Such characteristics position Z-Wave as an ideal choice for our research.

Moreover, Z-Wave provides a strong security framework, including multiple security levels, notably S0 and S2.

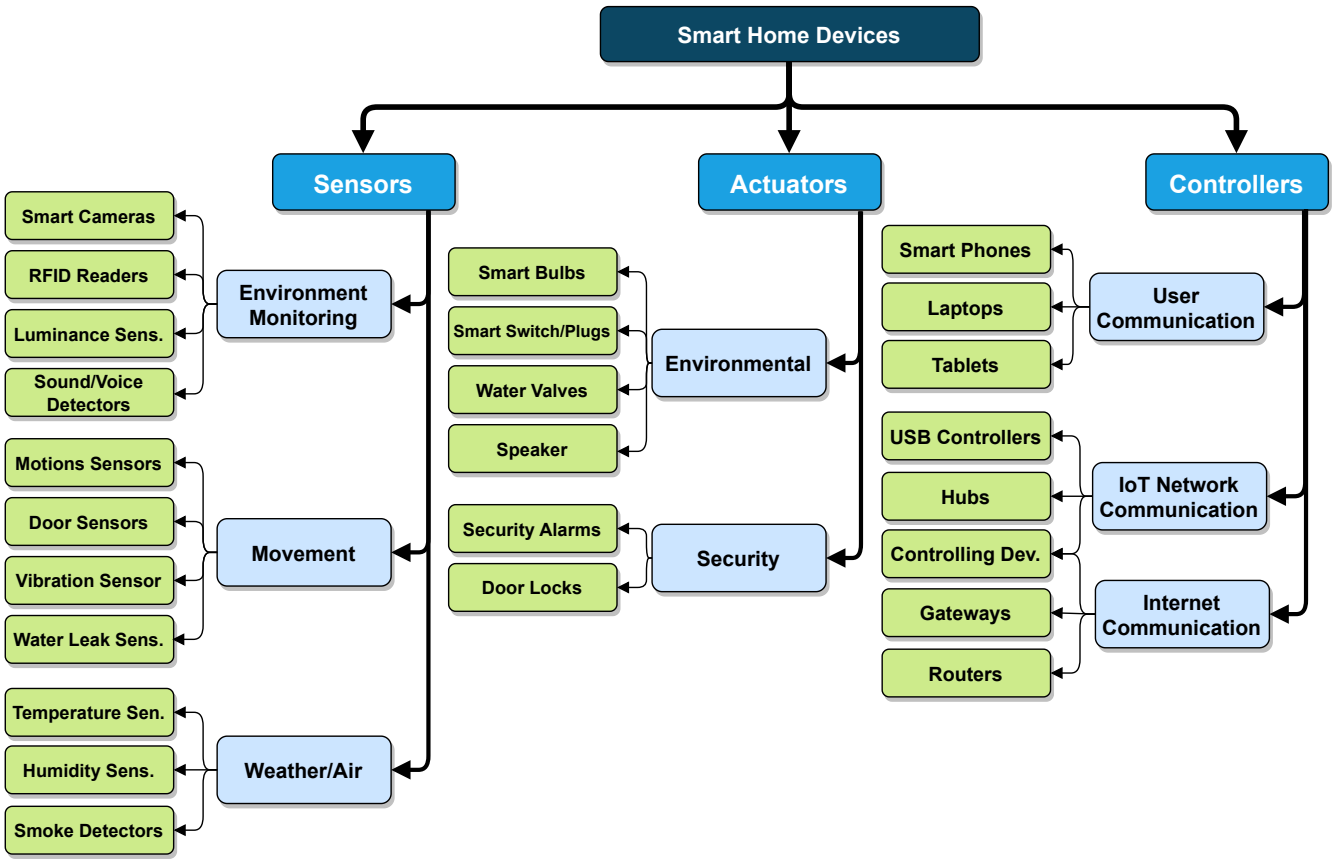


Figure 37: Taxonomy of smart home devices categories

The S2 framework, which employs advanced encryption and secure key exchange mechanisms, represents the current standard for secure communications within Z-Wave networks. In contrast, while less secure, the older S0 framework allows us to examine the spectrum of security implementations across different devices. This diversity in security protocols enables us to investigate vulnerabilities specific to the Z-Wave ecosystem and contributes to the richness of our dataset.

While other smart home communication protocols, such as Zigbee, Wi-Fi, and Bluetooth, are prevalent, our exclusive focus on Z-Wave allows us to create a homogeneous dataset free from the complexities inherent in multi-protocol environments. Such complexities could obscure the analysis of protocol-specific behaviors and vulnerabilities, limiting our findings' depth. The decision to focus exclusively on the Z-Wave protocol in our testbed is informed by several critical factors that highlight the limitations of alternative smart home protocols such as Wi-Fi and Zigbee.

Wi-Fi is a widely adopted communication protocol known for its high data transfer rates, but it presents significant drawbacks for smart home applications. One of the primary concerns is its high power consumption, which makes it less suitable for battery-operated devices commonly used in smart homes, such as sensors and locks. Wi-Fi networks typically support a limited number of connected devices—usually around 32 per access point. Modern smart homes often incorporate more devices than this limitation allows, so network congestion can quickly become an issue, leading to reduced performance and reliability. The potential for interference from multiple devices vying for bandwidth in a household can further complicate matters, particularly in environments where numerous Wi-Fi networks coexist. In contrast, Z-Wave offers a higher capacity, allowing connections of up to 232 devices per network. It is a more scalable solution for the diverse devices commonly found in smart homes.

While also designed for low-power, low-data-rate applications and supporting mesh networking capabilities, Zig-

bee poses its own challenges. It operates within the same 2.4 GHz frequency band as Wi-Fi, which can lead to substantial interference, particularly in urban environments where the density of Wi-Fi networks is high. In settings such as downtown areas, where numerous devices are constantly transmitting signals, the likelihood of signal degradation increases significantly. This interference can impact the performance and reliability of Zigbee networks, making them less dependable in smart home and smart office scenarios where consistent communication is essential. Moreover, while Zigbee can support more devices than Wi-Fi, it can still fall short of Z-Wave's flexibility and resilience in a smart home ecosystem.

By concentrating our research on Z-Wave, we can take advantage of its optimized energy efficiency, robust mesh networking capabilities, and superior resistance to interference. These attributes enhance the overall performance of smart home systems and ensure the longevity and reliability of the devices within the network. The Z-Wave protocol is adept at maintaining stable connections in environments with numerous competing signals, providing a clear advantage over Wi-Fi and Zigbee. Ultimately, focusing solely on Z-Wave facilitates a more comprehensive understanding of its operational dynamics, vulnerabilities, and security mechanisms, vital for advancing smart home technology and ensuring user safety. This focused approach ensures that the resulting dataset clearly represents Z-Wave's communication dynamics and security characteristics.

In summary, the selection of the Z-Wave protocol as the exclusive communication standard in our testbed is underpinned by its widespread adoption, robust security features, and suitability for smart home applications. By concentrating on Z-Wave, we aim to generate a homogeneous dataset that captures the intricacies of its communication dynamics and security vulnerabilities. This focused approach will lay the groundwork for future research, exploring the complexities of multi-protocol environments while ensuring a thorough understanding of Z-Wave's capabilities and challenges. While our current focus on the Z-Wave protocol provides a solid foundation for understanding its specific security dynamics, future research endeavors will extend the testbed to encompass other communication protocols such as Zigbee, Wi-Fi, and Bluetooth.

4.2.4 Devices Selection

Based on the prepared device taxonomy, we have procured multiple devices for each device type from different vendors, as outlined in Table 6. The selection of multiple devices per type ensures a comprehensive understanding of the functionality, performance, and security characteristics of each device type in our testbed. By procuring several instances of the same device type, such as water sensors or smart locks, we can explore variations in device behavior arising from differences in manufacturing tolerances, firmware versions, or operational algorithms. This enables us to analyze how different models within the same category respond to identical conditions, thereby uncovering potential vulnerabilities that might be overlooked if only a single device were used.

Moreover, having multiple devices of the same type allows us to place each under different conditions or in various locations within our simulated smart home environment. This broader approach facilitates a more robust evaluation of device performance and reliability, as we can assess how each unit operates under varying scenarios, such as network stress or environmental factors. Ultimately, this strategy enhances the richness of our dataset, ensuring that our findings accurately reflect the diverse operational landscapes encountered in real-world smart home environments. Furthermore, this comprehensive analysis aids in creating a more general profiling for each device type, allowing us to better understand their capabilities, performance metrics, and potential security vulnerabilities.

Moreover, to enhance the robustness of our study further, we have included multiple devices from various manufacturers. This inclusion is crucial for capturing the variability in device performance and security implementations.

Different vendors often introduce proprietary enhancements and variations in implementing Z-Wave standards, leading to distinct operational behaviors. For instance, some devices may utilize the more advanced S2 security features, while others may rely on the less secure S0 implementation. This diversity reflects real-world scenarios in which smart home environments consist of devices from various manufacturers, thus allowing us to evaluate how these variations impact overall security and functionality.

In addition to vendor diversity, our selection process focused on including devices commonly found in consumer smart homes. Smart bulbs, cameras, door locks, and environmental sensors (e.g., temperature and humidity) were prioritized due to their widespread adoption. Including these devices ensures that our research remains relevant to contemporary smart home security concerns. Moreover, some devices were chosen specifically for their known vulnerabilities, as documented in previous research, allowing us to simulate specific attack vectors and evaluate the effectiveness of security mechanisms.

We also considered the ease of integration when selecting devices. For example, some devices were chosen because of their open APIs or compatibility with widely used smart home platforms, such as Home Assistant or Amazon Alexa. This allows for more flexible control of devices in our testbed and the ability to simulate interactions with third-party services, which are common in real-world smart homes. The ability to test how third-party integrations impact security is crucial, as vulnerabilities often arise from these interactions.

Furthermore, we have ensured that the selected devices represent basic and advanced functionalities within their respective categories. For example, in the case of security devices, we have included standard motion detectors and more advanced smart cameras capable of facial recognition. Similarly, within the actuator category, we have incorporated simple, smart plugs and more complex devices like smart plugs, which can dynamically send the voltage, watts, and another electricity usage. Additionally, for the Z-Wave protocol, we have ensured the inclusion of devices with varying security levels, specifically incorporating both S0 and S2 to analyze the protocol's security features comprehensively. By including devices with varying levels of complexity, we can study the security implications of both basic and advanced functionalities, reflecting the diverse capabilities of modern smart homes.

Smart Home Hub

To facilitate the integration and analysis of Z-Wave devices within our smart home testbed, we selected the Z-Wave.Me hub as the central controller. This decision was motivated by its unique capabilities tailored to the specific requirements of our research. Unlike other prominent hubs, such as Samsung SmartThings, the Z-Wave.Me hub is explicitly designed to support the entire spectrum of Z-Wave devices. This ensures seamless compatibility and robust performance, which are critical for the diverse set of devices employed in our study.

A pivotal reason for this selection was the Z-Wave.Me hub's ability to capture and save raw Z-Wave packets transmitted between devices. This feature is indispensable for our research, as it allows us to conduct an in-depth analysis of the communication protocols, including packet structure, timing, and security mechanisms. Such detailed insights are not typically accessible through more mainstream hubs, which prioritize user-friendly interfaces over advanced diagnostic capabilities.

Furthermore, the Z-Wave.Me hub offers unparalleled flexibility in managing Z-Wave networks, including advanced features like network topology visualization and real-time debugging. These capabilities are particularly beneficial for our study, as they enable precise monitoring and control of the testbed environment. By leveraging these features, we can systematically evaluate device interactions, identify potential vulnerabilities, and simulate various attack scenarios with high accuracy.

In contrast, popular hubs like Samsung SmartThings and others often cater to a broader range of protocols and

prioritize interoperability with non-Z-Wave devices. While this generalist approach enhances their appeal for consumer applications, it introduces limitations in specialized research contexts. For instance, such hubs may abstract or aggregate Z-Wave communications, thereby obscuring critical details needed for a thorough protocol analysis. This abstraction would have posed significant challenges to our goal of comprehensively evaluating Z-Wave security and performance.

In summary, the Z-Wave.Me hub's specialized focus on Z-Wave technology, coupled with its advanced packet-capturing and network management features, made it the ideal choice for our research. By selecting this hub, we ensured the fidelity and depth of our analysis, enabling us to uncover nuanced insights into the behavior and vulnerabilities of Z-Wave devices within a smart home environment.

In conclusion, the selection of multiple devices from various vendors in our testbed increases the realism, coverage, and relevance of our research. By considering vendor diversity, device functionality, protocol compatibility, and known vulnerabilities, we ensure that our testbed captures real-world smart homes' complex interactions and security challenges. This approach aligns with best practices in the field and provides a solid foundation for generating comprehensive, meaningful datasets that will contribute to advancing smart home security research.

4.2.5 Benign Traffic

This section details the process of generating benign traffic based on natural human interactions within a real IoT smart home environment. The traffic was collected using devices deployed in a lab environment, with specific placements and repeated actions designed to simulate real-world scenarios.

To ensure the data's authenticity, devices were installed strategically to reflect actual use cases. For example, two door sensors were mounted on the lab's main and cabinet doors, respectively. A shock/tilt sensor was also placed on the lab door to monitor vibrations and tilting actions. Motion sensors were distributed across the lab to capture different types of movement. These included one sensor near the entrance, another near the network rack, and additional sensors near commonly used equipment, such as laptops, to observe varying motion patterns and device interactions. This diversity in device placement allowed us to capture a wide range of behaviors for each sensor type.

The data generation involved repeating everyday actions to ensure the captured data mirrored real-world scenarios. For instance, door locks were operated regularly as students arrived and left the lab, reflecting typical daily activity. Water sensors were placed in locations where exposure to water was predictable. One sensor was positioned near an area where water spilled frequently in the mornings when people made coffee. At the same time, another was installed near a sink, where water was usually present throughout the day, except during the night when the lab was unoccupied.

We replicated daily routines for smoke detectors by triggering them at specific times. These tests used different types of smoke, including incense, vapes, cigarettes, burning paper, and candles, to evaluate the sensor's ability to detect varying smoke types such as CO and CO2. A siren was also set to sound at regular intervals each day to simulate rings or emergency alerts, further contributing to the authenticity of the generated data.

Additionally, automated activities were integrated into the setup to mirror real-world IoT applications. For example, motion sensors were linked to a lamp via Home Assistant, ensuring that the lamp would automatically turn on whenever motion was detected for 30 seconds. Similar automations were created for other device interactions, reflecting how IoT systems manage routine activities.

We also tested similar device types in different conditions to capture a range of operational behaviors. For instance,

temperature sensors and smart plugs were used in multiple locations to evaluate their performance under varied conditions. Multiple smart plugs connected to high-power-consuming devices such as a fan, toaster, and boiler were tested for disconnections resulting from excessive power consumption. Water sensors and motion detectors were similarly placed in different environments and subjected to daily experimental variations, ensuring a broad dataset of behaviors.

We encountered typical challenges in real-world smart home setups throughout the data collection period. For example, the hub was restarted multiple times due to device inclusion errors and malfunctions that made it unresponsive. Additionally, some devices required battery replacements due to the extended duration of the experiments, and a few devices had to be excluded and re-included in the network after malfunctioning. These interruptions mirrored common issues in actual IoT environments and added to the realism of the captured data.

Overall, the careful placement of devices, the repeated real-world interactions, and the periodic adjustments made during the experiment resulted in a comprehensive and authentic dataset. We endeavored to simulate a fully operational smart home environment with realistic actions, environmental conditions, and challenges, ensuring the benign traffic accurately represents typical IoT smart home behaviors. Detailed information regarding the devices, their connections, and the experimental setup can be found in Table 6, which provides an overview of the testbed devices.

4.2.6 Threat Scenarios

In creating the dataset, a primary focus was placed on identifying attack surfaces based on the designed architecture, followed by executing various potential attack scenarios. As illustrated in Fig. 36, the architecture encompasses five main attack surfaces.

The first surface is the **Z-Wave Hub**, which functions as a crucial intermediary, linking various smart home devices to the internet (for smart home monitoring and management by the users). This hub manages local communications and serves as a web service (utilizing Apache), exposing it to a range of internet-based threats.

The second surface is the **WebSocket service** of the Z-Wave Hub, which further facilitates communication between the hub and external tools, such as Amazon Alexa and Home Assistant. This service is essential for remote control and automation of smart devices, yet it also presents significant security challenges, as it is susceptible to exploitation by malicious actors.

The third surface is the **wireless communication channel** established via the Z-wave protocol, which facilitates the devices' communication with the Z-wave Hub. The inherent vulnerabilities of wireless protocols can be exploited, posing risks to the integrity and security of the connected devices.

The fourth surface is the **MQTT server**, which acts as a messaging broker within the smart home ecosystem, coordinating communication between devices and users. As a result, it also becomes a target for various attack vectors, including those aimed at manipulating device behavior or intercepting data.

The last surface involves the integration of **third-party applications** like Amazon Alexa and Home Assistant, which seamlessly connect and manage various devices (using different protocols) within a smart home environment. These applications often rely on interconnected services, and their exposure increases the likelihood of security breaches, particularly if proper safeguards are not in place.

To understand the range of possible attacks, we can categorize them based on the protocols and services involved. Internet-based attacks primarily target the Z-Wave Hub Webservice, WebSocket service, MQTT server, and associated third-party applications, all operating on the TCP protocol. Each attack vector may employ specific appli-

cation layer protocols, allowing for various methodologies such as HTTP-based, WebSocket-based, MQTT-based, and TCP-based attacks.
The techniques used in these attacks are diverse and can include:

- ***DDoS (Distributed Denial of Service)***: This attack aims to overwhelm a service by flooding it with excessive traffic, rendering it unavailable to legitimate users. Attackers utilize a network of compromised devices to send a massive volume of requests to the target, exhausting its resources. A successful DDoS attack can lead to system unavailability, preventing users from controlling their devices and potentially compromising home security.

- ***Botnet Attacks***: A botnet is a collection of interconnected devices controlled by an attacker to perform various malicious activities. Attackers infect devices with malware, allowing them to be remotely controlled for tasks like sending spam or conducting DDoS attacks. A botnet attack targeting a smart home can lead to unauthorized access and control over connected devices, resulting in increased energy consumption and possible privacy breaches.

- ***Brute Force Attacks***: This technique systematically tries all possible password combinations to gain unauthorized access to a system. Attackers use automated tools to attempt login with various username-password pairs until the correct one is found. Successful brute force attacks may grant attackers unauthorized access, allowing them to manipulate devices, steal personal information, and disrupt household operations.

- ***Information Gathering***: This involves collecting data about a target system to identify potential vulnerabilities. Techniques may include scanning networks, probing services, and examining system configurations. Effective information gathering can give attackers the insights needed to launch targeted attacks.

- ***Vulnerability Exploitation***: This attack leverages known vulnerabilities in software or systems to gain unauthorized access or cause damage. Attackers identify and exploit flaws in applications or protocols, often using automated tools to find and take advantage of these weaknesses. Exploiting vulnerabilities in a smart home can allow attackers to gain control over the system, leading to unauthorized surveillance, device manipulation, and potential safety hazards for the household.

In parallel, Z-Wave-specific attacks also present unique challenges and can be categorized into various types, including:

- ***Replay Attacks***: This involves capturing valid data transmissions and re-sending them to trick a system into executing commands. An attacker records a valid communication session and later retransmits it, potentially bypassing authentication mechanisms. Replay attacks can trick the system into executing unauthorized commands, leading to unintended actions on connected devices, such as unlocking doors or turning on appliances without user consent.

- ***DDoS Attacks***: Similar to the Replay attacks, but specifically targeting Z-Wave communications to overwhelm the hub. Attackers flood the Z-Wave network with excessive requests, causing legitimate communication to be disrupted. This can incapacitate smart home systems, disrupt legitimate communication, and render devices inoperable, preventing users from accessing critical functionalities like lighting and security systems. Additionally, with the hub being unresponsive, the devices will continue to send packets to find the hub, leading to battery depletion.

- ***Vulnerability Exploitation***: Exploiting specific weaknesses in the Z-Wave protocol or implementations to gain unauthorized access. Attackers may leverage known vulnerabilities in the Z-Wave specifications or device firmware to compromise security. Successful exploitation can allow attackers to control devices or intercept communications and potentially disrupt household operations.

- ***Jamming***: This attack disrupts wireless communication between devices. Attackers emit radio signals on the same frequency as the Z-Wave devices, preventing legitimate communications. Jamming can disrupt communication between the smart home hub and connected devices, rendering the entire system inoperable during the attack and leaving users without control over their environment.

- ***Fuzzing***: This technique sends random data to devices to discover vulnerabilities. Attackers systematically introduce unexpected inputs to observe how devices respond, potentially revealing weaknesses. Fuzzing can expose critical vulnerabilities, leading to potential compromises in device security.

- ***Command Injection***: This involves sending malicious commands to a device or application to execute unintended actions. Attackers craft specific inputs that exploit a lack of input validation, allowing them to manipulate device behavior. Successful command injection can lead to unauthorized control over devices, resulting in unauthorized control and significant disruption of services, such as turning off security systems or altering temperature settings.

Table 7 provides a detailed schedule of attacks and their corresponding targets, serving as a roadmap for understanding the dynamics and implications of these threat scenarios. We utilized a setup consisting of 8 Raspberry Pis to execute the network attack scenarios. A laptop running Ubuntu 20.04 was employed for the Z-Wave-specific attacks, equipped with various USB sticks, boards, and antennas to transmit the relevant signals. Detailed information about the devices used in the experiments can be found in Table 6, which outlines the testbed devices.

4.2.7 Data Capturing

The foundation of any network analysis begins with data capturing. In our research, we employed a dual-source data-capturing strategy to acquire a diverse range of traffic from two distinct types of networks: the IP-based internet network traffic and Z-Wave network traffic. These two sources required unique capturing techniques due to their vastly different nature. Where IP-based protocols like TCP, HTTP/S, WebSocket, and MQTT dominate traditional internet traffic, Z-Wave operates on the 908.42 MHz frequency, necessitating specialized tools and approaches for effective data acquisition.

Regarding IP network traffic capture, the primary and most critical source of our data collection was the Z-Wave hub machine. To initiate the capture process, we first utilized Wireshark to monitor and record traffic from the designated network interface card of the Z-Wave hub. Traffic was saved into PCAP files in manageable chunks of 100 megabytes, facilitating smoother analysis. Wireshark was initially selected due to its capability for live monitoring, enabling real-time traffic oversight and the ability to detect any network issues promptly. However, during attack simulations, we encountered performance bottlenecks; Wireshark struggled with the heavy traffic loads, leading to the risk of data loss due to unresponsiveness. To mitigate this, we switched to Tcpdump, a more efficient, command-line-based tool that consumes minimal resources. Tcpdump allowed us to maintain the same setup of capturing 100-megabyte PCAP files, proving to be a more scalable solution for handling large volumes of network data without the overhead of a graphical interface.

In parallel, we captured the internet network traffic of the network hub, which linked the MQTT broker and the Home Assistant server to the internet. For this purpose, a network TAP (Test Access Point) was deployed, copying all traffic that flowed through the network hub to the capturing machine. Wireshark was also employed here, effectively logging all ingress and egress traffic to build a comprehensive dataset of the MQTT, WebSocket communication, and other IP-based protocols operating within the smart home network.

Z-Wave traffic presented a unique challenge, as conventional IP network tools are not typically designed to capture these protocol signals as they are in different frequency ranges. To overcome this, we leveraged the Z-Wave.me hub's Zniffer feature, which captures the last 5,000 packets of Z-Wave traffic. We further automated this process by scripting a solution that regularly extracted and saved these packets, organizing them into labeled files based on the timestamp to ensure no loss of valuable Z-Wave data.

A key characteristic of our dataset is its comprehensive nature. When setting up the testbed, we captured all traffic, including device inclusions and exclusions, normal operational behaviors, and unexpected anomalies. For instance, device inclusion errors or hub restarts were captured, providing insight into typical smart home environment scenarios. These interactions highlighted important aspects of network behavior, such as how Z-Wave nodes are reassigned to new node IDs upon re-inclusion, all of which are carefully documented for further analysis.

We were aware of potential capturing errors, especially during high-traffic periods or system faults. We constantly monitored the testbed devices to prevent this, ensuring minimal data loss by detecting and fixing any capture failures within hours. Even on weekends and holidays, we stayed alert, fixing and restarting related processes and services when necessary.

The data-capturing process officially commenced on June 26, 2024, and continued until Nov. 15, 2024. From September 6, we expanded our scope to include MQTT broker and Home assistant traffic. On September 23, we began executing attack scenarios, aligning them with the schedule provided in Table 7. The timestamps of these attacks were crucial in labeling the data, providing a clear distinction between benign and attack conditions for further analysis.

Our dataset's richness extends beyond the captured traffic volume and includes temporal patterns and device interaction insights. For instance, Figures 50, 51, 52, 53, 54, and 55 illustrate the average number of packets transmitted by each device per week, which includes benign days, IP attack days, and Z-Wave attack days. Further temporal analysis in Figures 44, 45, 46, 47, 48, and 49 breaks down traffic flow during specific time windows, revealing distinct patterns in packet transmission from devices like door locks and water sensors, with co-occurrences detected across specific timeframes.

Additionally, Figures 38, 39, 40, 41, 42, and 43 compares packet transmission between weekdays and weekends during benign periods, showing how device behavior fluctuates depending on the time of week.

This data is further segmented by device comparisons in figures 56, 57, 58, and 59, which categorizes packet counts per device, illustrating the similar behavior of devices in the same group. For example, all the devices in group 8 (Fig. 59 are power-connected devices.

4.2.8 Traffic Analyzers

Converting raw traffic data into structured, analyzable formats in network security and analysis is crucial for applying machine learning and deep learning techniques. Traffic analyzers serve this role by parsing intricate packet-level data and transforming it into meaningful datasets that facilitate deeper analysis. Given the diverse nature of network environments, different analyzers are required to handle various types of traffic effectively. This section

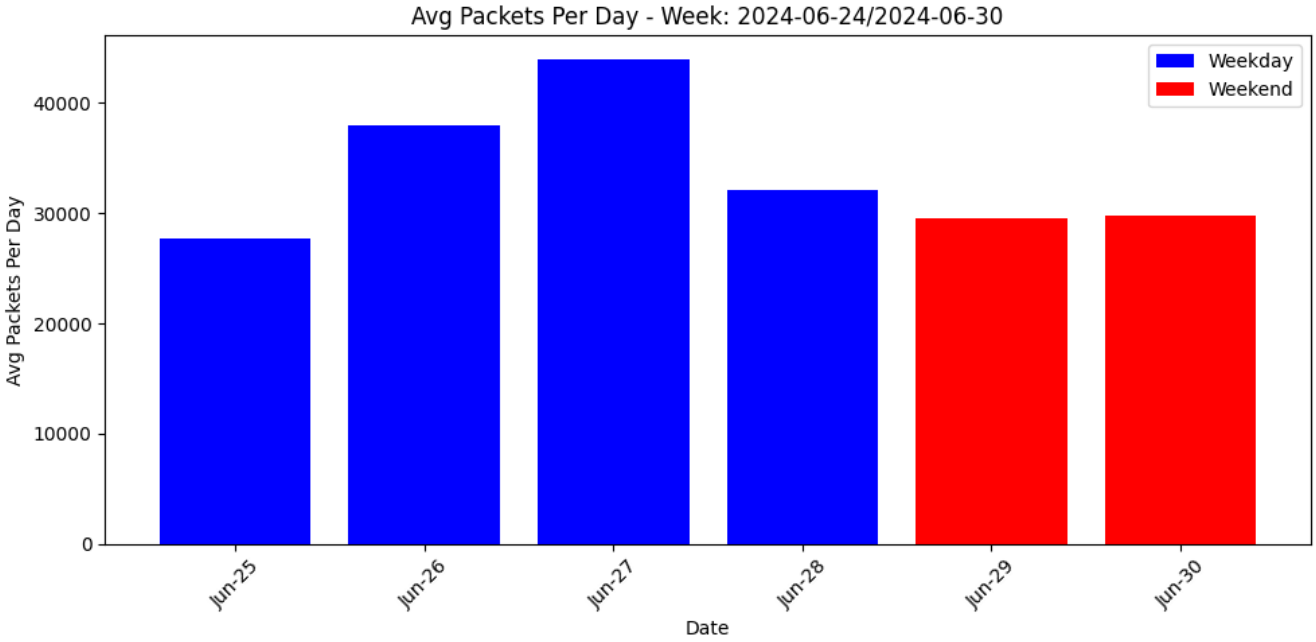


Figure 38: Week one of capturing and setup. Includes benign traffic.

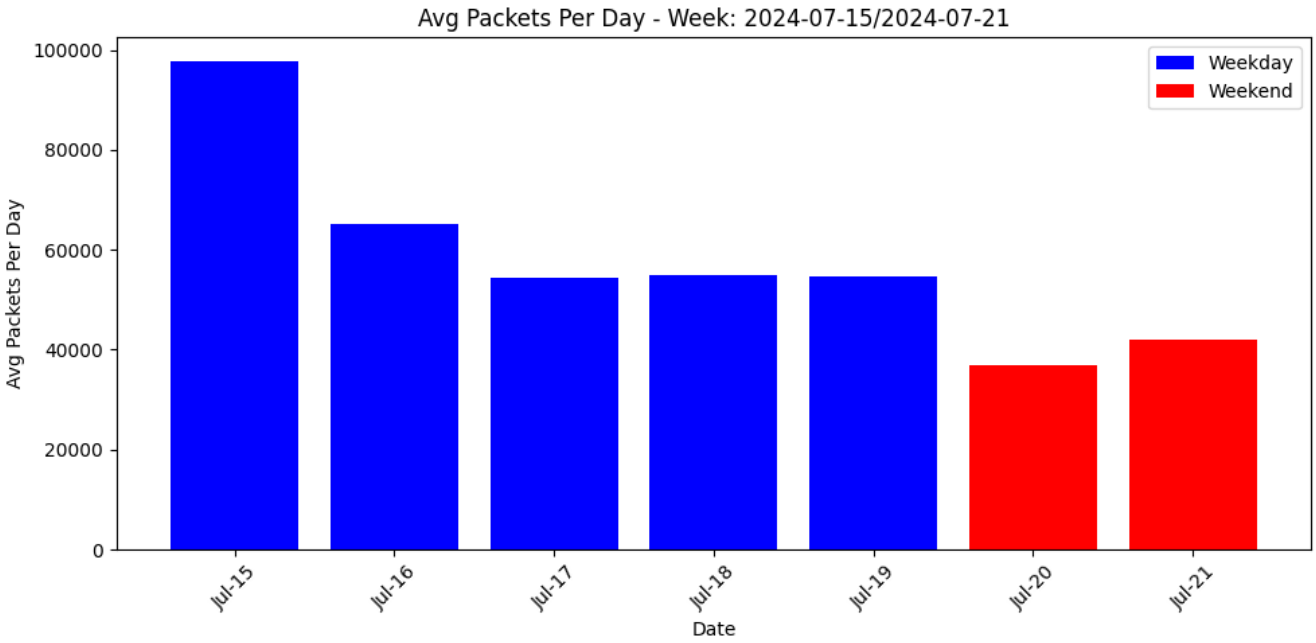


Figure 39: Week four of capturing and setup. Includes benign traffic.

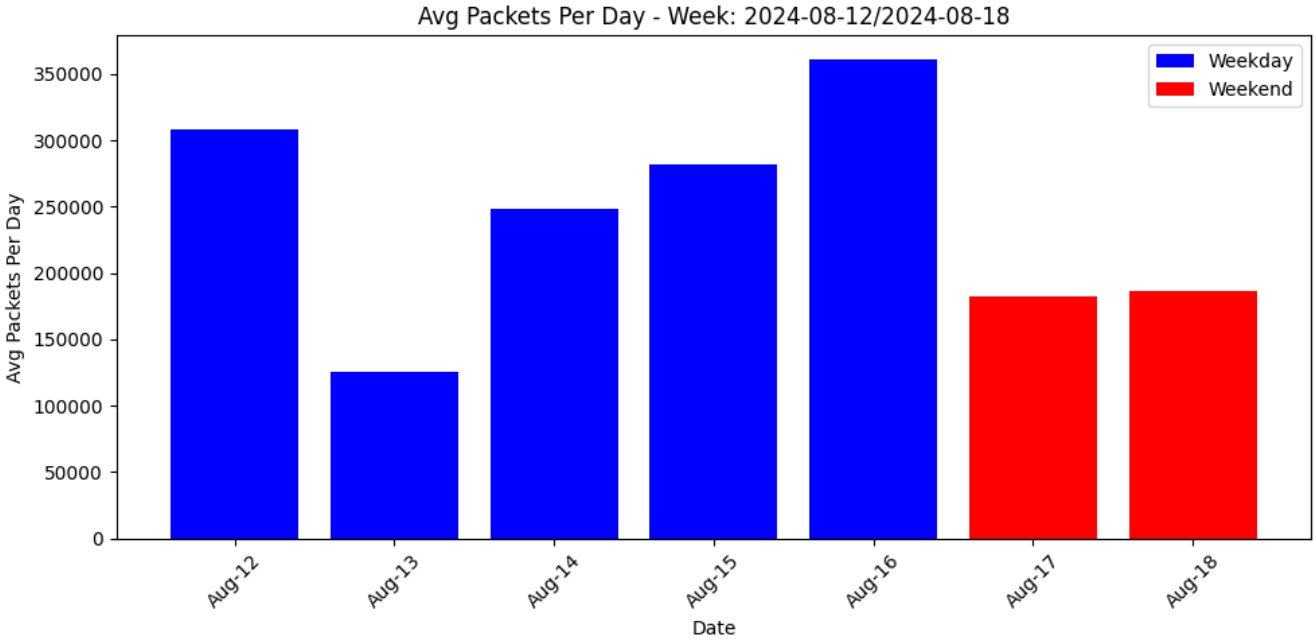


Figure 40: Week eight of capturing and setup. Includes benign traffic.

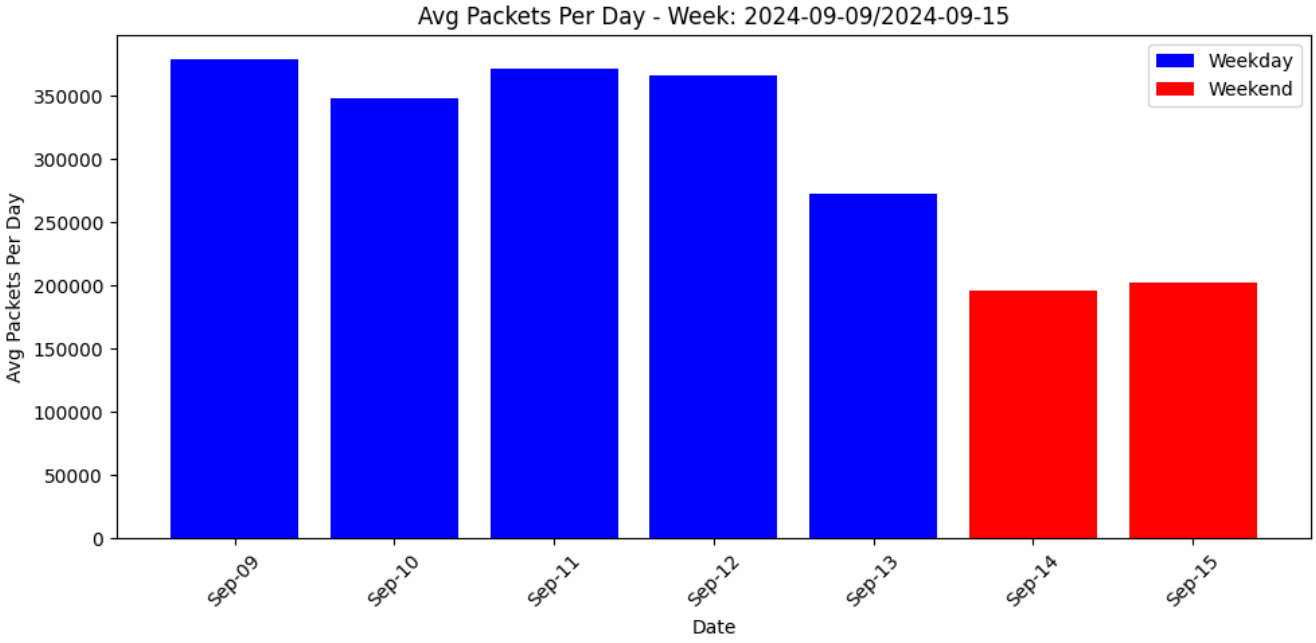


Figure 41: Week twelve of capturing and setup. Includes benign traffic.

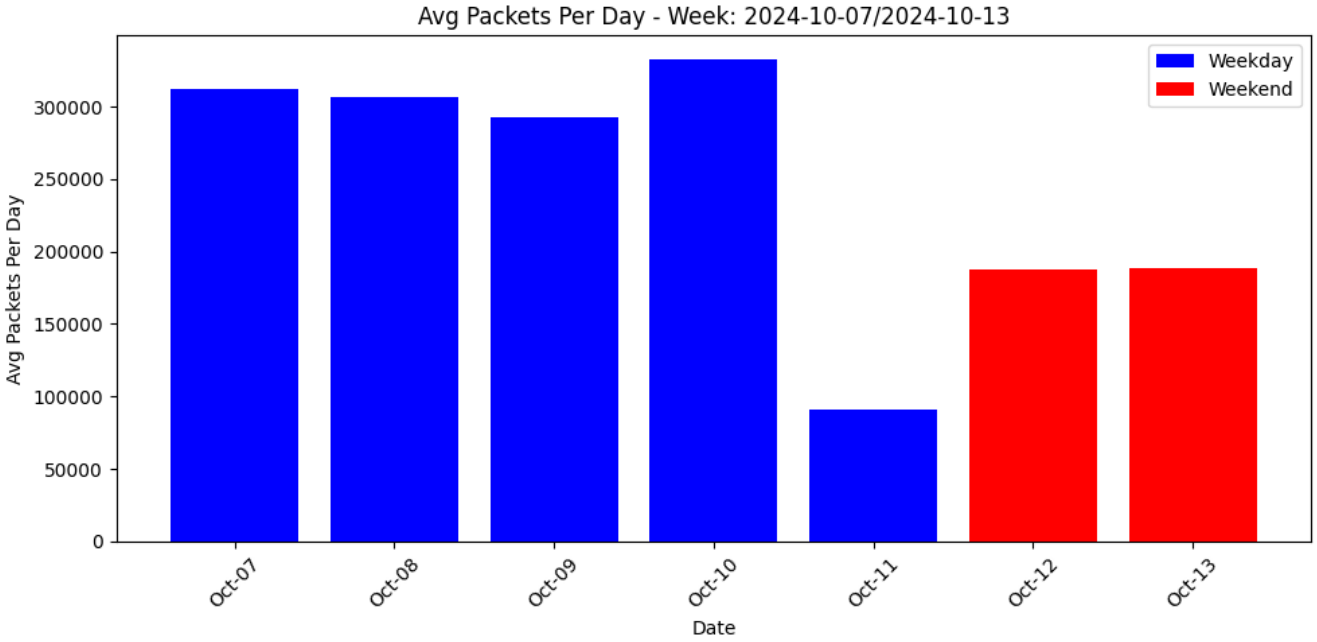


Figure 42: Week sixteen of capturing and setup. Includes IP-based attack traffic.

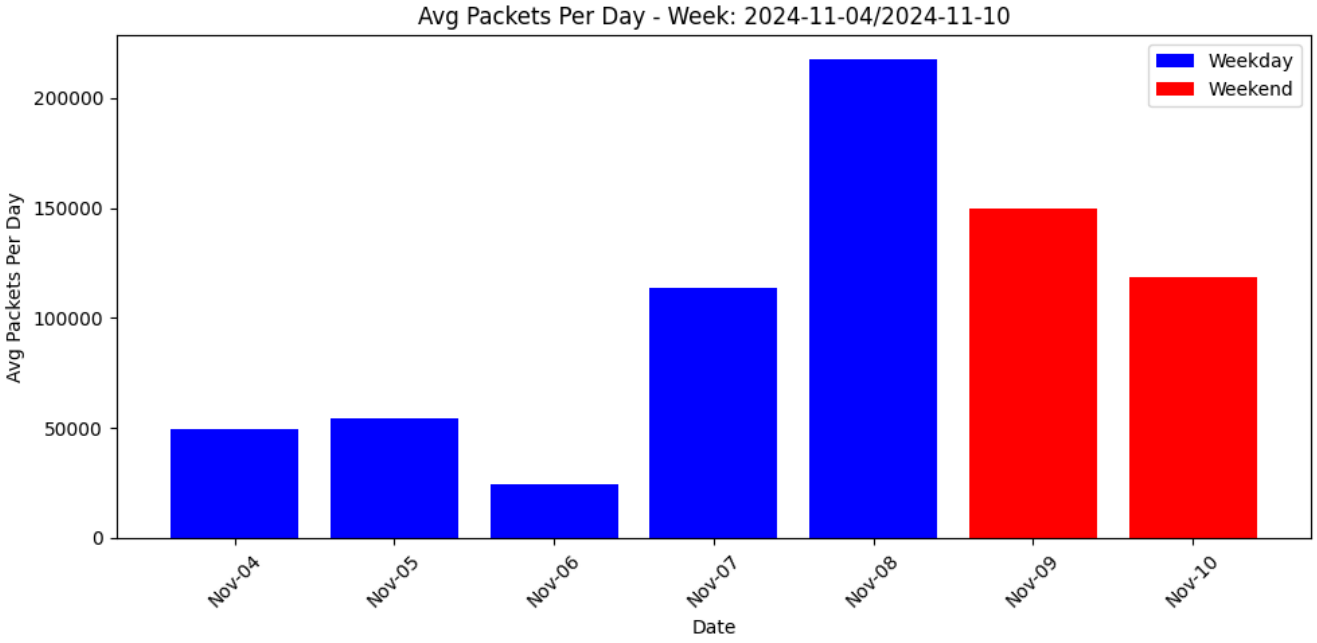


Figure 43: Week twenty of capturing and setup. Includes Z-wave-based attack traffic.

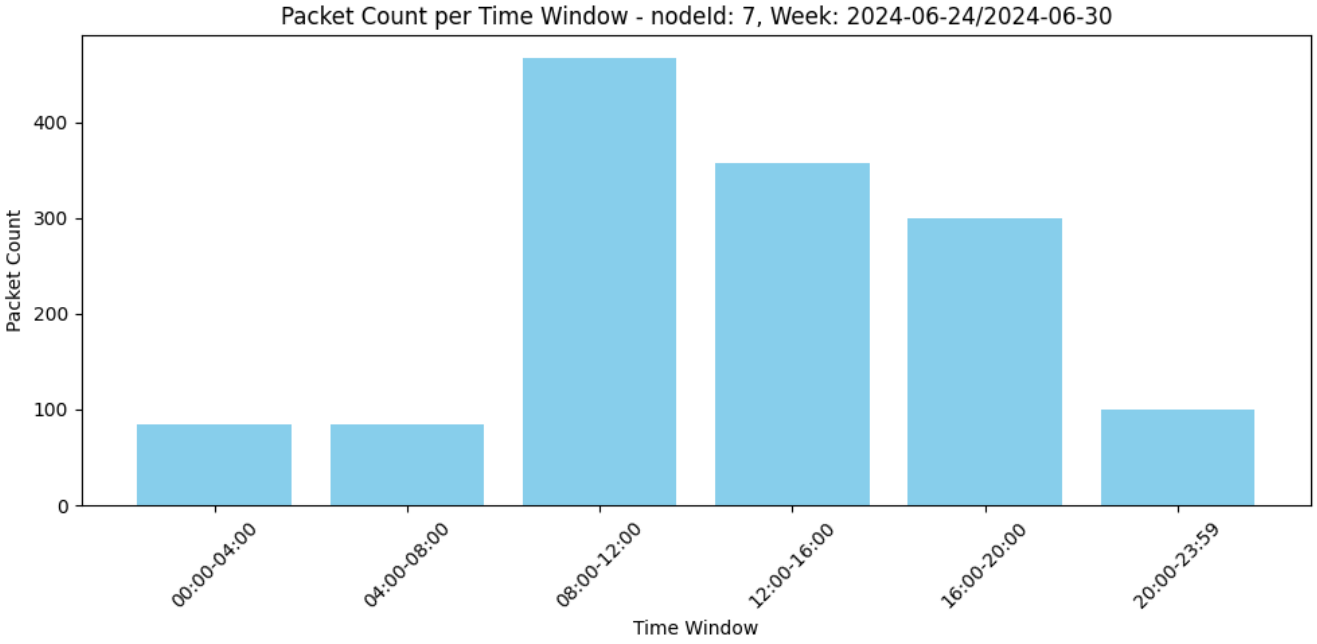


Figure 44: Average packet count of device #7 (tilt sensor on the entrance door) during the week of 2024-06-24/2024-06-30.

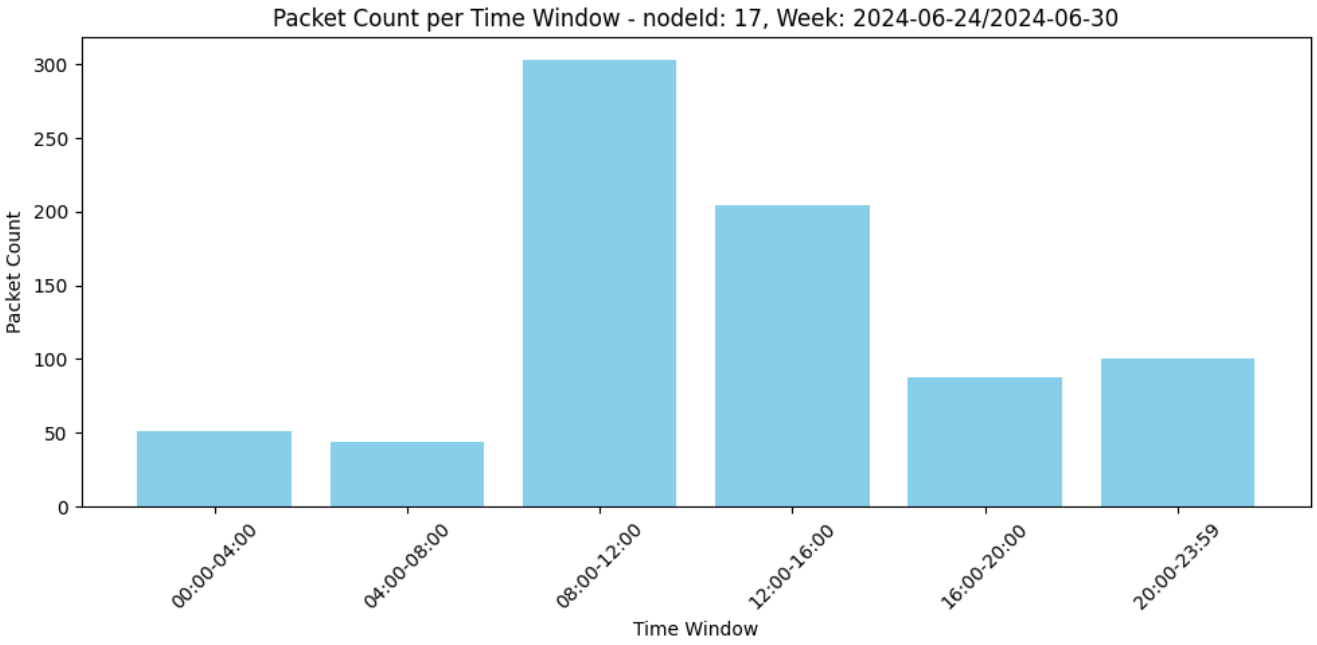


Figure 45: Average packet count of device #17 (motion sensor during the week of 2024-06-24/2024-06-30.

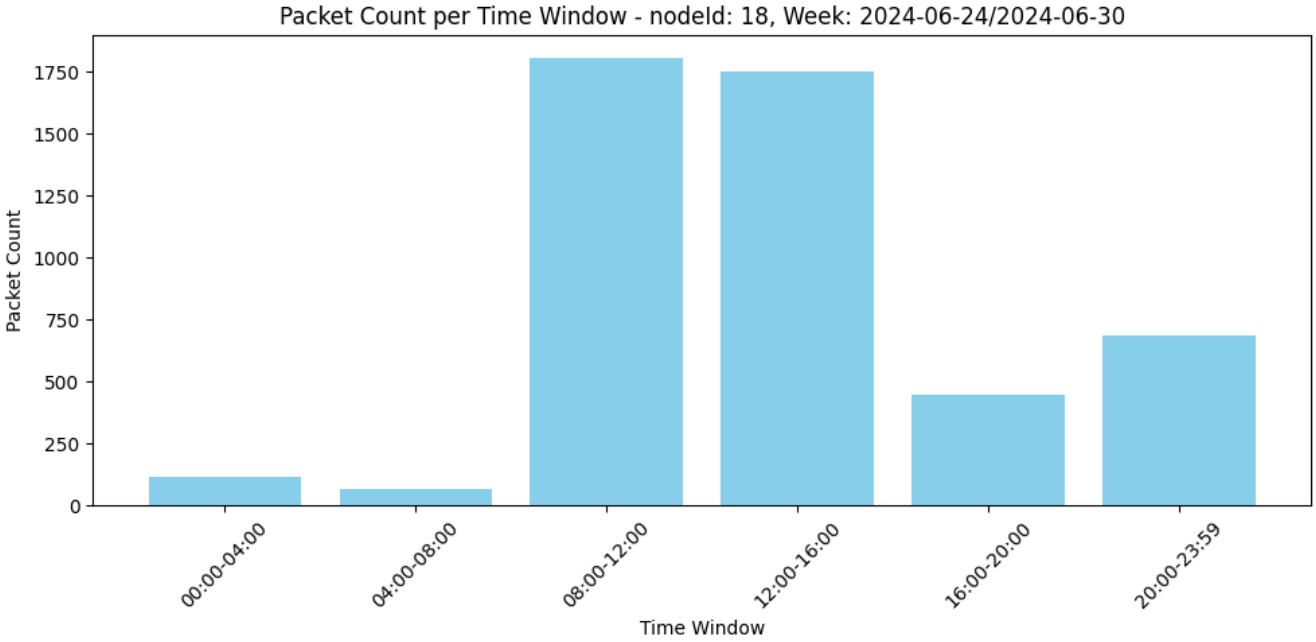


Figure 46: Average packet count of device #18 (motion sensor) during the week of 2024-06-24/2024-06-30.

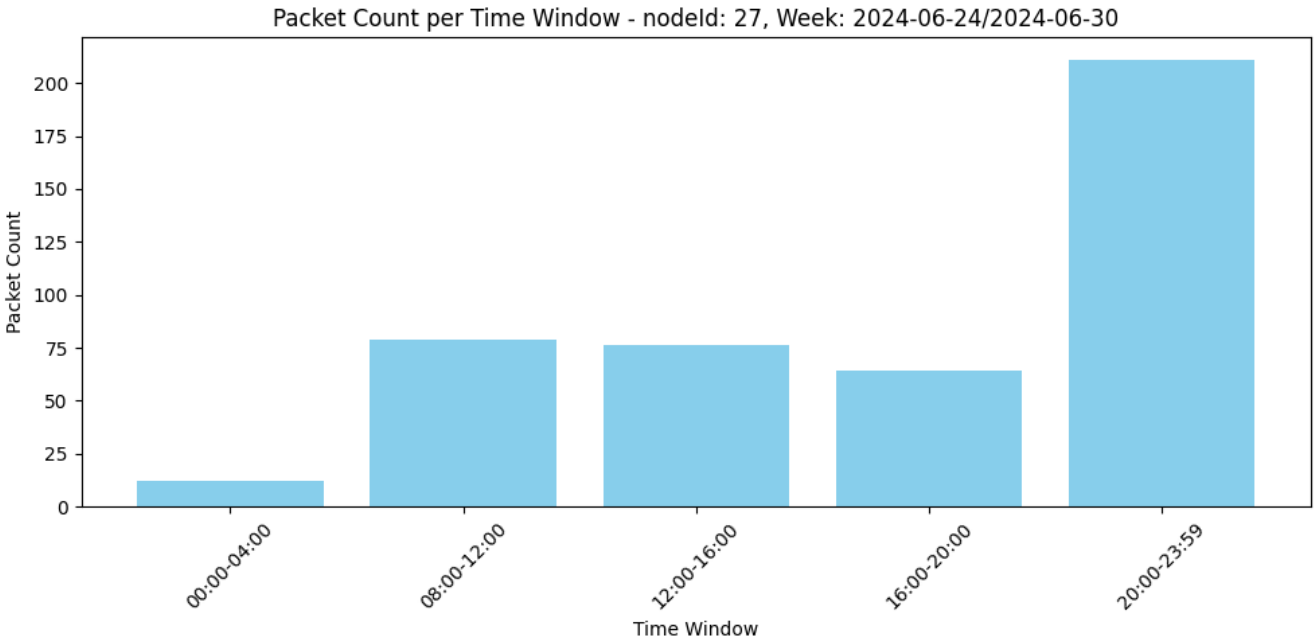


Figure 47: Average packet count of device #27 (plug connected to a lamp) during the week of 2024-06-24/2024-06-30.

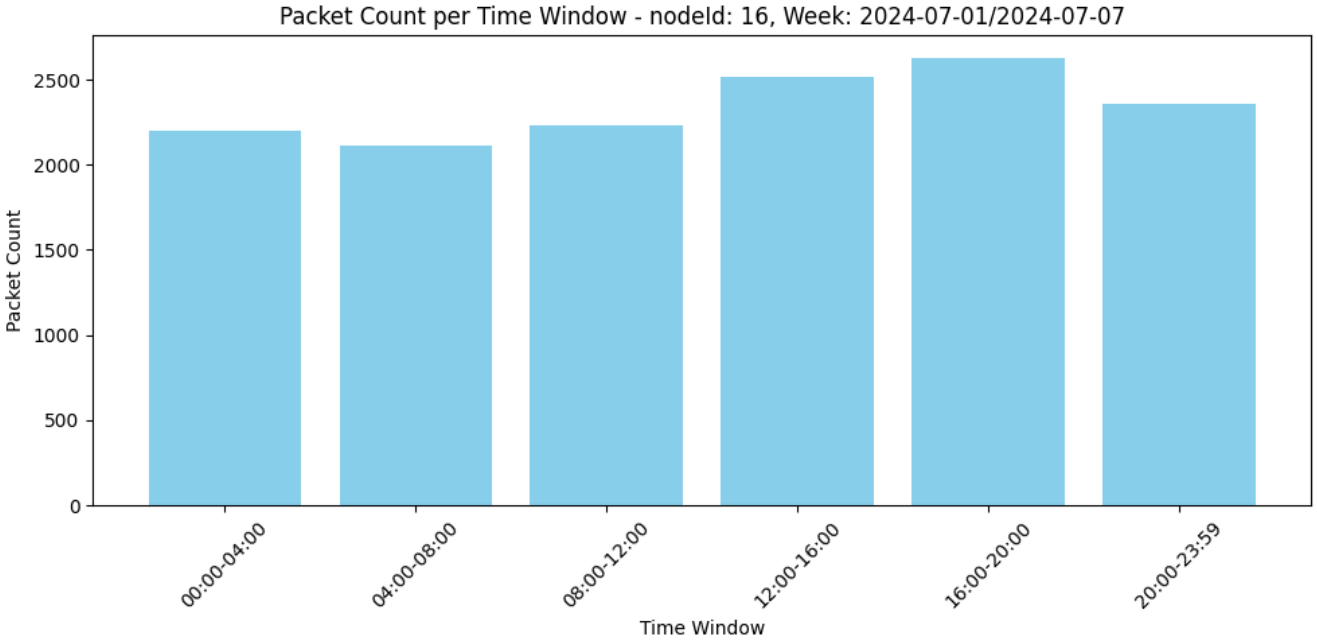


Figure 48: Average packet count of device #16 (plug connected for any use) during the week of 2024-07-01/2024-07-07.

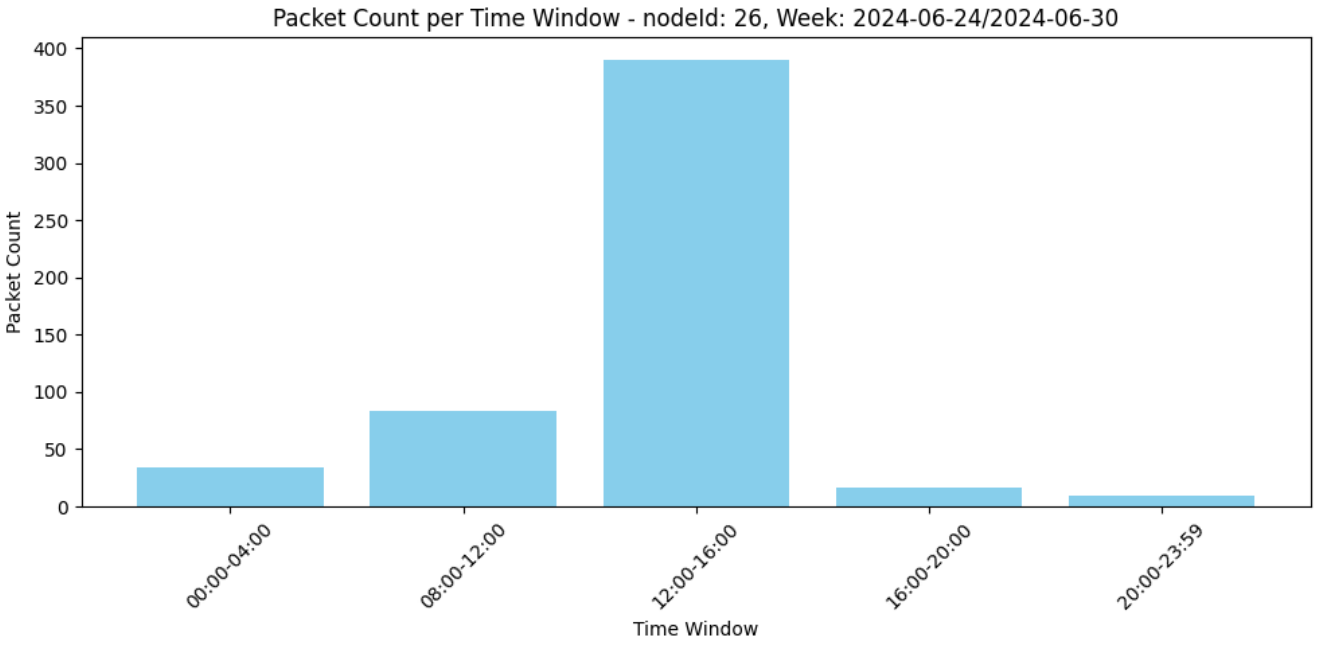


Figure 49: Average packet count of device #26 (door lock) during the week of 2024-06-24/2024-06-30.

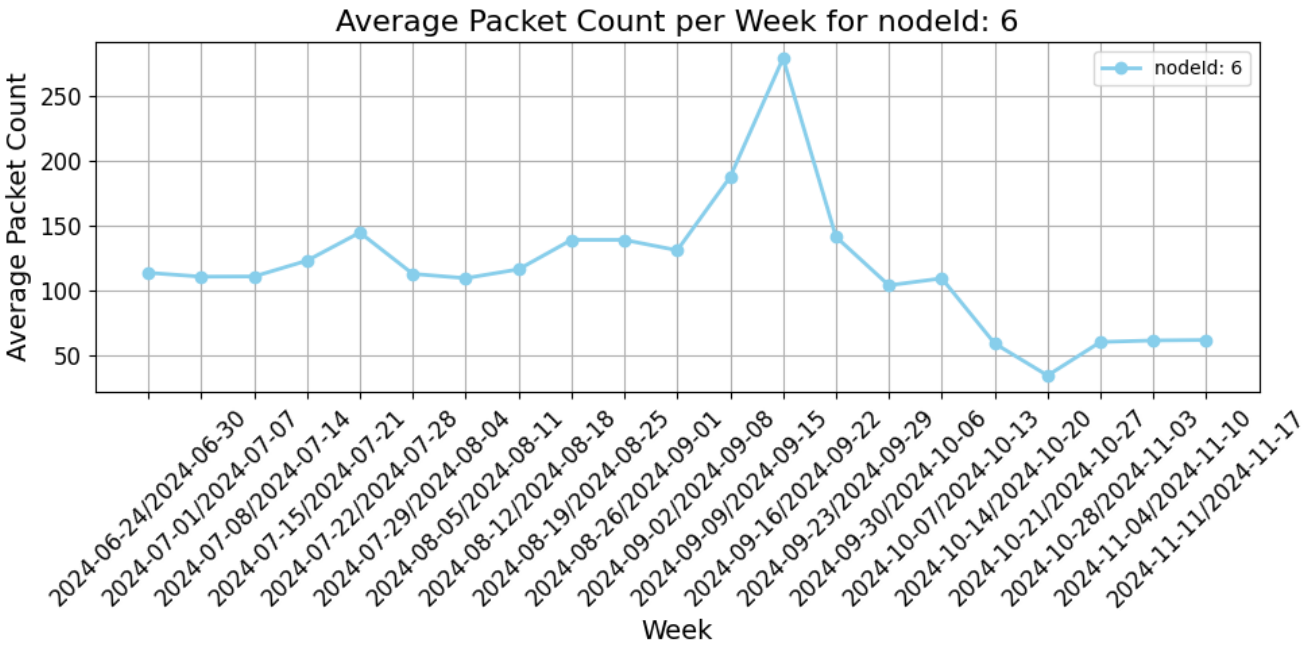


Figure 50: Average packets per week for device #6 (door sensor on the cabinet door)

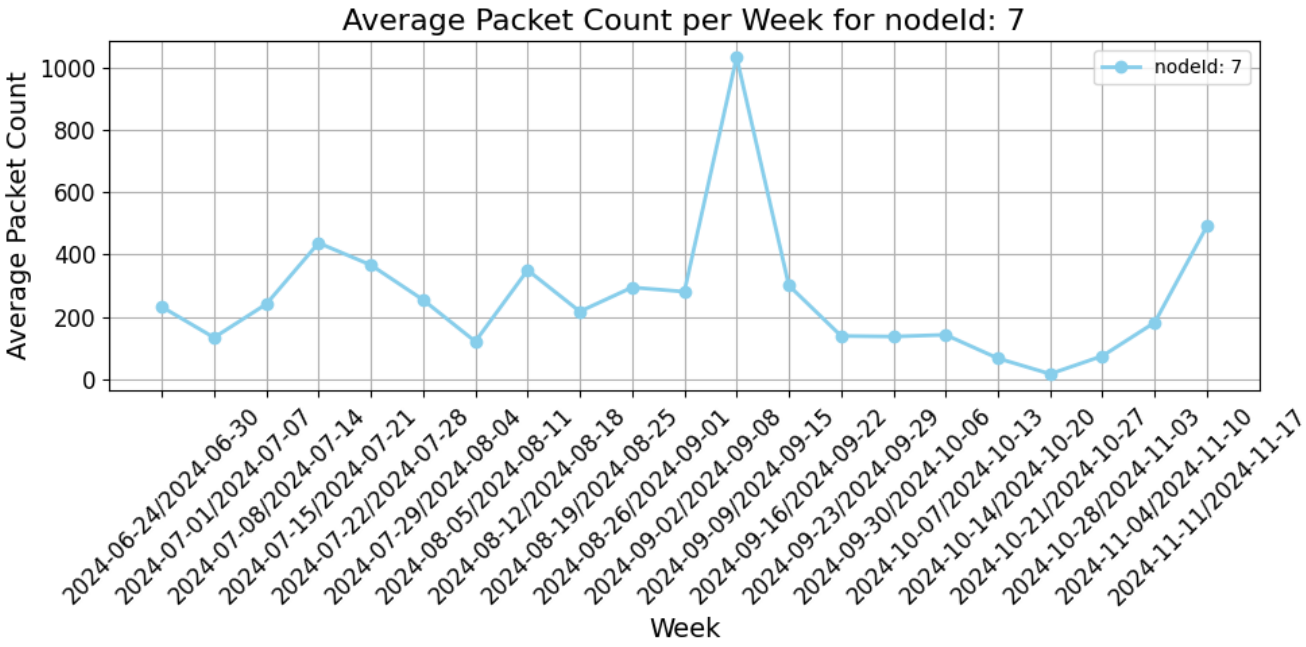


Figure 51: Average packets per week for device #7 (tilt sensor on the entrance door)

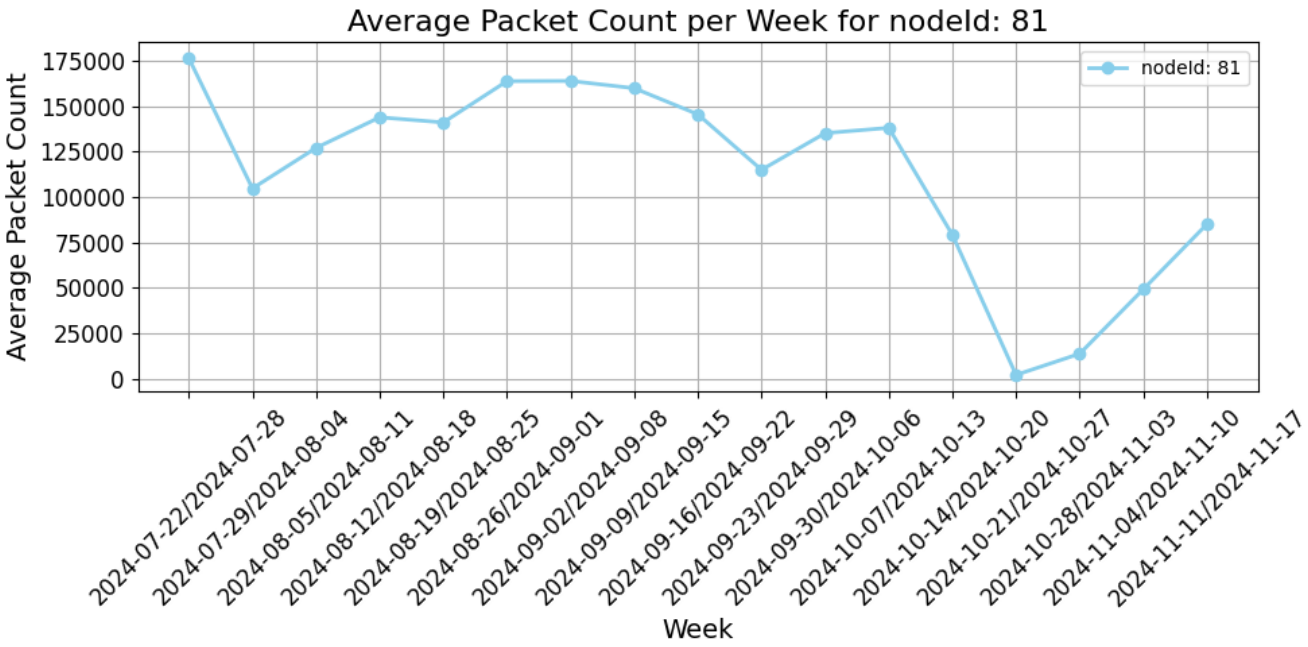


Figure 52: Average packets per week for device #81 (plug connected to lamps)

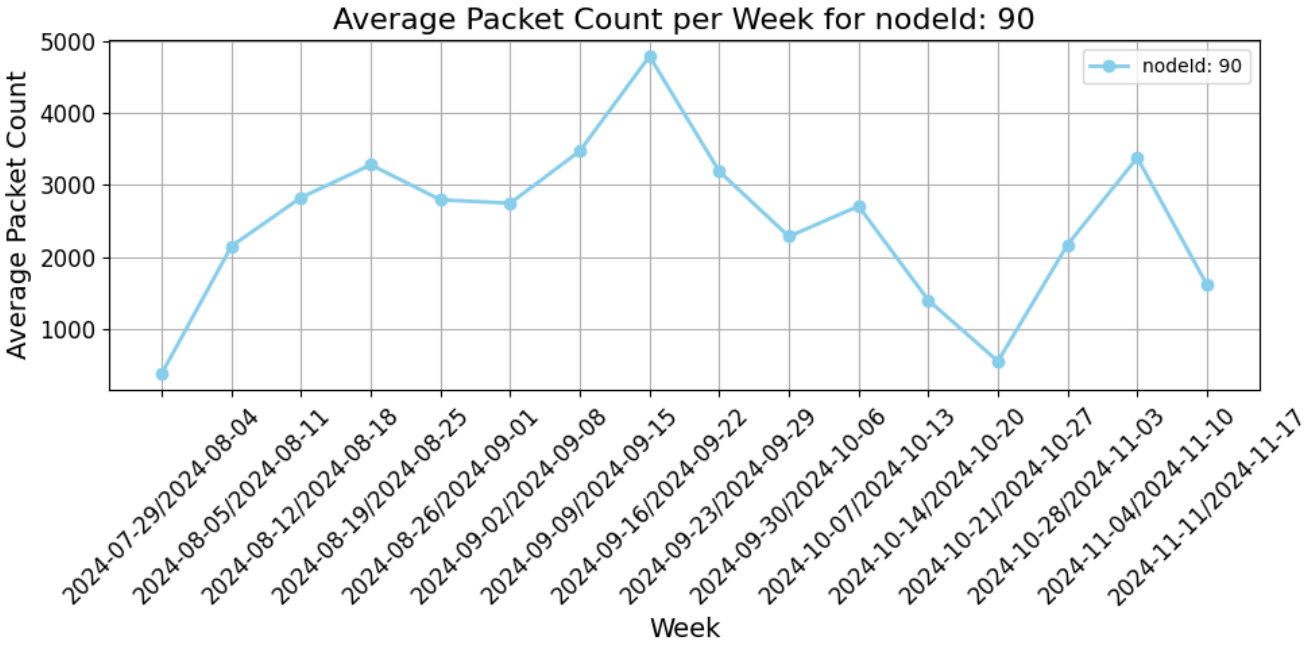


Figure 53: Average packets per week for device #90 (plug connected for any use)

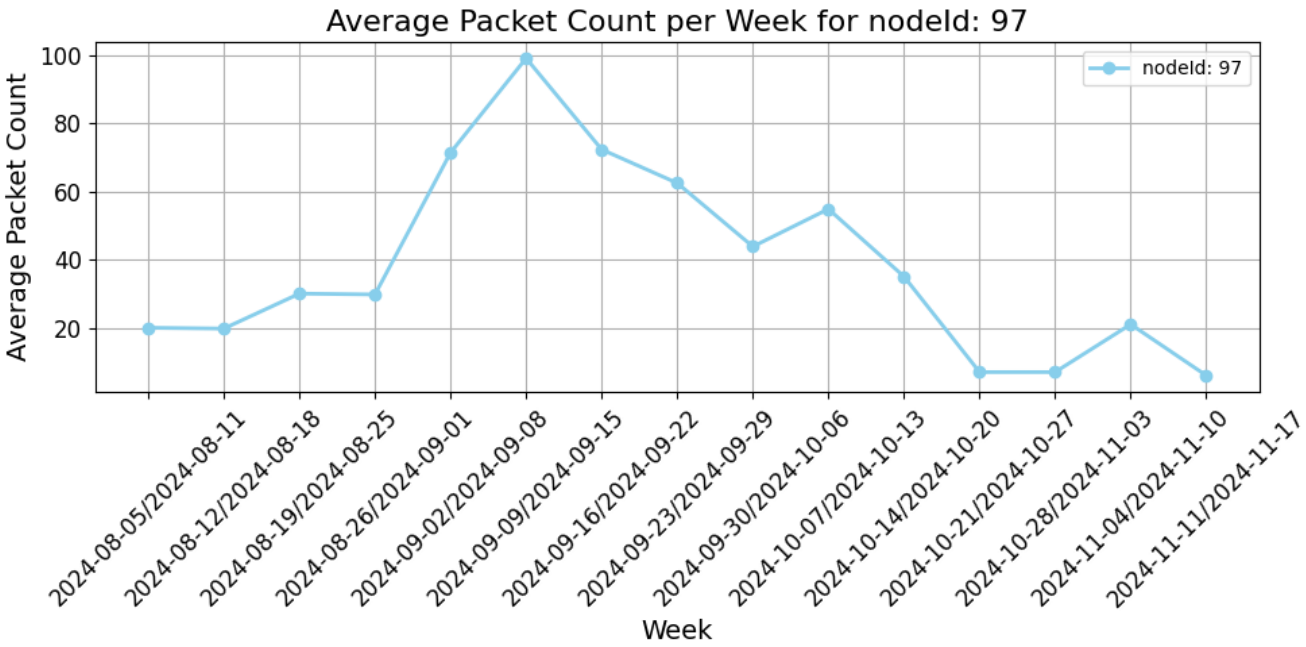


Figure 54: Average packets per week for device #97 (siren)

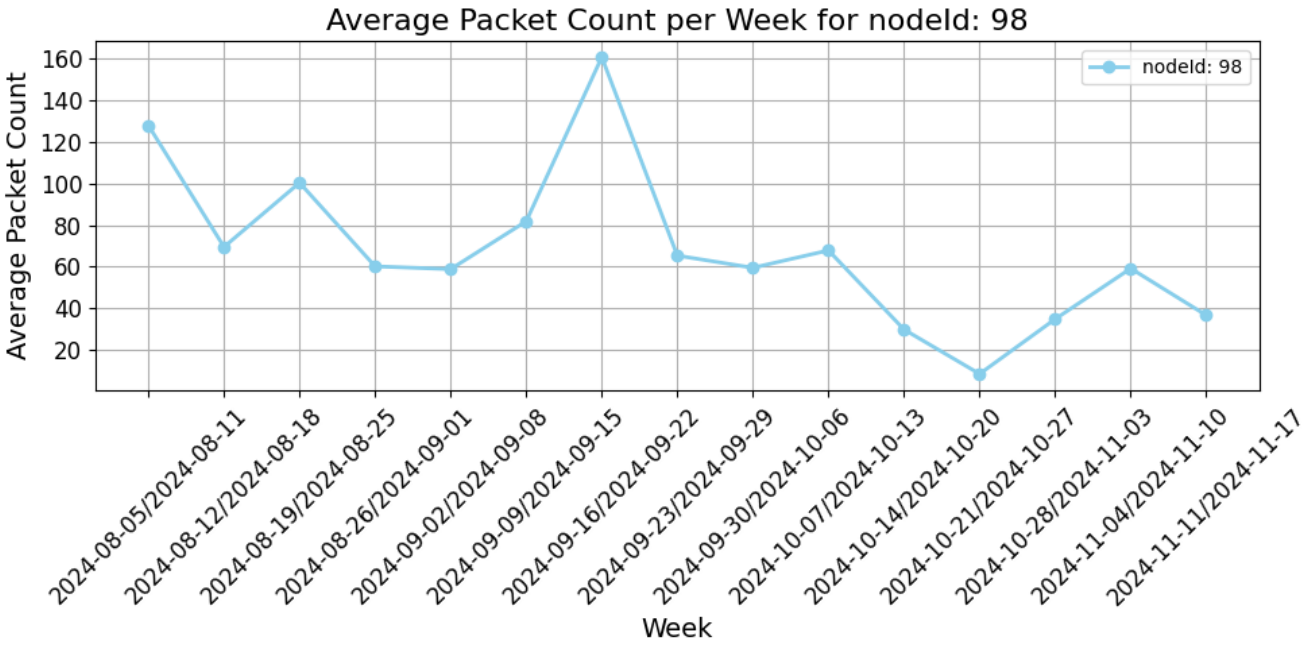


Figure 55: Average packets per week for device #98 (door sensor on the entrance door)

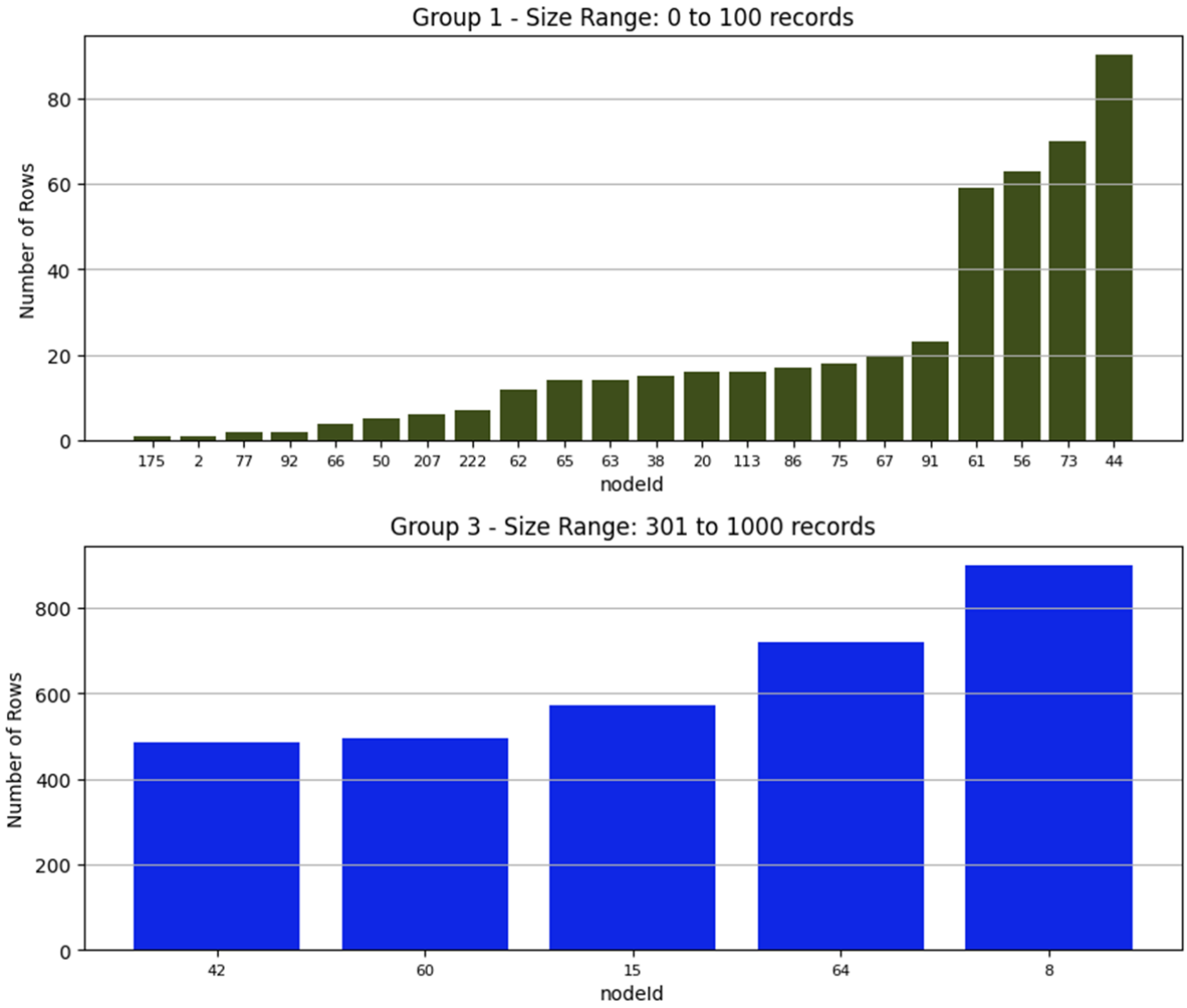


Figure 56: Devices packets count comparison in the first month.

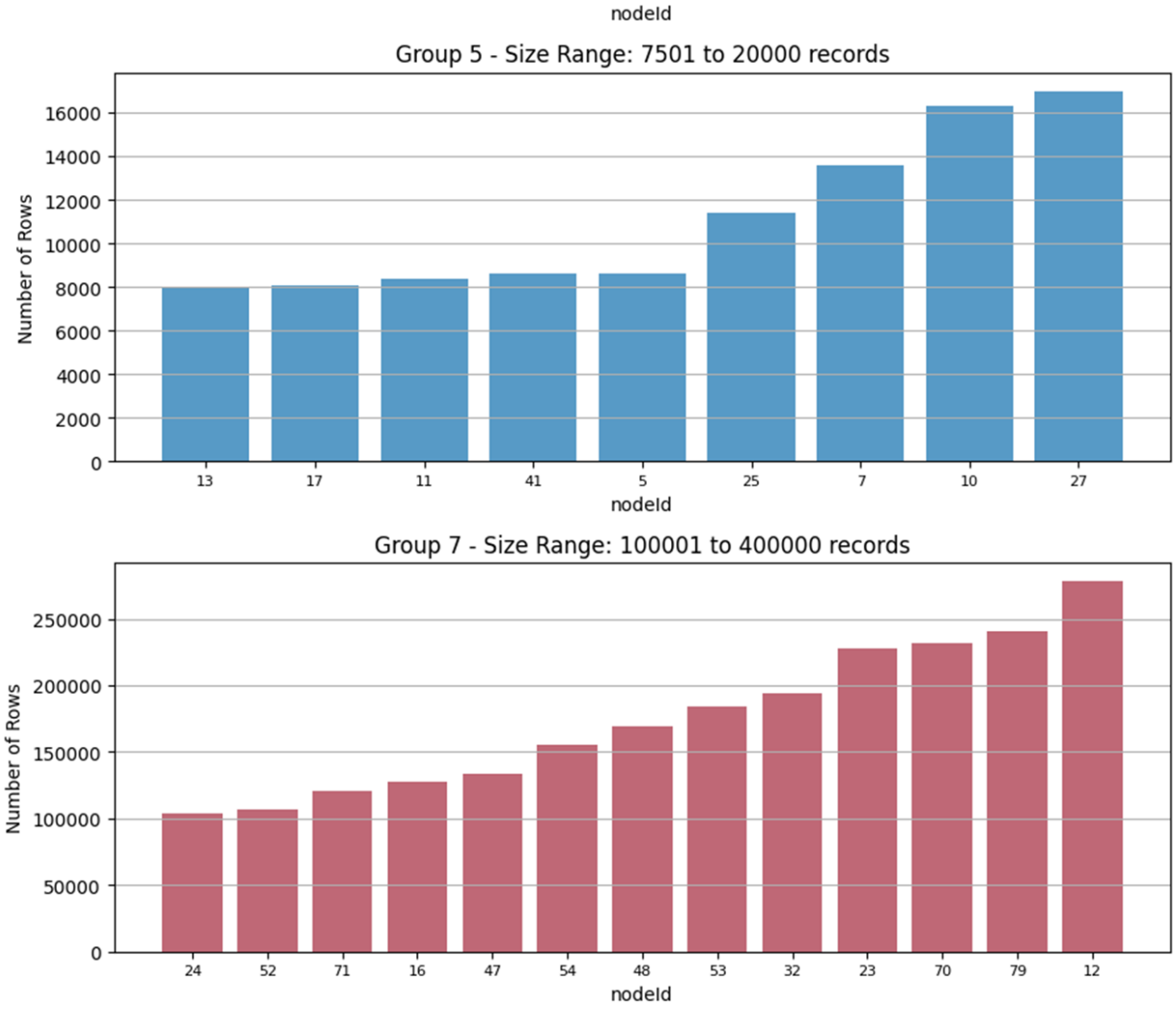


Figure 57: Devices packets count comparison in the first month.

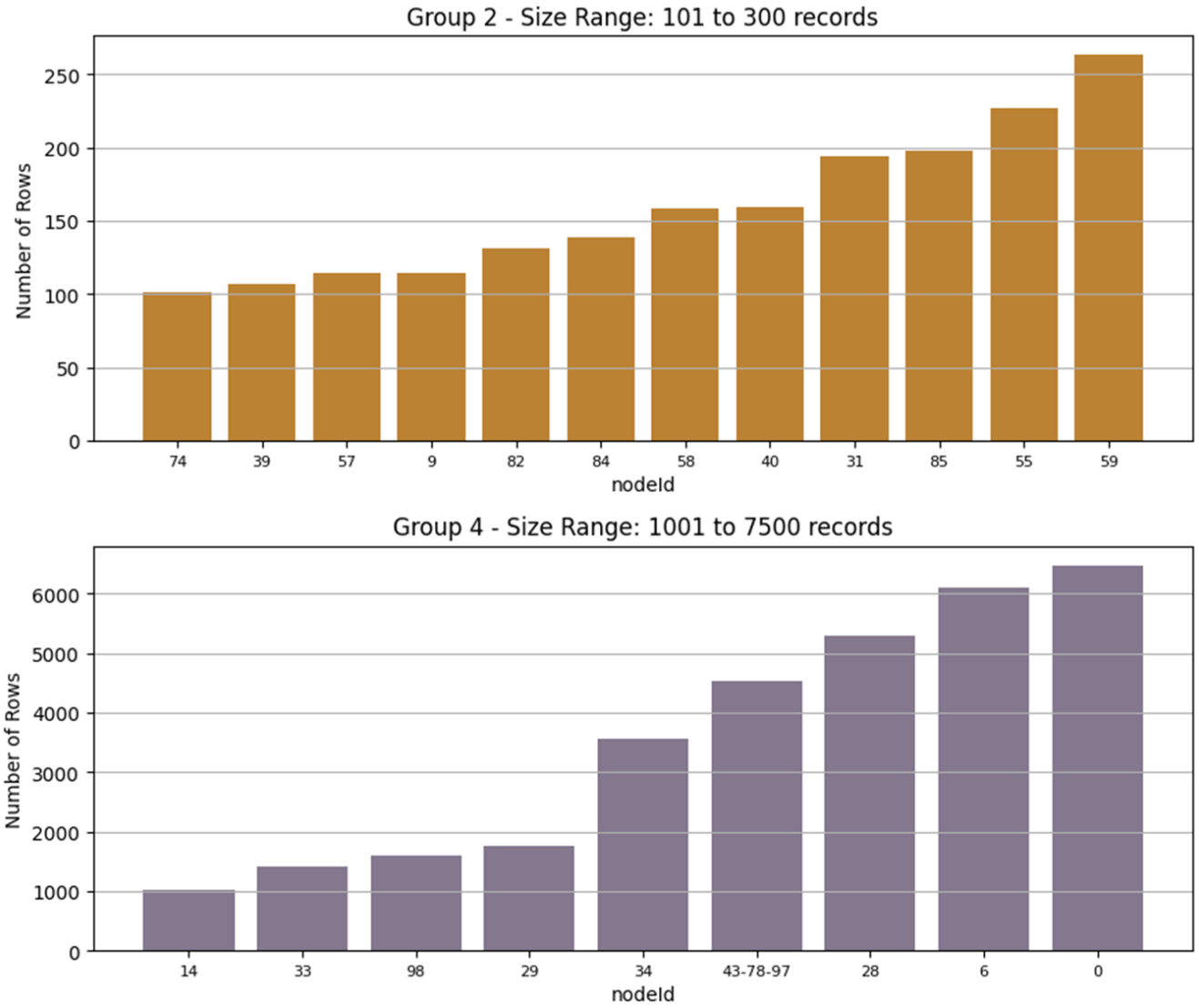


Figure 58: Devices packets count comparison in the first month.

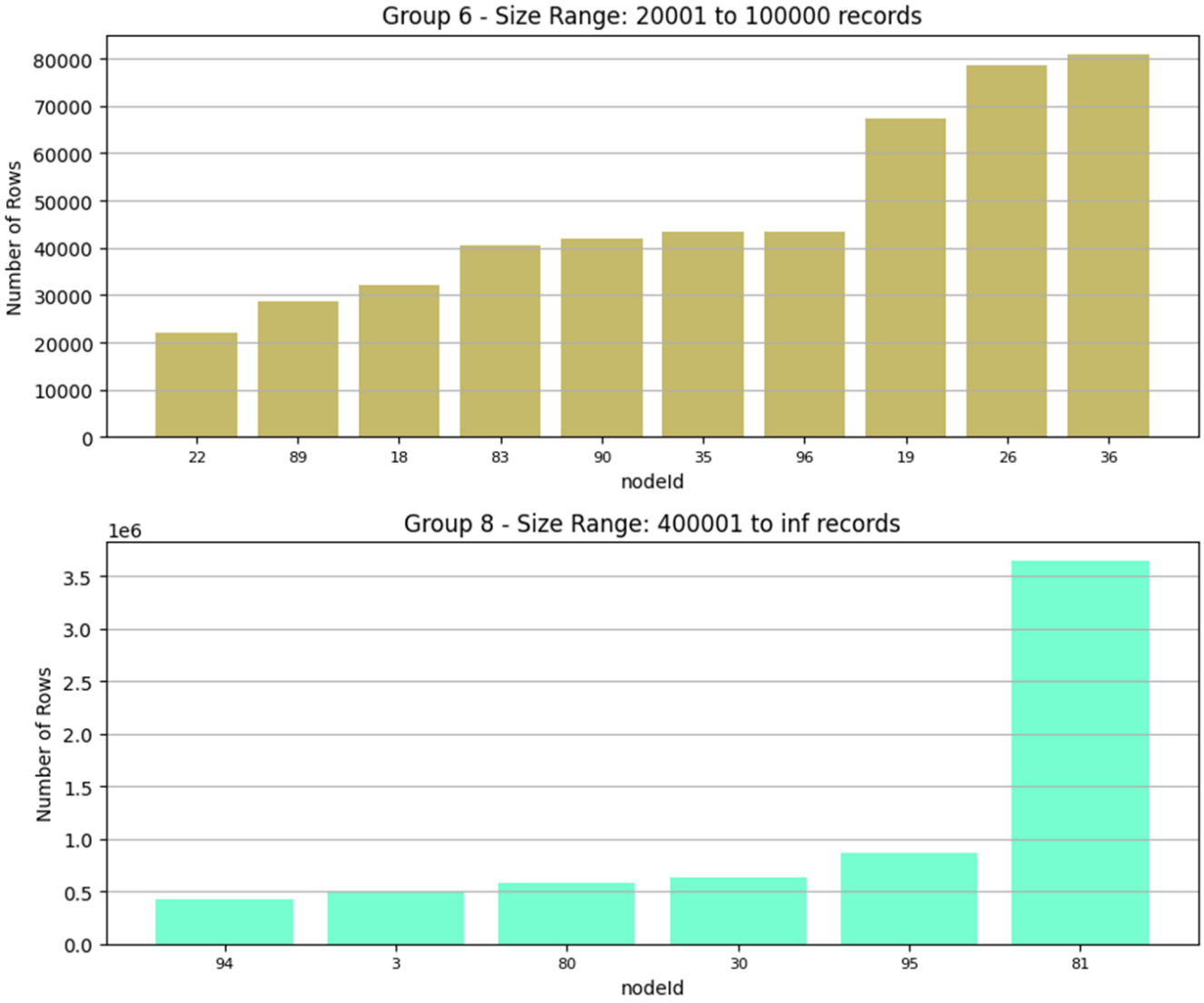


Figure 59: Devices packets count comparison in the first month.

introduces the analyzers employed for different data sources in our testbed, detailing their selection criteria and unique contributions to network traffic analysis.

- ***IP Traffic Analyzer***

  As illustrated in Fig. 36, the network traffic collected in our testbed primarily consists of TCP-based protocols, including HTTP/S, WebSocket, and MQTT. Due to this focus on TCP traffic, our analysis concentrated on identifying and employing analyzers capable of efficiently capturing and interpreting these protocols.

  A thorough evaluation of several publicly available network traffic analyzers, including well-known tools such as CICFlowMeter, NTLFlowLyzer, and NFStream, was conducted. Following a detailed comparison, NTLFlowLyzer emerged as the most suitable choice for this research. This decision was based on its extensive feature set, strong performance metrics, and exceptional ability to process and analyze large volumes of data. Additionally, NTLFlowLyzer is an open-source tool that is actively maintained with regular updates, which ensures adaptability for future research needs or changes in data formats [108, 109].

  NTLFlowLyzer proved to be a robust and scalable solution that aligns seamlessly with our research objectives. Please refer to its official GitHub repository for additional information and detailed documentation on this tool.

- ***Z-wave Traffic Analyzer***

  While traffic analyzers exist for conventional PC networks and widely used protocols such as TCP and HTTP/S, to the best of our knowledge, no analyzers specifically tailored for Z-Wave or other IoT protocols exist. Addressing this gap, we have designed and developed ZwaveNetLyzer, the first traffic analyzer for IoT protocols, focusing on Z-Wave network traffic in the first version.

  ZwaveNetLyzer is designed to parse raw Z-Wave data, aggregate it into meaningful Z-Wave flows, and extract relevant features from them. These features are critical for the subsequent analyses conducted in this work, serving as the foundation for network behavior profiling, anomaly detection, and threat analysis. The extracted features are detailed on GitHub page of the BCCC [110].

  The development of this analyzer is a key contribution of this research, designed entirely by the authors and made publicly available through the Behavior-Centric Cybersecurity Center (BCCC) GitHub repository [110]. This tool fills a critical need in the research community and provides a platform for further IoT traffic analysis and security exploration. Please visit the official GitHub repository for further details and comprehensive information about this tool.

4.2.9 CSV Generation

Once the raw network traffic data was captured from our testbed, it was processed through the designated traffic analyzers, resulting in structured CSV files. These CSV files provide a comprehensive breakdown of IP-based and Z-Wave traffic, allowing for deeper analysis and feature extraction. By structuring the raw captured data into these CSV formats, the resulting CSV files serve as the foundation for further explorations into network behavior, device profiling, anomaly detection, attack identification, etc., offering insights into how conventional and IoT traffic flows within a smart home environment.

The CSV files generated for the IP network traffic focus on TCP-based protocols. For a more granular understanding of traffic patterns, we conducted multiple rounds of analysis on the same pcap files, adjusting the maximum time window for flow creation across several intervals.

The selection of time windows for creating flows in both types of CSV files was guided by a strategic consideration of immediate detection capabilities and long-term analysis. We employed a range of time intervals to capture varying levels of traffic behavior and anomaly patterns. The rationale behind using these multiple windows stems from different attacks manifesting over different periods. Some malicious activities, such as certain DDoS attacks or brute force attempts, can be detected within shorter time windows due to their bursty nature. In contrast, slow or stealthy attacks, such as reconnaissance or data exfiltration, often unfold over a longer duration, requiring extended observation to be effectively identified.

By incorporating smaller time windows, such as 10s and 30s, we aim to capture fast-developing attack scenarios and respond to critical threats as quickly as possible. Such rapid detection is essential in environments where even minor delays could be detrimental. However, smaller windows may result in sparser data or incomplete patterns, which can sometimes hinder accuracy. As a result, larger time windows, like 300s or 3600s, allow for more comprehensive data collection, accommodating both slow-developing attacks and providing richer context for more complex detection methods. This approach balances early detection and a more thorough, data-rich analysis, allowing us to detect a broad range of attack behaviors with varying degrees of immediacy and precision.

Converting raw traffic into CSV format standardizes the data and primes it for subsequent stages of analysis, including feature engineering and machine learning model development. With this structured representation of TCP and Z-Wave traffic, we can thoroughly explore network performance, device communication patterns, and security vulnerabilities. The detailed breakdown of flows ensures a comprehensive dataset well-suited for advanced traffic analysis and security assessment.

4.3 Concluding Remarks

This section addresses three primary contributions of this work. Firstly, it fills a significant gap in available labeled datasets for smart home security research by presenting a new IoT smart home dataset that covers most of the current IoT smart home datasets. The dataset includes real-world data from over 50 devices and features more than 100 diverse attack scenarios, offering a comprehensive foundation for advancing IoT intrusion detection research. Secondly, this section introduces a comprehensive taxonomy of smart home IoT devices, categorizing them based on their behaviors and functionalities. This taxonomy serves as a valuable framework for structuring future IoT security studies and designing targeted detection solutions. Additionally, an in-depth analysis of the Z-Wave protocol highlights its specific vulnerabilities, underscoring the need for protocol-aware security measures. Finally, it conducts a detailed evaluation of existing datasets, identifying their limitations in scale, diversity, and relevance. This analysis underscores the importance of the newly created dataset in addressing these gaps.

5 Experiments & Results

This section presents the experimental results obtained by applying the new dataset to the proposed model. The analysis begins with a detailed examination of feature selection results, providing insights into the most relevant features across different classifiers. Following this, the experimental scenarios employed in evaluating the model are described in depth, covering various attack categories and sub-categories. Finally, the selected features are processed through learning algorithms to perform classification and identification, aiming to assess the effectiveness and accuracy of the proposed approach under multiple conditions.

5.1 Data Pre-processing

In addition to the pre-processing steps outlined in the proposed model, further pre-processing specific to this dataset was conducted to refine class structures and optimize classification efficiency. The dataset contains over 100 distinct classes, many representing subtle variations within broader attack types. Therefore, classes with similar behavioral characteristics were consolidated to reduce complexity. For instance, various strategies under the SYN Flood attack were merged into a single class to provide a more cohesive and manageable representation.

Furthermore, data was organized into primary and secondary categories to improve hierarchical classification within the proposed model's multi-layer structure. The first layer categorizes data based on high-level attack characteristics, grouping similar behaviors into unified categories. This organization was guided by Table 7, which is the attack schedule table, ensuring alignment with the main categories presented in that table. This pre-processing step facilitates targeted classification in the model's initial layer, enhancing the model's interpretability and computational efficiency.

5.2 Feature Selection

This section delves into the feature selection process, employing two prominent methods, Analysis of Variance (ANOVA) and Information Gain. These methods were used to determine the most relevant features for each classifier in the model's hierarchical framework. In the first layer, which encompasses eight primary attack categories with two classifiers per category to handle both IP-based and IoT-based attack classes, a total of 16 classifiers were constructed. For the second layer, additional classifiers were developed based on the more granular sub-categories within each primary category.

Tables 3 and 4 present the selected features for representative classifiers. Given our model's extensive number of classifiers, 16 in the first layer and approximately 80 in the second layer, we have highlighted only a subset of classifiers to provide a focused view. Table 3 illustrates the features chosen for IP-based attack classification, while Table 4 displays those selected for IoT-based classification. These tables emphasize the distinct feature requirements across classifiers, reflecting the unique characteristics of IP and IoT attack profiles and supporting optimizing classifier performance within each domain.

Table 3: Top 40 selected features for IP-based classifier

Class	*Info Gain Algorithm*	*ANOVA Algorithm*
Benign	total/max/min/mean/med/mod_{hdr_bytes}, max/min/mean/med/mod_{fw_hdr_bytes}, fw/bwd_{init_win_bytes}, fw_pkts_rate, rst_flag_cnts, segment_size_min, pkts_rate, fw_syn/syn/_{%_in_total}, duration, fw_syn_%_in_fw_pkts, bw_syn_%_in_bw_pkts, bw_ack_%_in_bwd_pkts, handshake_state, {pkts_IAT}_mean/std/max/total/cov, f_rst_cnts, {fw_pkts_IAT}_mean/std/max/total/med/cov	bw_pkts_cnt, min/mean/med/mod_{hdr_bytes}, min_{fw/bwd_hdr_bytes}, pkts_cnt, mean/med/mod_{bwd_hdr_bytes}, duration, segment_size_min, fw/bw_init_win_bytes, ack/rst/f_ack/f_rst/b_ack_flag_cnts, fw_syn/bw_psh/syn/rst_%_in_total, fw_rst_%_in_fw_pkts, bw_psh/syn_%_in_bw_pkts, {bw_pkts_IAT}_mean/max/min/total/med/mod, mean/med_{fw/pkts_DT},
Information Gathering	total/max/min/mean/med/cov/mod_{hdr_bytes}, total/min/mean/med/mod_{fw_hdr_bytes}, duraiton fw_segment_size_mean/max/min/med/mod, {segment_size}_mean/max/min/std/var/med/mod, fw_init_win_bytes, down_up_rate, rst_cnts, fw_syn/syn/ack_{%_in_total}, bw_ack_%_in_total, fw_syn_%_in_fw_pkts, bw_syn_%_in_bw_pkts, handshake_state, var/std/cov_{pkts_DL}	segment_size_cov, rst/f_rst/bwd_fin_flag_cnts, fin/fw_syn/syn/ack/bw_fin/psh/rst_%_in_total, fw_syn/ack_%_in_fw_pkts, bw_fin/psh_%_in_bw_pkts, {bw/pkts_IAT}_mean/max/min/total/med/mod, mean/std/max_fw_{pkts_DT}, std/mode_bwd_{pkts_DL}, mean/var/std_fw_{hdr_bytes_DL}
Botnet	duration, total/max/min/mean_{hdr_bytes}, total/min/mean_{fw_hdr_bytes}, pkts_rate, fw/bw_segment_size_mean/max, segment_size_mean/max/min/std/var, rst_flag_cnts, fw_syn/syn/rst/_{flag_%_in_total}, fw_syn_%_in_fw_pkts, bw_ack_%_in_bw_pkts, {pkts_IAT}_mean/max/min/total/med/mod, handshake_state, fw_pkts_rate, cov_pkts_DL {fw_pkts_IAT}_mean/max/min/total/med/mod,	max_{hdr_bytes}, fin/f_rst/bw_fin_flag_cnts, fin/fw_fin/bw_fin/syn/rst_%_in_total, fw_fin/syn/rst_%_in_fw_pkts, {fw/bw/pkts_IAT}_mean/max/min/ total/med/mod, handshake_duration, mean/max_fw_{pkts_DT}, mean/med_fw_{pkts_DL_time}
ACK Flood	max/mean_{hdr_bytes}, med/mod_{fw_hdr_bytes}, max/min/mean/med/mod_{bw_hdr_bytes}, fw_init_win_bytes, fw/bw_{pkts_rate}, syn/fw_syn/fw_ack/bw_rst_{flag_%_in_total}, fw_syn/fw_ack/_%_in_fw_pkts, duration bw_ack/bw_rst_%_in_bw_pkts, rst_flag_cnts, {fw/bwd_pkts_IAT}_mean/max/min/total/med/mod, handshake_state, mean/mode/med_{pkts_DT}	mean/std/med/cov/mod_{fw/hdr_bytes}, fw/bw_init_win_bytes, {fw/bw}rst_flag_cnts, syn/fw_syn/ack/bwd_ack/rst_%_in_total, fw_syn/ack_%_in_fw_pkts, handshake_state, bw_ack/rst_%_in_bw_pkts, {bw_pkts_IAT}_mean/max/min/med/mod, min/mean/mode/med_{fw_pkts_DT}, min/std_{hdr_bytes_DL}

TOR DDoS	otal/max/min/mean/med/mod_{hdr/fwd_hdr_bytes}, mod_bw_hdr_bytes, segment_size_mean/min/med, fw/bw_init_win_bytes, pkts_rate, fw_pkts_rate, down_up_rate, syn/rst/fw_syn/fw_rst_{flag_cnts}, syn/fw_syn_{flag_%_in_total}, subflow_fw_pkts, fw_syn_%_in_fw_pkts, bw_ack_%_in_bw_pkts, {packet_IAT}_max/total, handshake_state, {fw_pkts_IAT}_mean/std/max/total/var, duration	fw_payload_bytes_max, max/min/mean/med/mod_{fw/hdr_bytes}, fw_segment_size_max, segment_size_min/med, fw_init_win_bytes, syn/fw_syn_flag_cnts, rst/fw_syn/bw_ack_%_in_total, fw_rst_%_in_fw_pkts, {fw/pkts_IAT}_std/cov, {bw_pkts_IAT}_mean/max/min/total/med/mod, max/mean/mode/std/med_{fw/pkts_DT},
TCP Slowloris	{fwd/payload}_bytes_max/std {fw/segment}_size_mean/max/std/med/cov, {bw_pkts_IAT}_mean/max/min/total/med/mod, min/mean/mode/std/med/cov_{fwd/pkts_DL}, min/mean/std/med_{fw/payload_bytes_DL},	max/min/mean/med/mod_{fw/hdr_bytes}, fw_segment_size_cov, fw/bw_init_win_bytes, fw_packets_rate, rst/f_rst_flag_cnts, bw_syn_%_in_total, fw_ack/rst_%_in_fw_pkts, bw_syn/ack_%_in_bw_pkts, handshake_state, {bw_pkts_IAT}_mean/max/min/total/med/mod, var/std_{hdr_bytes_DL}, mean/mode/med_{fw_payload_bytes_DL}
Websocket BruteForce	duration, total/max/min/mean_{fw/hdr_bytes}, {segment_size}_mean/max/min/std/var/cov, fw/bw_{init_win_bytes}, rst_flag_cnts, syn/fw_syn_{flag_%_in_total}, fw_syn_%_in_fw_pkts, bw_ack_%_in_bw_pkts, {fw/pkts_IAT}_mean/max/total/med/mod, handshake_duration, handshake_state, min/max/var/std_{pkts_DL}	fw_std/cov/var_{hdr_bytes}, fin/fw_fin/bw_fin_flag_cnts, bw_fin/fw_fin/fin/syn/rst_%_in_total, fw_fin/syn/rst_%_in_fw_pkts, bw_fin_%_in_bw_pkts, delta_start, {bw/pkts_IAT}_mean/max/min/total/med/mod, max/mean/var/std/med_{bw_pkts_DT} mode/var/std/mode_{fw_hdr_bytes_DL}
Get Flood	total/max/min/mean/med/mod_{fw/bw/hdr_bytes}, fw_{segment_size_min}, fw/bw_init_win_bytes, rst/fw_rst_flag_cnts, handshake_state, duration syn/fw_syn_{flag_%_in_total}, fw_syn_%_in_fw_pkts, bw_ack_%_in_bw_pkts, {fw/pkts_IAT}_mean/max/min/total/med/mod,	max/min/mean/std/mod_{bw/fw/hdr_bytes}, segment_size_min, std_{hdr_bytes_DL}, fw/bw_init_win_bytes, down_up_rate, rst/f_rst_flag_cnts, bw_syn/fw_syn/syn/ack/rst_%_in_total, fw_syn/rst_%_in_fw_pkts, bw_syn/ack_%_in_bw_pkts, {bw_pkts_IAT}_mean/max/min/total/med
Cookie based DDoS	total/max/min/mean/med/mod_{fw/hdr_bytes}, segment_size_mean/min/med/mod, handshake_state, fw/bw_{init_win_bytes}, fw/_{pkts_rate}, duration rst/f_rst_flag_cnts, syn/fw_syn_{flag_%_in_total}, fw_syn_%_in_fw_pkts, bw_ack_%_in_bw_pkts, {fw/pkts_IAT}_mean/max/total, subflow_fw_pkts	max_fw_{pkts_DT}, syn/rst/fw_syn_flag_cnts, max/min/mean/med/mod_{fw/hdr_bytes}, segment_size_min/med/mode, syn/rst/fw_syn_%_in_total, fw_init_win_bytes, {bw_pkts_IAT}_mean/max/min/med/mod, mean/mode/std/med_{fw/pkts_DT}

Table 4: Top 40 selected features for IoT-based classifier

Class	*Info Gain Algorithm*	*ANOVA Algorithm*
Benign	data_bytes_rate, entropy_hex_data, most/least_common_channel, data_field_entropy, fw/_hex_data_pattern_len_var, fw_most/least_common_channel, min_speed, total/mean/var/std/skew/coeff_data_field_size, fw_variance/std_pkts_DT, fw_max/std_pkts_DD fw/bw{max}/min/mean/med/std_scheme_state, max/mean/var/std/skew/coeff_var_pkts_DD,	bw_min_speed, avg/med/max_rssi, mode_speed, bw_avg/mode/stddev/max_rssi, dst_id, fw/bw_{hex_data_pattern_len_var}, bw_std/var/std/fw_var_{data_field_size}, fw_max/coeff_var/max/std_{num_hops}, bw_var/std_scheme_state, bw_rssi_range/var, bw_max/var/std/coeff/max/std_{tries}, fw_max/bw_max/max/std_pkts_DD
Z-wave Attacks (general category)	min_speed, fw/hex_data_pattern_len_var, fw/data_field_entropy, fw/entropy_hex_data, fw_most/least_common_channel, total/max/mean/mode/var/std_{data_field_size}, fw_total/mean_data_field_size, coeff_num_hops, max/min/mean/med_{bw/fw/scheme_state}, mean/variance/std/skew/coeff_{pkts_DD}	avg/mode/bw_avg/med/mode/min/max_{speed}, avg/med/max/min_{bw/rssi}, bw_rssi_range/var, bw_hex_data_pattern_len_var, dst_id var_{bw/data_field_size}, max/std_{fw/num_hops}, mean/std_{bw/delivery_time}, max/std_{bw/pkts_DD} bw_min/var/std_{scheme_state}, max/mean/bw_var/std_{bw/tries}
Layer7 D/DoS (HTTP-based)	bwd_avg/med/mode/min/max_{bw/speed}, most/least_{fw/common_channel}, max/min/mode/med_{fw/bw/data_field_size}, max/mode_num_hops, speed_range, max/min/std/median_{fw/bw/scheme_state}, min_{bw_tries}, max_{fw/pkts_DD}	src_id, duration, bw_std/var_speed, speed_range, min_{bw/rssi}, rssi_range, hex_data_pattern_len_var, bw_coeff_num_hops, incr_data_change, fw_hex_data_pattern_len_var, bw_hex_data_pattern_len_var, max/var/std/coeff_var_data_field_size, bw_std_data_field_size, fw/skew_pkts_DT, fw_incr_data_change, fw_var/std_{num_hops}, coeff_delivery_time/bw_coeff_delivery_time, bw_std_scheme_state, coeff_var/bw_coeff_tries, bw/max/fw_max/mean/var/std/med_pkts_DD,

Layer4 D/DoS (TCP-based)	bwd_avg/med/mode/min/max_{bw/speed}, most/least_{fw/common_channel}, max/min/mode/med_{fw/bw/data_field_size}, max/mode_num_hops, speed_range, max/min/std/median_{fw/bw/scheme_state}, min_{bw_tries}, max_{fw/pkts_DD}	bw/entropy_of_hex_data, {bw/avg_channel}_usage/stability, fw/bw{data_field_entropy}, coeff_tries, fw/bw{incremental_data_change}, bw_{unique_hex_patterns}_cnt, bw/fw_{unique_data_entries}, speed_range, max/std/var_{bw/speed}, bw/_pkts_count, total/max_data_field_size, skew_pkts_DT, bw_coeff_num_hops, fw/bw_{max_pkts_DD}, coeff_{bw/delivery_time}, rssi_range, bw_std_{scheme_state}, min_{bw/rssi}
Jamming	pkts_cnt, avg/std/med/mode/var_{bw/speed}, avg/med/std/min/var_{rssi}, rssi_range, fw/hex_data_pattern_length_variability, channel_stability, channel_stability, max/mean/var/std_{fw/bw/data_field_size}, max/mean/var/coeff_{delivery_time}, mean/var/std/coeff_{scheme_state}, mean/std/skew/coeff_{fw/pkts_DD}	fw/pkts_cnt, skew_{bw/speed}, std_{rssi}, fw/hex_data_pattern_length_variability, avg_channel_usage, incremental_data_change, total/max/var/std/coeff_{fw/data_field_size}, skew/coeff_{fw/pkts_DT}, mean/med_{number_of_hops}, channel_stability, max/std/skew_{pkts_DD}, min/mean/med/coeff/std_{bw/scheme_state}
Fuzzing with correct CRC	avg/std/max/var/med/min_{bw/speed}, std/var_{rssi}, fw_channel_stability, fw_mean_pkts_DT, rssi_range, speed_range mean/std/skew/coeff/med_{number_of_hops}, max_{delivery_time}, mean/var/_{scheme_state}, mean/var/std/skew/coeff_{fw/packets_DD},	dst_id, pkts_cnt, channel_stability, avg/med/mode/std/max_{speed} var_rssi, avg_channel_usage, fw_hex_data_pattern_length_variability, time_series_analysis_packet_intervals, skew/coeff_{pkts_DD}, speed_range, bw_cross_correlation_speed_rssi, total_data_field_size, mod/med/skew/std_{fw/bw/data_field_size}, mean/med_{number_of_hops}, med/var/std/_{scheme_state},
Replay	avg/med/std/max/var_{bw/speed}, rssi_range, fw/entropy_of_hex_data, fw/data_field_entropy, var/std/mean_{number_of_hops}, speed_range, med/min_{bw/delivery_time}, mean/var/std/coeff/med_{scheme_state}, coeff/med_{tries}, std_rssi, bw_mode_rssi, avg/med/std/max/var_{bw/speed}, mean/var/std/skew/coeff_{fw/pkts_DD},	avg/med/mod/std/max/var_{bw/speed} std/max/var_{bw/rssi}, bw/rssi_range, bw_cross_correlation_speed_rssi, speed_range, dst_id max/mean/var/std/med_{bw/scheme_state}, mean/var/std/median_{bw/tries}, bw_std_packets_DD

Table 5: Performance results.

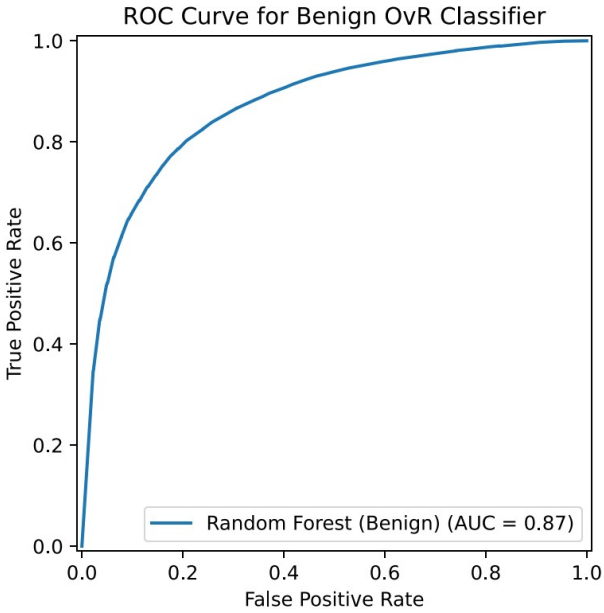
Class	*Pre.*	*Rec.*	*F1.*	*Class*	*Pre.*	*Rec.*	*F1.*	*Class*	*Pre.*	*Rec.*	*F1.*
Some OvR Classifiers Results (Individually)											
Benign IP	0.92	0.96	0.94	***Botnet***	1.00	1.00	1.00	***BYPASS***	1.00	1.00	1.00
WS BruteFroce	1.00	1.00	1.00	***Info. Gathering***	1.00	1.00	1.00	***SYN-Flood***	1.00	1.00	1.00
Jamming	0.72	0.71	0.71	***Fuzzing***	0.81	0.82	0.82	***TCP-Flood***	0.95	0.88	0.91
ACK-Flood	1.00	1.00	1.00	***COOKIE***	0.91	0.89	0.90	***GET-Flood***	0.99	0.99	0.99
SYN-PSH-URG	1.00	1.00	1.00	***HEAD Flood***	0.97	0.93	0.95	***ZwaveGeneral***	0.95	0.96	0.95
TOR	0.91	0.87	0.89	***Zwave-Replay***	0.85	0.69	0.73	***Benign Zwave***	0.69	0.57	0.59
First Layer Results											
Benign	0.99	0.99	0.99	***Info. Gathering***	1.00	1.00	1.00	***Layer 4 Attack***	0.97	0.97	0.97
Botnet	0.99	1.00	0.99	***Jamming***	0.95	0.96	0.95	***Layer 7 Attack***	0.96	0.98	0.97
Fuzzing	0.94	0.95	0.94	***Replay***	0.94	0.93	0.93	***ZSlowDoS***	0.88	0.88	0.88
WebSocket	0.96	1.00	0.98	***BruteForce***	0.99	0.99	0.99	***MQTT***	0.99	0.99	0.99
Zero-day-1 (Botnet)	0.98	0.98	0.98	***Zero-day-2 (Jamming)***	0.88	0.86	0.87	***Zero-day-3 (Layer-7)***	0.94	0.95	0.94
Second Layer Results - WebSocket Category											
BruteForce	1.00	1.00	1.00	***Encoded***	0.99	0.99	0.99	***Malformed***	0.97	0.98	0.97
TBC	0.96	1.00	0.98	***Zero-day (TBC)***	0.92	0.96	0.94	—			
Second Layer Results - Layer-4 Category											
TCP-Flood	0.96	0.97	0.96	***SYN-Flood***	0.98	0.98	0.98	***Connection***	0.97	0.97	0.97
NULL-TCP	1.00	1.00	1.00	***ACK-Flood***	1.00	1.00	1.00	***SYN-PSH***	1.00	1.00	1.00
SYN-URG	1.00	1.00	1.00	***SYN-PSH-URG***	1.00	1.00	1.00	***Slowloris***	0.99	1.00	0.99
Zero-day-1 (ACK-Flood)	0.98	0.98	0.98	***Zero-day-2 (Slowloris)***	0.96	0.96	0.96	***Zero-day-3 (SYN-Flood)***	0.94	0.95	0.94

5.3 Performance Results

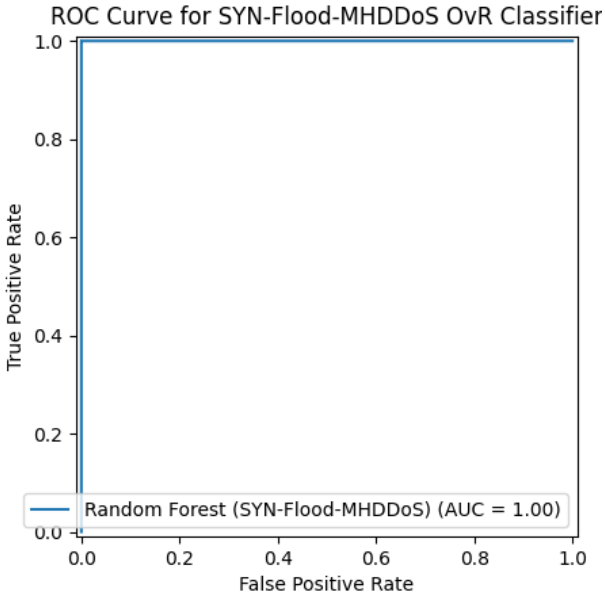
To assess the performance of the proposed model, comprehensive testing was conducted across various scenarios, with detailed evaluation metrics provided in Table 5. The evaluation was structured in a multi-layer approach, where the performance of each layer was independently measured alongside the performance of individual classifiers to highlight the impact of the proposed multi-layer architecture.

The model was tested in eight scenarios to evaluate its zero-day detection capabilities. In three of these scenarios, one primary attack category was excluded during the training phase to simulate a zero-day condition, allowing an assessment of the first layer. Of these excluded categories, two were IP-based classes, and one was IoT-based. This process was then repeated for sub-category exclusions, where four IP-based were withheld during training, further testing the model's second layer. These experiments provided valuable insights into the model's resilience and adaptability in diverse attack scenarios.

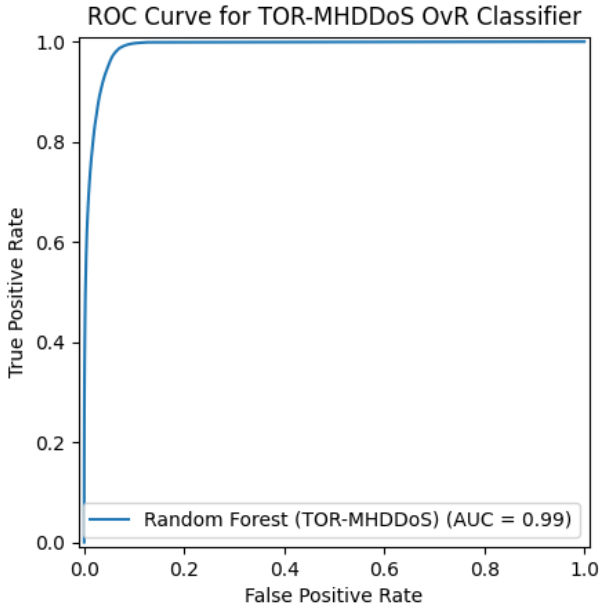
For illustration, the Receiver Operating Characteristic (ROC) curves and confusion matrices of selected classifiers are presented in Fig. 60 and Fig. 61, respectively, to exemplify the classification performance across different scenarios.



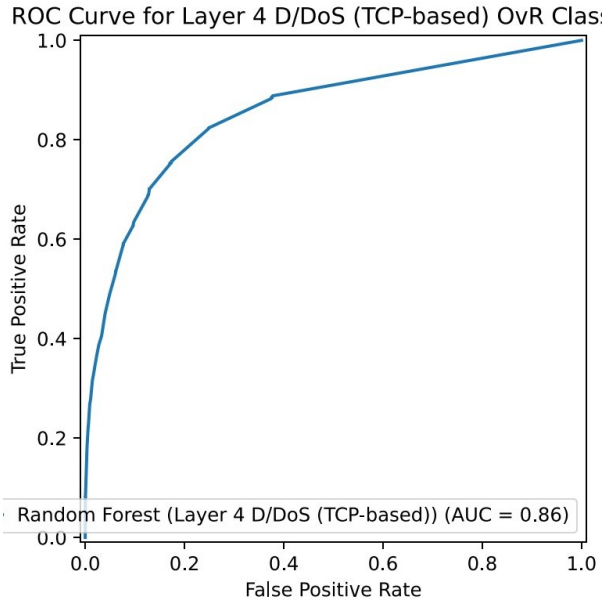
(a) ROC Curve for Benign classifier using Z-wave data



(b) ROC Curve for SYN-Flood classifier using IP data

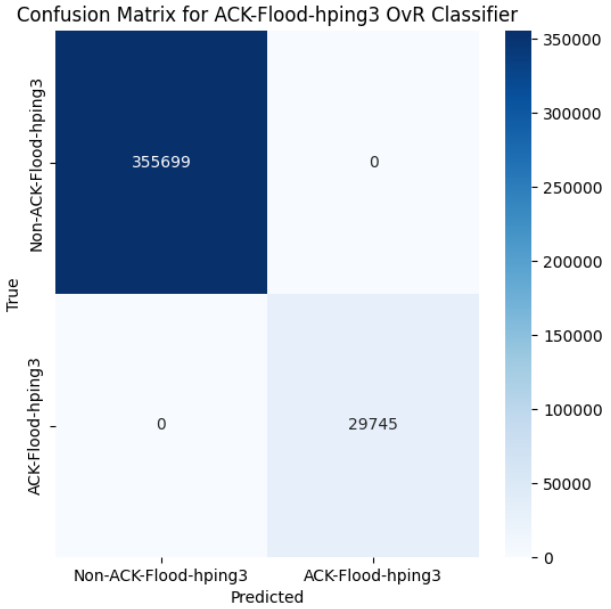


(c) ROC Curve for TOR classifier using IP data

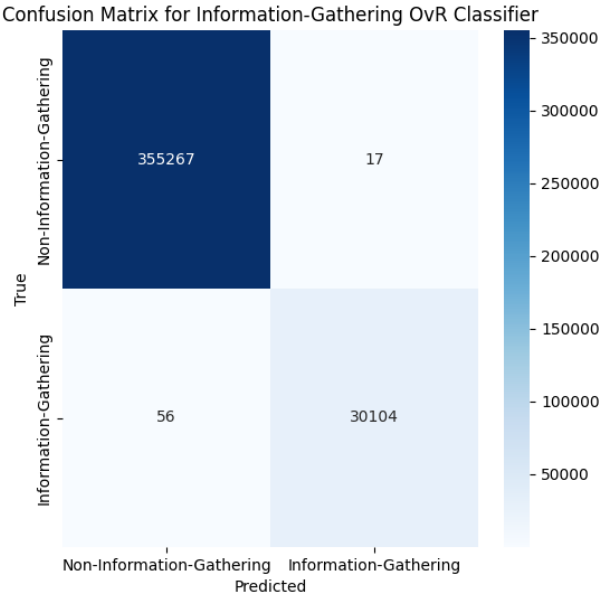


(d) ROC Curve for TCP attacks classifier using Z-wave data

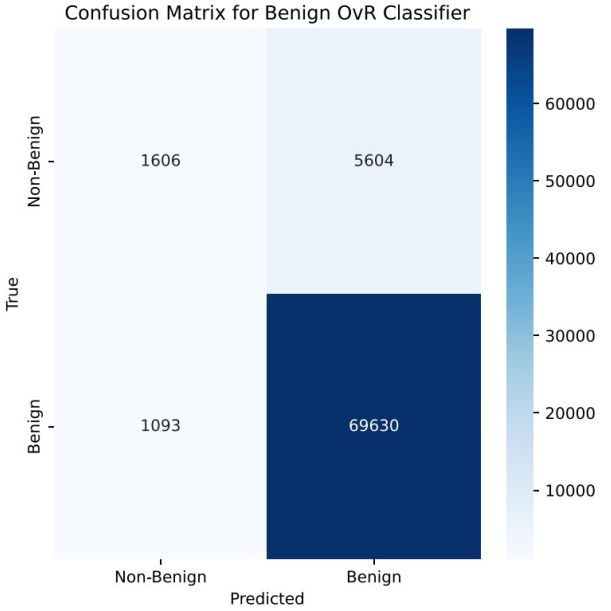
Figure 60: ROC curve for some of the classifiers.



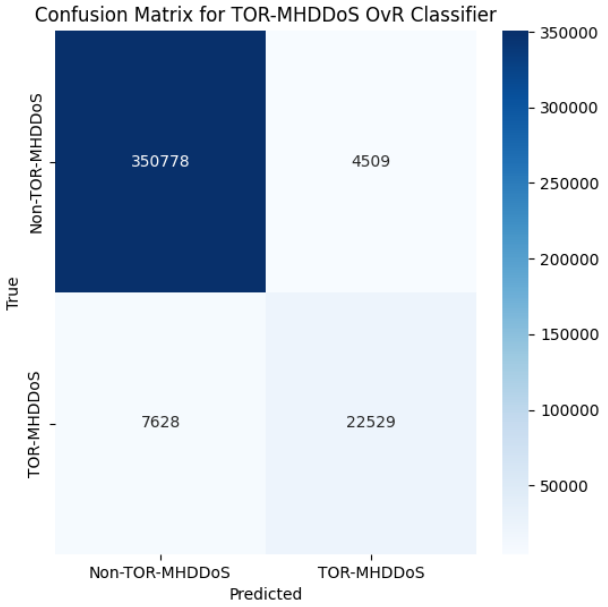
(a) ACK Flood in sub-category classifier using IP data



(b) Info. Gathering in main category classifier using IP data



(c) Benign in main category classifier using Z-wave data



(d) TOR in sub-category classifier using IP data

Figure 61: Confusion Matrix for some of the classifiers.

5.4 Concluding Remarks

This section focuses on evaluating the created dataset and the performance of the proposed detection system. it demonstrates the effectiveness of the proposed detection system through rigorous experimentation, achieving high detection rates and low false positives across various attack scenarios, including zero-day attacks. These results validate the system's robustness and its capability to improve smart home IoT security comprehensively.

6 Analysis and Discussion

This section analyzes the experimental results obtained from the proposed intrusion detection system and the created dataset. The discussion begins with an in-depth examination of the dataset, exploring its characteristics and properties from various perspectives. It then transitions to the initial phase of the experimentation, focusing on the feature selection process and the relation between the selected features and the related classes. Finally, the performance of the proposed detection system is evaluated, with a particular emphasis on its key capabilities, such as detecting zero-day activities. The discussion highlights the system's effectiveness in addressing diverse attack scenarios and its resilience in handling unseen threats.

6.1 Created Dataset Analaysis

This subsection evaluates the different steps involved in creating a dataset and discusses how each step can impact the final result in analyzing the IoT Smart Home environment.

6.1.1 Testbed Architecture

The design of the testbed architecture, illustrated in Fig. 36, demonstrates a forward-thinking and multi-faceted approach to replicating smart home environments. By integrating both wired and wireless communication protocols, this architecture mirrors the complexities inherent in real-world smart homes, providing a holistic platform for security analysis. Its capacity to handle multiple communication standards, including Z-Wave, TCP, MQTT, HTTP/S, and WebSockets, ensures that it captures the heterogeneity of smart home setups. This is especially critical given the diversity of devices and vendors within modern ecosystems, which often complicate security measures due to varying protocol implementations and vulnerabilities.

The inclusion of bidirectional data flows, enabling the capture of local (device-to-hub) and remote (hub-to-internet) communications, underscores the testbed's capability to evaluate attacks across multiple vectors. This dual-layered capturing mechanism is particularly valuable for identifying patterns that link local anomalies to broader systemic vulnerabilities. For instance, a man-in-the-middle (MITM) attack targeting local device communications may manifest as subtle anomalies in internet-level interactions, which the testbed can effectively analyze.

The testbed's modularity and extensibility are standout features, as they allow it to remain relevant amidst rapid technological advancements. The ability to incorporate new devices, protocols, and attack vectors ensures the platform can adapt to emerging threats and trends. This flexibility is essential in research, enabling continuous innovation and exploration.

From a behavioral perspective, the testbed's ability to capture device metadata provides insights into normal and anomalous operations. This metadata can reveal subtle patterns, such as changes in device response times or data transmission frequencies, that may signal underlying security issues. By simulating attacks based on known vulnerabilities, the testbed offers a controlled environment for understanding attacker behaviors and testing defense mechanisms, such as intrusion detection systems.

Overall, the architecture's design aligns with best practices in smart home research while introducing advanced features for capturing and analyzing data comprehensively. This positions it as a robust tool for studying vulnerabilities, developing defensive strategies, and fostering a deeper understanding of the smart home threat landscape.

6.1.2 Devices Taxonomy

The proposed taxonomy of smart home devices, shown in Fig. 37, reflects a systematic and protocol-agnostic framework for classifying and analyzing device interactions. This categorization into controllers, sensors, and actuators ensures comprehensive coverage of functional roles within a smart home, laying a solid foundation for dataset creation and security analysis. The taxonomy transcends individual technologies by focusing on functional groups rather than specific protocols, offering a universally applicable framework for diverse smart home setups.
Including diverse devices, from security-centric components like smart locks and cameras to environmental sensors like air quality detectors, underscores the testbed's commitment to capturing a wide spectrum of operational contexts. This diversity is critical for understanding how different devices interact within the ecosystem and identifying potential vulnerabilities unique to specific device types.
Technically, the protocol-agnostic approach enhances the testbed's scalability and flexibility. This ensures the research remains relevant as new communication standards emerge, allowing cross-protocol comparisons. The taxonomy's structured framework provides a clear pathway for integrating additional devices and functionalities, which is crucial for advancing smart home security research.
In summary, the device taxonomy is a practical tool for organizing the testbed and a theoretical framework for understanding smart home operations. Its emphasis on diversity, functionality, and protocol independence ensures that the resulting datasets are comprehensive and meaningful, supporting a wide range of research objectives.

6.1.3 Devices Selection

The careful selection of devices outlined in this study plays a critical role in ensuring the comprehensiveness and robustness of the generated dataset. By incorporating multiple devices per type from different manufacturers, the testbed accounts for variations inherent to real-world smart home environments. Differences in manufacturing tolerances, firmware versions, and operational algorithms often introduce subtle but significant deviations in device behavior. These variations, often overlooked in single-device studies, can have profound implications for performance and security. For example, understanding how different models of water sensors respond to identical network stress scenarios helps identify potential vulnerabilities unique to specific implementations.
Furthermore, placing identical devices in diverse conditions and environments adds another dimension to the dataset's richness. This approach ensures the collected data encapsulates a wide range of real-world scenarios, from environmental stressors like humidity and temperature changes to network dynamics under varying loads. Such diversity improves the reliability of performance evaluations and enhances the dataset's relevance to practical applications. By analyzing the behavior of devices subjected to different environmental and network conditions, researchers can derive insights into their robustness and identify patterns that may indicate potential points of failure or exploitation.
Including devices from various vendors introduces another layer of variability, reflecting the heterogeneity of real-world IoT ecosystems. Variations in security implementations, such as differences between S0 and S2 security frameworks in Z-Wave devices, underscore the importance of evaluating vendor-specific behaviors. By capturing these nuances, the testbed enables a more granular analysis of how proprietary enhancements and standard deviations impact overall system security and functionality. For instance, the contrast between devices employing advanced S2 security and those relying on the less secure S0 framework highlights vulnerabilities that could be exploited in mixed-device environments.

Another strength of this methodology is prioritizing devices with known vulnerabilities. By deliberately including devices with documented weaknesses, the study evaluates typical use cases and provides a platform for simulating targeted attack vectors. This approach allows for a realistic assessment of existing security mechanisms and offers opportunities for developing and testing new defensive strategies.
Moreover, the focus on ease of integration and compatibility with widely used platforms, such as Home Assistant and Amazon Alexa, ensures that the dataset remains relevant to contemporary smart home ecosystems. By simulating interactions with third-party services, the study captures an often-overlooked aspect of IoT security: vulnerabilities arising from integrations. Testing these interactions provides valuable insights into how third-party services influence overall system security and where potential risks may emerge.
The deliberate inclusion of basic and advanced functionalities within device categories further enhances the dataset's applicability. By comparing the security implications of simple devices, such as standard motion detectors, with those of advanced systems, like smart actuators with voice recognition, the testbed reflects the diverse capabilities of modern smart homes. This comparison sheds light on the trade-offs between functionality and security, helping to identify whether more complex devices inherently introduce greater risks.
In summary, the devices selected for this study embody a holistic approach to dataset creation, encompassing vendor diversity, functional complexity, and real-world relevance. This thoroughness strengthens the validity of the study's findings and provides a robust foundation for advancing smart home security research. The comprehensive profiling enabled by this approach supports a nuanced understanding of device behaviors, performance metrics, and security challenges in diverse operational scenarios.

6.1.4 Benign Traffic

The generation of benign traffic represents a cornerstone of this study, reflecting the natural interactions within a real IoT smart home environment. The dataset authentically captures the nuances of everyday IoT operations by deploying devices strategically and simulating routine activities. The careful placement of sensors, such as door and motion detectors, mirrors real-world use cases, ensuring the dataset's relevance to practical applications.
The diverse actions performed during data collection, including door operations, water sensor placements, motion tracking, etc., highlight the study's commitment to replicating real-life scenarios. For example, placing water sensors near frequently used areas, such as coffee stations and sinks, captures realistic variations in device behavior based on environmental conditions. Similarly, using multiple types of smoke to test smoke detectors demonstrates a thorough understanding of device capabilities and their real-world limitations.
Integrating automated activities, such as linking motion sensors to lamps via Home Assistant, reflects the interconnected nature of IoT systems in modern homes. By simulating these interactions, the study not only evaluates the functionality of individual devices but also examines how these interactions influence system-wide behaviors and security. This approach provides insights into potential vulnerabilities arising from automation and third-party integrations, which are critical for developing effective mitigation strategies.
Including devices under varying conditions, such as testing smart plugs with high-power-consuming appliances, further enhances the dataset's robustness. By evaluating performance under stress scenarios, the study captures the operational boundaries of these devices, identifying points of failure that may not be apparent under normal conditions. This level of detail is particularly valuable for security research, as it helps uncover vulnerabilities that could be exploited during peak loads or environmental extremes.
The challenges encountered during the data collection, such as device malfunctions and hub restarts, add another

layer of realism to the dataset. These disruptions mirror the common issues faced in real-world IoT environments, providing an opportunity to analyze how such challenges impact system reliability and security. For instance, the need to reintegrate malfunctioning devices offers insights into potential vulnerabilities introduced during reconfiguration processes.

In conclusion, the benign traffic generated in this study provides a comprehensive and authentic representation of typical IoT smart home behaviors. The combination of strategic device placement, diverse interactions, and real-world challenges ensures the dataset's relevance and applicability to contemporary security research. This foundation supports a nuanced analysis of IoT systems, highlighting their strengths and vulnerabilities in realistic scenarios.

6.1.5 Data Capturing

The data-capturing process represents a foundation of this research, serving as the primary enabler for in-depth network traffic analysis across both IP-based and Z-Wave protocols. By employing a dual-source data acquisition strategy, the research ensures a comprehensive approach to capturing traffic from diverse networks, each with unique characteristics and technical requirements.

The dataset's strength lies not only in its volume but also in its breadth and granularity. The research encapsulates a realistic snapshot of smart home network behavior by capturing a wide range of scenarios, from normal operations to anomalies like device inclusion errors and hub restarts. Including temporal patterns, such as weekday versus weekend traffic variations and device-specific packet transmission trends, enriches the dataset with behavioral dimensions.

Figures and tables referenced, such as comparisons between benign and attack traffic or protocol-specific breakdowns, provide a solid basis for analyzing how network stressors alter device and protocol behavior. For example, the visualizations showing changes in packet transmission during attack phases, in figures 50, 51, 52, 53, 54, 55, 56, 57, 58, and 59 offer a clear narrative of how smart home devices react to threats, enhancing our understanding of their resilience and vulnerabilities.

Exploring recurring patterns, such as the repeated interaction sequence among Door Lock 26, Door Lock 19, and Water Sensor 35, provides an intriguing behavioral perspective. The observed approximate 80% correlation between these devices during specific time windows highlights the potential for profiling device interactions and identifying deviations during abnormal conditions. Such insights are invaluable for anomaly detection and building predictive models in smart home environments.

The vigilance in mitigating capturing errors, especially during high-traffic periods, reflects a proactive approach to ensuring dataset integrity. Constant monitoring and timely intervention to address faults underline the importance of reliability in long-term data collection projects. The expanded scope to include MQTT broker, Home Assistant, and Amazon Alexa traffic further broadens the dataset's relevance, aligning well with the project's goals of comprehensive analysis.

The strategic alignment of attack scenarios with specific timestamps facilitates precise labeling of benign and attack conditions, which is critical for subsequent analysis. This structured approach ensures that the dataset can effectively support studies on intrusion detection, network performance under stress, and protocol-specific vulnerabilities.

The study's comprehensive approach to threat scenario identification and data capturing lays a solid groundwork for advancing smart home security. The dual focus on technical vulnerabilities and behavioral impacts ensures

a holistic understanding of the ecosystem's dynamics. The dataset's richness, underscored by its temporal and protocol-specific analyses, provides valuable insights for academic research and practical application in developing more secure and user-friendly smart home technologies.
In conclusion, this research's data-capturing strategy is technically rigorous and behaviorally insightful. By addressing the challenges inherent in diverse network environments and emphasizing an accurate approach to monitoring and recording, the process has laid a robust foundation for in-depth traffic analysis. The dataset's richness and alignment to real-world smart home scenarios position it as a valuable resource for advancing our understanding of IP-based and Z-Wave networks under various conditions.

6.1.6 Traffic Analyzers and CSV Generation

The development of ZwaveNetLyzer, the first IoT network traffic analyzer, is a significant contribution to analyzing IoT traffic. Unlike traditional IP traffic, IoT protocols like Z-Wave operate in a distinct environment where devices often communicate in bursts, and the traffic characteristics vary considerably from conventional network traffic.
ZwaveNetLyzer addresses this gap by focusing on parsing and structuring Z-Wave data, enabling extracting relevant features that serve as the backbone for IoT security research. By creating a custom analyzer for Z-Wave traffic, this work not only fills a critical void in the research landscape but also paves the way for a more comprehensive understanding of IoT device behavior and vulnerabilities. The ability to convert raw Z-Wave data into structured flow information allows for the identification of patterns specific to IoT network traffic, including anomalous behaviors that could indicate security breaches or inefficiencies in device communication.
Furthermore, ZwaveNetLyzer's open-source nature ensures that the broader research community can benefit from its development. This collaborative aspect is crucial for advancing IoT security, as it enables the analysis of a wide range of devices and communication patterns in future studies. The tool's design provides a solid foundation for future security applications, such as intrusion detection systems and anomaly detection platforms tailored to IoT environments.

6.2 Feature Selection Analysis

The feature selection process for each traffic class in the dataset has been carefully designed to capture the essential characteristics of the traffic patterns and behaviors associated with benign and attack scenarios. The selected features were chosen based on their potential to differentiate between normal and malicious activities and their ability to provide valuable insights into the nature of the network traffic.
This subsection provides an in-depth analysis of the selected features for both IP-based and IoT (Z-Wave) classifiers. Each class is associated with specific features derived from various IP and IoT network packet attributes. The importance of each feature category is explained based on its relevance to the detection of benign and attack behaviors and its potential to capture unique characteristics of traffic patterns. The features selected by the Information Gain and ANOVA algorithms are discussed in the following subsections.

6.2.1 IP-based Selected Features

For the ***Benign*** class, a variety of statistical features were selected, including aggregate values such as total, maximum, minimum, mean, median, and mode for header bytes (`hdr_bytes`) and flow header bytes (`fw_hdr_bytes`). These features help characterize typical packet size and transmission rate traffic patterns. Features such as `fw_pkts_rate`,

rst_flag_cnts, and segment_size_min capture the traffic dynamics in terms of flow rates and connection behaviors. Additionally, metrics like fw_syn%in_total and bw_syn%in_bw_pkts focus on the specific characteristics of SYN packets within the flows, which can be indicative of normal communication patterns. The selection also includes time-related features, such as the packet inter-arrival times (pkts_IAT) and their respective statistical measures, which help in understanding the temporal distribution of packets during benign traffic scenarios.

For the ***Information Gathering*** class, the feature set includes similar statistical measures for packet sizes and flow headers, with an emphasis on flow segment sizes (fw_segment_size_mean) and initialization window sizes (fw_init_win_bytes), which are often seen in reconnaissance activities. The inclusion of flags such as fw_syn%in_fw_pkts and bw_ack%in_total helps distinguish traffic that involves scanning or probing behavior. Additionally, features related to packet inter-arrival times, such as bw_pkts_IAT, provide valuable information on the rate and timing of requests characteristic of information-gathering activities. The selection of handshake_state is also critical, as it provides insight into the state of the connection, which can reveal reconnaissance activity.

The ***Botnet*** class is distinguished by features that capture abnormal flow behaviors typical of botnet communication. The duration and packet rate features are essential for identifying prolonged connections that might indicate botnet activity. Additionally, features like fw_syn%in_fw_pkts and bw_ack%in_bw_pkts help identify anomalies in packet flags and their distributions. Botnet traffic often exhibits distinct traffic flows, and the features related to packet inter-arrival times (pkts_IAT) and the frequency of flags like SYN and RST are crucial for distinguishing between botnet activity and normal traffic. Including handshake state and flow rate features further enhances the ability to detect botnets based on their command-and-control communications.

For the ***ACK Flood*** class, features such as fw_syn%in_fw_pkts and bw_ack%in_bw_pkts focus on the high rate of ACK packets typical of this attack. The ratio of flags such as syn, fw_syn, and fw_ack helps to identify the flood-like nature of the traffic. The statistical measures for the flow headers and packet sizes, including fw_hdr_bytes and bw_hdr_bytes, capture the potential anomalies in packet size distributions during an ACK flood attack. The duration of the traffic flow is also a key feature, as ACK floods can sustain large packets over extended periods.

The ***TOR DDoS*** class involves Distributed Denial of Service (DDoS) attacks that often originate from the TOR network. Features such as fw_payload_bytes_max and fw_pkts_rate capture the heavy volume of data characteristic of DDoS attacks. The use of flags like fw_syn%in_fw_pkts and bw_ack%in_bw_pkts identifies the high rate of SYN and ACK packets that are typically part of the attack traffic. The temporal features, such as pkts_IAT and their associated statistics, provide insight into the rapid request generation in a DDoS attack.

The ***TCP Slowloris*** attack is characterized by its ability to open many connections by sending incomplete HTTP requests. The features selected for this attack focus on packet size distributions and the inter-arrival times of packets. Metrics such as fw_segment_size_mean and bw_pkts_IAT_mean help detect the slow but consistent nature of the attack. The feature fw_payload_bytes_max is useful for identifying large payloads, often part of a Slowloris attack.

For the ***Websocket BruteForce*** class, features related to the flow's duration and header sizes (fw_hdr_bytes) are essential for identifying repetitive connection attempts that are characteristic of brute-force attacks. The selected packet inter-arrival times and segment size features help capture the bursty nature of brute-force login attempts. Additionally, the feature handshake_state is critical for detecting failed or incomplete handshakes during brute-force attacks.

The ***Get Flood*** class involves flooding a target with HTTP GET requests. Features such as fw_hdr_bytes and bw_hdr_bytes capture the size of the requests, while fw_syn%in_fw_pkts and bw_ack%in_bw_pkts help identify

the large number of SYN and ACK packets that are characteristic of GET floods. Including packet inter-arrival time features like fw_pkts_IAT helps detect the rapid and repeated requests that typify GET flood attacks.
Finally, the ***Cookie-based DDoS*** attack involves overwhelming a target by flooding it with requests that appear to be from legitimate users. Features such as fw_hdr_bytes and segment_size_mean capture the characteristics of the HTTP headers and the size of the requests, which are critical for detecting the attack. Additionally, features such as fw_syn%in_fw_pkts and bw_ack%in_bw_pkts help distinguish the attack traffic from normal communication by analyzing the flag distributions and packet flows.

6.2.2 IoT-Zwave-based Selected Features

For the ***Benign*** class, the feature set includes a variety of statistical measures and characteristics of packet data. Notable features include data_bytes_rate, entropy_hex_data, and most/least_common_channel, which describe typical traffic patterns in the absence of attack activity. Features such as fw_hex_data_pattern_len_var and fw_most/least_common_channel capture the consistency and variability of data patterns, which are indicative of normal operation. Other features, such as min_speed, total/mean/var/std/skew/coeff_data_field_size, and fw_max/std_pkts_DD, reflect the expected distribution and variability in packet sizes, transmission speeds, and flow characteristics that are common in benign communication.
For ***Z-Wave Attacks (General Category)***, the feature set is designed to capture irregularities in traffic patterns that may indicate an attack. Similar to the benign class, key features like min_speed, fw_data_field_entropy, and fw_entropy_hex_data are included to detect unusual patterns. Additional features such as mean/std_data_field_size and coeff_num_hops help identify abnormal packet size distributions and unusual routing behaviors. The inclusion of features like fw_total/mean_data_field_size further aids in detecting changes in the volume of data transmitted, which can be indicative of malicious activity. These features help to distinguish between benign behavior and attack-induced anomalies in the Z-Wave network.
For ***Layer 7 D/DoS (HTTP-based)*** attacks, which are IP-based attacks, the feature set includes a variety of bandwidth and speed-related metrics. Features such as bwd_avg/med/mode/min/max_bw/speed are crucial for detecting sudden spikes or drops in traffic volume, a hallmark of DDoS attacks. Additionally, features like fw_common_channel and fw_bwd_data_field_size capture any changes in the structure or characteristics of the communication channel. Layer 7 DDoS attacks often result in sudden disruptions of service, which can be detected through abnormalities in packet distribution and channel usage (because of the hub being unresponsive), represented by features like max/min/mode/med_fw/bw/data_field_size.
For ***Layer 4 D/DoS (TCP-based)*** attacks, which are also IP-based, similar features are employed. These include mode/min/max_bw/speed and fw_common_channel. Additional features, such as bw_entropy_of_hex_data, fw_bw_data_field_entropy, and coeff_tries, are incorporated to capture the specific behaviors of TCP-based DDoS attacks (again because of the hub being unresponsive). These attacks often manipulate connection states, and these features help detect such manipulations by identifying irregularities in the flow of data and control packets.
For ***Jamming*** attacks, which aim to disrupt communication in the IoT network, the feature set includes metrics such as pkts_cnt, avg/std/med/mode/var_bw/speed, and rssi_range. These features capture signal strength and packet transmission disruptions, which are characteristic of jamming. The channel_stability and fw_hex_data_pattern_length_variability features are also included to identify interference with the wireless channel. Jamming attacks often cause high speed and signal quality variability, which can be detected using features like mean/var/std/coeff_scheme_state.

In the case of ***Fuzzing with Correct CRC*** attacks, features like `bw/speed` features categories and `fw_channel_stability` are useful for detecting irregular traffic patterns that may result from fuzzing attacks. Fuzzing attempts often involve sending malformed or malicious packets to the network, leading to irregular packet timings and data patterns. Including features such as `fw_mean_pkts_DT` and `rssi_range` helps detect these irregularities by examining packet intervals and signal strength variations.

For ***Replay*** attacks, which involve the reuse of previously captured traffic, the feature set includes `bw/speed` features category, `fw_entropy_of_hex_data`, and `fw_data_field_entropy`. These features help capture the repetition of previously observed traffic patterns. Additionally, features such as `var/std/mean_number_of_hops` and `bw_mode_rssi` are used to detect patterns indicative of packet repetition, distinguishing replay attacks from new legitimate traffic.

In summary, the feature selection for the IoT-Zwave dataset captures a broad range of characteristics related to both normal communication and various attack types. These features include metrics related to bandwidth, packet sizes, data entropy, channel stability, signal strength, and more. By incorporating features that reflect the statistical distribution of network behavior and specific patterns indicative of different attack types, the model can detect deviations from normal behavior in the presence of network attacks, particularly focusing on how IP-based DDoS attacks impact internal IoT Z-Wave communication. This comprehensive feature set is essential for effectively securing IoT networks against various threats.

Overall, each data type's selected features for each traffic class provide a comprehensive view of the IoT Smart Home characteristics, allowing for effective differentiation between benign and malicious behaviors. These features not only focus on packet sizes and flow rates but also on the temporal aspects of the smart home, which are critical for accurately classifying various attack types in malicious behavior analysis.

6.3 Performance Analysis

The performance analysis of the model, as shown in the Table 5, provides insights into how well individual classifiers perform in detecting specific attack categories, as well as how the model benefits from combining these classifiers into a layered architecture. The table presents a range of performance metrics, including precision (Pre.), recall (Rec.), and F1 score (F1) for each attack class in the system, both for individual classifiers and for the combined model, which incorporates multiple layers and weighted outputs.

When analyzing individual classifiers, some noteworthy patterns emerge. Certain classifiers, such as those for Botnet, BruteForce, and BYPASS, consistently achieve perfect scores (1.00) in all performance metrics, indicating their effectiveness in detecting their respective attack classes. These results reflect the robust capability of these classifiers in identifying these specific types of attacks with minimal error. On the other hand, classifiers for more complex attack types or those with limited data, such as Jamming, Fuzzing, and TOR, show slightly lower precision, recall, and F1 scores. This suggests that while these classifiers are still effective, they may encounter challenges distinguishing between attacks and benign traffic or require more refined feature selection or tuning.

The strength of the proposed model lies in its use of a multi-layer architecture, where individual classifiers are combined to optimize performance. This improvement is largely attributed to the system's ability to assign weights to the outputs of each classifier, effectively compensating for any weaknesses present in any one classifier. The system can correct individual classifier mistakes and refine the decision-making process at the combined output layer by optimizing these weights during training. This approach allows for a more holistic detection strategy, where errors from one classifier can be mitigated by the strengths of others, leading to more accurate and reliable

classification.

A noteworthy finding from the analysis is that, for certain classes, such as *Benign*, the performance of classifiers varies significantly depending on the data source used. Specifically, the IoT-based classifier struggles to accurately classify the input, while the IP-based classifier performs with high accuracy. This phenomenon is observed consistently across various attack types, where the data source more directly related to the attack type seems to offer superior classification results. For instance, IoT-specific attacks are better handled by the IoT data source, as the classifier trained on IoT traffic can capture the unique characteristics and patterns intrinsic to these types of attacks. In contrast, IP-based attacks are more effectively detected by classifiers trained on the IP traffic, leveraging the distinct traffic features and attack signatures inherent to the IP layer.

This distinction can be attributed to the fact that each data source encapsulates distinct layers of information. IoT traffic often involves different communication protocols, device types, and traffic patterns compared to IP traffic, affecting the classifiers' ability to generalize across different attack types. This domain-specific variance highlights the necessity of selecting the most appropriate data type for each attack scenario, as the attack features may manifest differently depending on the traffic type.

Therefore, it is evident that relying on a single data source for classification may lead to suboptimal performance, especially when dealing with a variety of attack scenarios. The results indicate that combining multiple data sources, each tailored to a specific attack type, enhances the overall classification accuracy. The proposed architecture, which integrates IoT and IP data sources through a weighted ensemble approach, effectively addresses this issue. By combining the classifiers' outputs using a weighted strategy, the architecture balances the strengths and weaknesses of each data type. The weighting mechanism assigns higher importance to the more relevant data source based on the context of the attack, ensuring that the most appropriate classifier is prioritized for each scenario. This method improves the overall classification performance and allows the system to dynamically adapt to different attack patterns, making it a robust solution for network traffic analysis in mixed environments.

In summary, this work demonstrates that a hybrid approach, which leverages the strengths of both IoT and IP data, can provide a more effective classification framework than using a single data type. With its weighted combination of classifiers, the proposed architecture offers a theoretically sound and practically viable solution for accurate classification across a wide range of benign and attack traffic types.

From a technical perspective, these results validate the choice of a multi-layered approach for improving detection accuracy across varied attack types, including both known and zero-day attacks. The integration of multiple classifiers, coupled with a weighted decision-making process, is a key factor in enhancing performance in detecting subtle and evolving threats. The analysis confirms that while individual classifiers might exhibit some weaknesses in isolation, combining their outputs into a single decision layer significantly mitigates these weaknesses, offering a more robust detection system.

In terms of conceptual insights, the performance results highlight the importance of model flexibility and adaptability in tackling a wide range of cyberattacks. The model's ability to refine and adjust through the weighted output system emphasizes the significance of not only detecting attacks but also optimizing the decision-making process to improve classification accuracy and reduce false positives or false negatives.

6.4 Zero Day Activity Detection

Zero-day activity detection is a critical aspect of network security, particularly as cyber threats evolve in sophistication. The model's ability to detect zero-day attacks is demonstrated through the performance of classifiers on

experiment scenarios like Zero-day-1 (Botnet), Zero-day-2 (Jamming), and Zero-day-3 (Layer-7). These attack types are often challenging to detect because they exploit previously unknown vulnerabilities, which traditional detection systems may fail to recognize.

The results for zero-day attacks reveal that, while the system performs exceptionally well in detecting familiar attack types, there is a slight variation in the performance for zero-day attack classes. For instance, Zero-day-1 (Botnet) exhibits a strong performance with precision, recall, and F1 scores consistently close to 1.00, suggesting that the proposed IDS can generalize well to new attack types within this category. On the other hand, Zero-day-2 (Jamming) and Zero-day-3 (Layer-7) show slightly lower performance, with precision and recall hovering around 0.88 and 0.94, respectively. This indicates that while the system can detect these zero-day attacks, specific characteristics or patterns in these attack types may pose challenges for classification.

The layered model's ability to incorporate these attacks into the detection process highlights its strength in addressing dynamic and evolving threats. While the classifier for Zero-day-2 (Jamming) does not perform as well as others, its detection rate still improves significantly in the first layer, reflecting the system's ability to adapt and fine-tune its detection capabilities. This improvement is because the system is not solely reliant on a single classifier but leverages the combined power of multiple classifiers to make a final decision.

From a theoretical perspective, zero-day detection poses unique challenges because these attacks often do not exhibit the same behaviors as known threats, requiring the system to generalize effectively from prior knowledge. The model's performance, particularly its precision and recall for zero-day attacks, demonstrates its capability to handle this challenge. The system benefits from the flexibility of machine learning models, such as Random Forest (RF), which can learn complex patterns from diverse data inputs and apply them to unfamiliar scenarios.

Conceptually, this highlights the importance of using diverse classifiers to detect zero-day attacks. Integrating classifiers trained on different attack types and a weighted decision layer improves the model's robustness in detecting novel attack patterns. This multi-faceted approach enables the model to adapt to new and previously unseen attack strategies, making it a valuable tool for proactive cybersecurity measures.

In conclusion, the results confirm that the proposed model effectively detects a wide range of attack categories, including zero-day attacks. While some attack types present challenges, the layered, weighted approach significantly enhances detection accuracy, demonstrating the model's capability to handle both known and unknown threats with high reliability. Combining multiple classifiers in a flexible framework provides a promising avenue for improving security systems in the face of constantly evolving cyber threats.

7 Conclusion & Future Work

This thesis presents a significant advancement in securing smart home IoT environments through the design and evaluation of a novel dual-tier intrusion detection system. The proposed system effectively addresses the limitations of traditional intrusion detection methods, offering the capability to detect zero-day threats and malicious activities across diverse scenarios. By analyzing both external internet traffic and internal IoT communications, the system ensures comprehensive coverage of potential intrusion points, making it uniquely suited to the complexities of smart home ecosystems.

To support the development and testing of the proposed model, this research introduces the largest smart home IoT dataset to date. Existing datasets often fall short in capturing the full range of IoT threats, either lacking device diversity or comprehensive attack scenarios. To overcome these gaps, we created a dataset comprising real-world data from over 50 devices collected over a five-month period. It includes more than 100 carefully designed attack scenarios, ranging from device-specific exploits to network-level intrusions. The dataset's diversity and scale enabled rigorous testing of the proposed system, revealing critical insights into the behavior of both benign and malicious traffic. Beyond this research, the dataset serves as a valuable resource for the broader community, advancing the state of IoT security research.

Experimental evaluation demonstrated the robustness of the proposed intrusion detection system. Key metrics showed the system's ability to achieve an accuracy of 98.7%, with a detection rate exceeding 99% across multiple attack categories, including zero-day attacks. This performance underscores the system's effectiveness in identifying novel and evolving threats. Furthermore, by incorporating an advanced feature selection process and analyzing the dataset through machine learning models, the system achieved a false positive rate of just 1.2%, highlighting its reliability for real-world deployment.

The analysis further underscored the importance of considering both internal and external network traffic for comprehensive IoT security. Internal communications between devices often exhibit distinct patterns that are pivotal for accurate profiling, while external traffic analysis ensures the detection of broader, internet-based threats. The Z-Wave protocol, specifically, was shown to present unique challenges due to its proprietary nature and limited encryption support, emphasizing the need for protocol-specific considerations in security models.

In addition to system performance, the study introduced a taxonomy of smart home IoT devices. This taxonomy offers a structured framework for researchers and developers, aiding in the design of targeted security solutions. Comparative evaluation of existing IoT datasets highlighted their limitations in scale, diversity, and realism, further reinforcing the contribution of the dataset developed in this work.

Looking ahead, this research lays a solid foundation for future advancements in IoT security. Expanding the dataset to include additional devices, protocols, and attack scenarios would further enhance the generalizability of intrusion detection models. Real-time deployment and testing of the proposed system in live smart home environments could validate its practical applicability. Additionally, exploring adaptive learning techniques to respond to evolving threats will be essential to maintaining security in dynamic IoT ecosystems.

In conclusion, this thesis advances the state of the art in IoT security by addressing the critical need for specialized detection systems and high-quality datasets. By bridging gaps in existing methodologies and providing actionable insights, the research contributes to building safer, more resilient smart home environments. These contributions pave the way for continued innovation, fostering a future where IoT technology can be embraced without compromising security.

Bibliography

[1] "Z-wave overview and diagram, https://z-wavealliance.org/technology-overview/."

[2] "Zigbee overview and diagram, https://csa-iot.org/all-solutions/zigbee/."

[3] "Wifi overview and diagram, https://www.oreilly.com/library/view/high-performance-browser/9781449344757/ch06.html."

[4] K. Cheng, C. Cheng, L. Zhang, J. Chen, and W. Luo, "Fingerprint recognition and classification of iot devices based on z-wave," in *CIBDA 2022; 3rd International Conference on Computer Information and Big Data Applications*, pp. 1–5, VDE, 2022.

[5] G. Spanos, K. M. Giannoutakis, K. Votis, and D. Tzovaras, "Combining statistical and machine learning techniques in iot anomaly detection for smart homes," in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pp. 1–6, IEEE, 2019.

[6] L. Babun, H. Aksu, L. Ryan, K. Akkaya, E. S. Bentley, and A. S. Uluagac, "Z-iot: Passive device-class fingerprinting of zigbee and z-wave iot devices," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pp. 1–7, IEEE, 2020.

[7] E. Anthi, L. Williams, M. Slowińska, G. Theodorakopoulos, and P. Burnap, "A supervised intrusion detection system for smart home iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9042–9053, 2019.

[8] A. Procopiou, N. Komninos, C. Douligeris, *et al.*, "Forchaos: Real time application ddos detection using forecasting and chaos theory in smart home iot network," *Wireless Communications and Mobile Computing*, vol. 2019, 2019.

[9] I. Cvitić, D. Peraković, M. Perivsa, and M. Botica, "Novel approach for detection of iot generated ddos traffic," *Wireless Networks*, vol. 27, no. 3, pp. 1573–1586, 2021.

[10] A. Aguru and S. Erukala, "Oti-iot: A blockchain-based operational threat intelligence framework for multi-vector ddos attacks," *ACM Transactions on Internet Technology*, 2024.

[11] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning ddos detection for consumer internet of things devices," in *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 29–35, IEEE, 2018.

[12] X.-H. Nguyen and K.-H. Le, "Robust detection of unknown dos/ddos attacks in iot networks using a hybrid learning model," *Internet of Things*, vol. 23, p. 100851, 2023.

[13] K. Doshi, Y. Yilmaz, and S. Uludag, "Timely detection and mitigation of stealthy ddos attacks via iot networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2164–2176, 2021.

[14] K. Mittal and P. Khurana Batra, "Graph-ensemble fusion for enhanced iot intrusion detection: leveraging gcn and deep learning," *Cluster Computing*, pp. 1–28, 2024.

[15] J. D. Fuller, B. W. Ramsey, M. J. Rice, and J. M. Pecarina, "Misuse-based detection of z-wave network attacks," *Computers & Security*, vol. 64, pp. 44–58, 2017.

[16] L. A. C. Ahakonye, C. I. Nwakanma, J. M. Lee, and D.-S. Kim, "Machine learning explainability for intrusion detection in the industrial internet of things," *IEEE Internet of Things Magazine*, vol. 7, no. 3, pp. 68–74, 2024.

[17] U. Zukaib, X. Cui, C. Zheng, M. Hassan, and Z. Shen, "Meta-ids: Meta-learning based smart intrusion detection system for internet of medical things (iomt) network," *IEEE Internet of Things Journal*, 2024.

[18] M. Nobakht, V. Sivaraman, and R. Boreli, "A host-based intrusion detection and mitigation framework for smart home iot using openflow," in *2016 11th International conference on availability, reliability and security (ARES)*, pp. 147–156, IEEE, 2016.

[19] T. Gaber, A. El-Ghamry, and A. E. Hassanien, "Injection attack detection using machine learning for smart iot applications," *Physical Communication*, vol. 52, p. 101685, 2022.

[20] S. Rahman, S. Pal, S. Mittal, T. Chawla, and C. Karmakar, "Syn-gan: A robust intrusion detection system using gan-based synthetic data for iot security," *Internet of Things*, p. 101212, 2024.

[21] G. Guntoro, L. Lisnawita, and L. Costaner, "Enhancing cybersecurity: Innovative hybrid feature selection for intrusion detection," in *Proceedings of the 2nd International Conference on Environmental, Energy, and Earth Science, ICEEES 2023, 30 October 2023, Pekanbaru, Indonesia*, 2024.

[22] N. Elsayed, Z. S. Zaghloul, S. W. Azumah, and C. Li, "Intrusion detection system in smart home network using bidirectional lstm and convolutional neural networks hybrid model," in *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 55–58, IEEE, 2021.

[23] R. Heartfield, G. Loukas, A. Bezemskij, and E. Panaousis, "Self-configurable cyber-physical intrusion detection for smart homes using reinforcement learning," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1720–1735, 2020.

[24] D. Javeed, M. S. Saeed, M. Adil, P. Kumar, and A. Jolfaei, "A federated learning-based zero trust intrusion detection system for internet of things," *Ad Hoc Networks*, p. 103540, 2024.

[25] V. A. Rajasekaran, A. Indirajithu, P. Jayalakshmi, A. Nayyar, and B. Balusamy, "Gradient scaling and segmented softmax regression federated learning (gds-srffl): a novel methodology for attack detection in industrial internet of things (iiot) networks," *The Journal of Supercomputing*, pp. 1–27, 2024.

[26] C. K. Nkuba, S. Woo, H. Lee, and S. Dietrich, "Zmad: Lightweight model-based anomaly detection for the structured z-wave protocol," *IEEE Access*, 2023.

[27] C. Fu, Q. Zeng, and X. Du, "{HAWatcher}:{Semantics-Aware} anomaly detection for appified smart homes," in *30th USENIX Security Symposium (USENIX Security 21)*, pp. 4223–4240, 2021.

[28] M. Gajewski, J. Mongay Batalla, G. Mastorakis, and C. X. Mavromoustakis, "Anomaly traffic detection and correlation in smart home automation iot systems," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 6, p. e4053, 2022.

[29] R. Paudel, T. Muncy, and W. Eberle, "Detecting dos attack in smart home iot devices using a graph-based approach," in *2019 IEEE international conference on big data (big data)*, pp. 5249–5258, IEEE, 2019.

[30] S. Kumar and A. Kumar, "Image-based malware detection based on convolution neural network with autoencoder in industrial internet of things using software defined networking honeypot," *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108374, 2024.

[31] S. I. Popoola, B. Adebisi, M. Hammoudeh, H. Gacanin, and G. Gui, "Stacked recurrent neural network for botnet detection in smart homes," *Computers & Electrical Engineering*, vol. 92, p. 107039, 2021.

[32] S. Datta, A. Kotha, K. Manohar, and U. Venkanna, "Dnsguard: a raspberry pi-based ddos mitigation on dns server in iot networks," *IEEE Networking Letters*, vol. 4, no. 4, pp. 212–216, 2022.

[33] M. Yamauchi, Y. Ohsita, M. Murata, K. Ueda, and Y. Kato, "Anomaly detection in smart home operation from user behaviors and home conditions," *IEEE Transactions on Consumer Electronics*, vol. 66, no. 2, pp. 183–192, 2020.

[34] L. Costa, J. P. Barros, and M. Tavares, "Vulnerabilities in iot devices for smart home environment," in *Proceedings of the 5th International Conference on Information Systems Security e Privacy, ICISSP 2019.*, vol. 1, pp. 615–622, SciTePress, 2019.

[35] B. Hammi, S. Zeadally, R. Khatoun, and J. Nebhen, "Survey on smart homes: Vulnerabilities, risks, and countermeasures," *Computers & Security*, vol. 117, p. 102677, 2022.

[36] B. D. Davis, J. C. Mason, and M. Anwar, "Vulnerability studies and security postures of iot devices: A smart home case study," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10102–10110, 2020.

[37] A. J. A. Majumder, C. B. Veilleux, and J. D. Miller, "A cyber-physical system to detect iot security threats of a smart home heterogeneous wireless sensor node," *IEEE Access*, vol. 8, pp. 205989–206002, 2020.

[38] M. Farooq and M. Hassan, "Iot smart homes security challenges and solution," *International Journal of Security and Networks*, vol. 16, no. 4, pp. 235–243, 2021.

[39] A. A. Zaidan, B. B. Zaidan, M. Qahtan, O. S. Albahri, A. S. Albahri, M. Alaa, F. M. Jumaah, M. Talal, K. L. Tan, W. Shir, *et al.*, "A survey on communication components for iot-based technologies in smart homes," *Telecommunication Systems*, vol. 69, pp. 1–25, 2018.

[40] B. L. R. Stojkoska and K. V. Trivodaliev, "A review of internet of things for smart home: Challenges and solutions," *Journal of cleaner production*, vol. 140, pp. 1454–1464, 2017.

[41] P. Sethi, S. R. Sarangi, *et al.*, "Internet of things: architectures, protocols, and applications," *Journal of electrical and computer engineering*, vol. 2017, 2017.

[42] W. Li, T. Yigitcanlar, I. Erol, and A. Liu, "Motivations, barriers and risks of smart home adoption: From systematic literature review to conceptual framework," *Energy Research & Social Science*, vol. 80, p. 102211, 2021.

[43] R. Hassan, F. Qamar, M. K. Hasan, A. H. M. Aman, and A. S. Ahmed, "Internet of things and its applications: A comprehensive survey," *Symmetry*, vol. 12, no. 10, p. 1674, 2020.

[44] T. Song, R. Li, B. Mei, J. Yu, X. Xing, and X. Cheng, "A privacy preserving communication protocol for iot applications in smart homes," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1844–1852, 2017.

[45] A. B. Ab Rahman and R. Azamuddin, "Comparison of internet of things (iot) data link protocols," tech. rep., Technical report, Washington University in St Louis, 2015.

[46] S. J. Danbatta and A. Varol, "Comparison of zigbee, z-wave, wi-fi, and bluetooth wireless technologies used in home automation," in *2019 7th International Symposium on Digital Forensics and Security (ISDFS)*, pp. 1–5, IEEE, 2019.

[47] G. A. Naidu and J. Kumar, "Wireless protocols: Wi-fi son, bluetooth, zigbee, z-wave, and wi-fi," in *Innovations in Electronics and Communication Engineering: Proceedings of the 7th ICIECE 2018*, pp. 229–239, Springer, 2019.

[48] R. Krejvc´i, O. Hujvnák, and M. vSvepevs, "Security survey of the iot wireless protocols," in *2017 25th Telecommunication Forum (TELFOR)*, pp. 1–4, IEEE, 2017.

[49] A. B. Varol, "Compilation of data link protocols: Bluetooth low energy (ble), zigbee and z-wave," in *2019 4th International Conference on Computer Science and Engineering (UBMK)*, pp. 85–90, IEEE, 2019.

[50] M. B. Yassein, W. Mardini, and T. Almasri, "Evaluation of security regarding z-wave wireless protocol," in *Proceedings of the Fourth International Conference on Engineering & MIS 2018*, pp. 1–8, 2018.

[51] K. Kim, K. Cho, J. Lim, Y. H. Jung, M. S. Sung, S. B. Kim, and H. K. Kim, "What's your protocol: Vulnerabilities and security threats related to z-wave protocol," *Pervasive and Mobile Computing*, vol. 66, p. 101211, 2020.

[52] T. Kim *et al.*, "A study of the z-wave protocol: implementing your own smart home gateway," in *2018 3rd International Conference on Computer and Communication Systems (ICCCS)*, pp. 411–415, IEEE, 2018.

[53] C. W. Badenhop, S. R. Graham, B. W. Ramsey, B. E. Mullins, and L. O. Mailloux, "The z-wave routing protocol and its security implications," *Computers & Security*, vol. 68, pp. 112–129, 2017.

[54] M. B. Yassein, W. Mardini, and A. Khalil, "Smart homes automation using z-wave protocol," in *2016 International Conference on Engineering & MIS (ICEMIS)*, pp. 1–6, IEEE, 2016.

[55] M. S. Stepanov, L. S. Poskotin, D. V. Shishkin, T. Timur, and A. R. Muzata, "The using of zigbee protocol to organize the" smart home" system for aged people," *T-Comm-* , vol. 15, no. 10, pp. 64–70, 2021.

[56] M. Gupta and S. Singh, "A survey on the zigbee protocol, it's security in internet of things (iot) and comparison of zigbee with bluetooth and wi-fi," in *Applications of artificial intelligence in engineering: proceedings of first global conference on artificial intelligence and applications (GCAIA 2020)*, pp. 473–482, Springer, 2021.

[57] A. Tomar, "Introduction to zigbee technology," *Global Technology Centre*, vol. 1, pp. 1–24, 2011.

[58] D. Gislason, *Zigbee wireless networking*. Newnes, 2008.

[59] S. Farahani, *ZigBee wireless networks and transceivers*. newnes, 2011.

[60] C. M. Ramya, M. Shanmugaraj, and R. Prabakaran, "Study on zigbee technology," in *2011 3rd international conference on electronics computer technology*, vol. 6, pp. 297–301, IEEE, 2011.

[61] C. J. Tan, W. K. Wong, C. T. Loh, and T. S. Min, "Wi-fi based smart home electrical appliances remote control for visually impaired person," in *2020 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, pp. 153–158, IEEE, 2020.

[62] E. Lamers, R. Dijksman, A. van der Vegt, M. Sarode, and C. de Laat, "Securing home wi-fi with wpa3 personal," in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–8, IEEE, 2021.

[63] X. Lei, G.-H. Tu, C.-Y. Li, T. Xie, and M. Zhang, "Secwir: Securing smart home iot communications via wi-fi routers with embedded intelligence," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, pp. 260–272, 2020.

[64] M. Bassoli, V. Bianchi, and I. De Munari, "A plug and play iot wi-fi smart home system for human monitoring," *Electronics*, vol. 7, no. 9, p. 200, 2018.

[65] S. Tozlu, M. Senel, W. Mao, and A. Keshavarzian, "Wi-fi enabled sensors for internet of things: A practical approach," *IEEE Communications Magazine*, vol. 50, no. 6, pp. 134–143, 2012.

[66] Y. Al Mtawa, H. Singh, A. Haque, and A. Refaey, "Smart home networks: Security perspective and ml-based ddos detection," in *2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 1–8, IEEE, 2020.

[67] T. Li, Z. Hong, W. Feng, L. Yu, and Z. Wen, "Ms-zerowall: Detecting zero-day multi-step attack in smart home using vae and hmm," *IEEE Transactions on Vehicular Technology*, 2024.

[68] S. I. Popoola, R. Ande, B. Adebisi, G. Gui, M. Hammoudeh, and O. Jogunola, "Federated deep learning for zero-day botnet attack detection in iot-edge devices," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3930–3944, 2021.

[69] J. Zhang, S. Liang, F. Ye, R. Q. Hu, and Y. Qian, "Towards detection of zero-day botnet attack in iot networks using federated learning," in *ICC 2023-IEEE International Conference on Communications*, pp. 7–12, IEEE, 2023.

[70] B. I. Hairab, M. S. Elsayed, A. D. Jurcut, and M. A. Azer, "Anomaly detection based on cnn and regularization techniques against zero-day attacks in iot networks," *IEEE Access*, vol. 10, pp. 98427–98440, 2022.

[71] V. Sharma, K. Lee, S. Kwon, J. Kim, H. Park, K. Yim, and S.-Y. Lee, "A consensus framework for reliability and mitigation of zero-day attacks in iot," *Security and Communication Networks*, vol. 2017, no. 1, p. 4749085, 2017.

[72] S. Reardon, M. D. Hssayeni, and I. Mahgoub, "Detection of zero-day attacks on iot," in *2024 International Conference on Smart Applications, Communications and Networking (SmartNets)*, pp. 1–5, IEEE, 2024.

[73] A. Rizzardi, S. Sicari, A. C. Porisini, *et al.*, "Nero: Neural algorithmic reasoning for zero-day attack detection in the iot: A hybrid approach," *Computers & Security*, vol. 142, p. 103898, 2024.

[74] S. Ramapatruni, S. N. Narayanan, S. Mittal, A. Joshi, and K. Joshi, "Anomaly detection models for smart home security," in *2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, pp. 19–24, IEEE, 2019.

[75] S. Ji, Z. Zhang, S. Ying, L. Wang, X. Zhao, and Y. Gao, "Kullback–leibler divergence metric learning," *IEEE transactions on cybernetics*, vol. 52, no. 4, pp. 2047–2058, 2020.

[76] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A.-R. Sadeghi, and S. Tarkoma, "Iot sentinel: Automated device-type identification for security enforcement in iot," in *2017 IEEE 37th international conference on distributed computing systems (ICDCS)*, pp. 2177–2184, IEEE, 2017.

[77] S. Yin, W. Zhang, Y. Feng, Y. Xiang, and Y. Liu, "Automatic iot device identification: a deep learning based approach using graphic traffic characteristics," *Telecommunication Systems*, vol. 83, no. 2, pp. 101–114, 2023.

[78] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-baiot—network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[79] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, *et al.*, "Understanding the mirai botnet," in *26th USENIX security symposium (USENIX Security 17)*, pp. 1093–1110, 2017.

[80] A. Marzano, D. Alexander, O. Fonseca, E. Fazzion, C. Hoepers, K. Steding-Jessen, M. H. Chaves, Í. Cunha, D. Guedes, and W. Meira, "The evolution of bashlite and mirai iot botnets," in *2018 IEEE Symposium on Computers and Communications (ISCC)*, pp. 00813–00818, IEEE, 2018.

[81] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Classifying iot devices in smart environments using network traffic characteristics," *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1745–1759, 2018.

[82] A. Hamza, H. H. Gharakheili, T. A. Benson, and V. Sivaraman, "Detecting volumetric attacks on lot devices via sdn-based monitoring of mud activity," in *Proceedings of the 2019 ACM Symposium on SDN Research*, pp. 36–48, 2019.

[83] A. Hamza, D. Ranathunga, H. H. Gharakheili, T. A. Benson, M. Roughan, and V. Sivaraman, "Verifying and monitoring iots network behavior using mud profiles," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 1, pp. 1–18, 2020.

[84] J. Ren, D. J. Dubois, D. Choffnes, A. M. Mandalari, R. Kolcun, and H. Haddadi, "Information exposure from consumer iot devices: A multidimensional, network-informed measurement approach," in *Proceedings of the Internet Measurement Conference*, pp. 267–279, 2019.

[85] N. Koroniotis, N. Moustafa, E. Sitnikova, and J. Slay, "Towards developing network forensic mechanism for botnet activities in the iot based on machine learning techniques," in *Mobile Networks and Management: 9th International Conference, MONAMI 2017, Melbourne, Australia, December 13-15, 2017, Proceedings 9*, pp. 30–44, Springer, 2018.

[86] A. Guerra-Manzanares, J. Medina-Galindo, H. Bahsi, and S. Nõmm, "Medbiot: Generation of an iot botnet dataset in a medium-sized iot network.," in *ICISSP*, pp. 207–218, 2020.

[87] A. Pashamokhtari, N. Okui, Y. Miyake, M. Nakahara, and H. H. Gharakheili, "Combining stochastic and deterministic modeling of ipfix records to infer connected iot devices in residential isp networks," *IEEE Internet of Things Journal*, vol. 10, no. 6, pp. 5128–5145, 2022.

[88] M. Erfani, F. Shoeleh, S. Dadkhah, B. Kaur, P. Xiong, S. Iqbal, S. Ray, and A. A. Ghorbani, "A feature exploration approach for iot attack type classification," in *2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pp. 582–588, IEEE, 2021.

[89] N. Moustafa, "A new distributed architecture for evaluating ai-based security systems at the edge: Network ton_iot datasets," *Sustainable Cities and Society*, vol. 72, p. 102994, 2021.

[90] M. T. Paracha, D. J. Dubois, N. Vallina-Rodriguez, and D. Choffnes, "Iotls: understanding tls usage in consumer iot devices," in *Proceedings of the 21st ACM Internet Measurement Conference*, pp. 165–178, 2021.

[91] I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, and E. Cambiaso, "Mqttset, a new dataset for machine learning techniques on mqtt," *Sensors*, vol. 20, no. 22, p. 6578, 2020.

[92] R. Kolcun, D. A. Popescu, V. Safronov, P. Yadav, A. M. Mandalari, R. Mortier, and H. Haddadi, "Revisiting iot device identification," *arXiv preprint arXiv:2107.07818*, 2021.

[93] H. Gandhi, M. Mehra, and V. Ribeiro, "Bond: Efficient and frugal dl model co-design for botnet detection on iot gateways," in *Proceedings of the First International Conference on AI-ML Systems*, pp. 1–7, 2021.

[94] D. Sikeridis, I. Papapanagiotou, and M. Devetsikiotis, "Blebeacon: A real-subject trial dataset from mobile bluetooth low energy beacons," *arXiv preprint arXiv:1802.08782*, 2018.

[95] D.-G. Akestoridis, M. Harishankar, M. Weber, and P. Tague, "Zigator: Analyzing the security of zigbee-enabled smart homes," in *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pp. 77–88, 2020.

[96] M. Anagnostopoulos, G. Spathoulas, B. Viaño, and J. Augusto-Gonzalez, "Tracing your smart-home devices conversations: A real world iot traffic data-set," *Sensors*, vol. 20, no. 22, p. 6600, 2020.

[97] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, and X. Bellekens, "Machine learning based iot intrusion detection system: An mqtt case study (mqtt-iot-ids2020 dataset)," in *International networking conference*, pp. 73–84, Springer, 2020.

[98] M. Malik, M. Dutta, J. Granjal, *et al.*, "Iot-sentry: A cross-layer-based intrusion detection system in standardized internet of things," *IEEE Sensors Journal*, vol. 21, no. 24, pp. 28066–28076, 2021.

[99] A. Alatram, L. F. Sikos, M. Johnstone, P. Szewczyk, and J. J. Kang, "Dos/ddos-mqtt-iot: A dataset for evaluating intrusions in iot networks using the mqtt protocol," *Computer Networks*, vol. 231, p. 109809, 2023.

[100] M. Al-Hawawreh, E. Sitnikova, and N. Aboutorab, "X-iiotid: A connectivity-agnostic and device-agnostic intrusion data set for industrial internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3962–3977, 2021.

[101] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-iiotset: A new comprehensive realistic cyber security dataset of iot and iiot applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40281–40306, 2022.

[102] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," *arXiv preprint arXiv:1802.09089*, 2018.

[103] S. Athanasoulias, F. Guasselli, N. Doulamis, A. Doulamis, N. Ipiotis, A. Katsari, L. Stankovic, and V. Stankovic, "the plegma dataset: Domestic appliance-level and aggregate electricity demand with metadata from greece," *Scientific Data*, vol. 11, no. 1, p. 376, 2024.

[104] S. Dadkhah, H. Mahdikhani, P. K. Danso, A. Zohourian, K. A. Truong, and A. A. Ghorbani, "Towards the development of a realistic multidimensional iot profiling dataset," in *2022 19th Annual International Conference on Privacy, Security & Trust (PST)*, pp. 1–11, IEEE, 2022.

[105] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "Ciciot2023: A real-time dataset and benchmark for large-scale attacks in iot environment," *Sensors*, vol. 23, no. 13, p. 5941, 2023.

[106] I. Ullah and Q. H. Mahmoud, "A scheme for generating a dataset for anomalous activity detection in iot networks," in *Canadian conference on artificial intelligence*, pp. 508–520, Springer, 2020.

[107] B.-C. C. C. (BCCC), "Bccc website page.," 2024. Available online at: `https://www.yorku.ca/research/bccc`.

[108] M. Shafi, A. H. Lashkari, and A. H. Roudsari, "Nlflowlyzer: Toward generating an intrusion detection dataset and intruders behavior profiling through network layer traffic analysis and pattern extraction," *Computers & Security*, p. 104160, 2024.

[109] M. Shafi, A. H. Lashkari, V. Rodriguez, and R. Nevo, "Toward generating a new cloud-based distributed denial of service (ddos) dataset and cloud intrusion traffic characterization," *Information*, vol. 15, no. 4, p. 195, 2024.

[110] B.-C. C. C. (BCCC), "Bccc github page.," 2024. Available online at: `https://github.com/ahlashkari/`.

Appendices

A Appendix 1

Table 6: Testbed devices.

ID	*Category*	*Type*	*Name*
1	**Sensors**	**Motion Sensor**	Zooz Z-Wave Plus 700 Series XS Open — Close Sensor ZSE41
2			Ecolink ZWave Plus Pet Immune PIR Motion Sensor
3			HomeSeer MS100+ G2 ZWave Plus Motion Sensor
4		**Door Sensor**	Zooz Z-Wave Plus 700 Series XS Tilt — Shock Sensor ZSE43
5			Aeotec Zwave Door / Window Sensor 7 Pro Open Close, Tilt, Dry Contact Input
6		**Water-Leak/ Freez Sensors**	Zooz Z-Wave Plus 700 Series XS Water Leak Sensor ZSE42
7			Ecolink ZWave Plus Water Sensor with Probe, Freeze Sensor
8			Aeotec ZWave 700 Water Sensor 7 Pro
9			Ring Alarm Flood and Freeze Sensor4SF1S8-0EN0
10		**Smoke Detectors**	First Alert Battery Powered Z-Wave Smoke Detector & Carbon Monoxide Alarm, Works with Ring Alarm Base Station, 2nd Generation
11			Ring Alarm Smoke and CO Listener
12		**Temperature Sensor**	Building 36 Temperature Sensor - Battery Powered
13		**Multi Sensors**	Aeotec ZWave TriSensor, Motion, Temperature, Light Intensity
14			Zooz Z-Wave Plus 700 Series 4-In-1 Sensor ZSE40
15			HomeSeer ZWave Flex Sensor, Temperature and Indicator Light Sensor
16			FIBARO Zwave Plus Multi Sensor for Motion, Temperature, Light
17	**Actuators**	**Smart Locks**	SCHLAGE BE469ZP CEN 619 Connect Smart Deadbolt with Alarm with Century Trim in Satin Nickel, Z-Wave Plus Enabled
18			Kwikset Home Connect 620 Keypad Connected Smart Lock with Z-Wave Technology Featuring SmartKey Security in Matte Black
19			ULTRALOQ U-Bolt Z-Wave Smart Lock with Door Sensor, 5-in-1 Keyless Entry Door Lock with Z-Wave Plus, Works with Ring Alarm, Smart Door Lock, Smart Lock Front Door, ANSI Grade 1 Certified
20			Alfred Touchscreen Keypad Pin + Bluetooth + Z-Wave (DB1-C-BL) Smart Door Lock

21		**Lamps**	Lightinginside Smart Candelabra LED Bulbs
22			Lightinginside Smart Candelabra LED Bulbs
23			Lightinginside Smart Candelabra LED Bulbs
24			Lightinginside Smart Candelabra LED Bulbs
25			Lightinginside Smart Candelabra LED Bulbs
26			Lightinginside Smart Candelabra LED Bulbs
27			Philips Hue White E26 LED Smart Bulb
28			Philips Hue White E26 LED Smart Bulb
29			Philips Hue White E26 LED Smart Bulb
30			Philips Hue White E26 LED Smart Bulb
31		**Smart Plugs**	ZOOZ Z-WAVE PLUS POWER SWITCH ZEN15 FOR HEAVY DUTY APPLIANCES
32		**Siren**	Aeotec Siren 6, Z-Wave Plus S2 Enabled Zwave Siren Safety Speaker, Wall-Mounted Sound & Light Security Intruder Zwave Alarm with Backup Battery, 110dB
33			Aeotec ZW162 Z-Wave Doorbell 6
34		**Smart Switches/Buttons**	FIBARO Zwave KeyFob, White
35			The Button, Z-Wave Scene Controller, Red - FIBFGPB-101-3
36		**Dimmer**	Honeywell Home Z-Wave Plug-in Smart Dimmer
37			Zwave Dimmer Plug 700 Series, Dimmable Smart plug Built-in Repeater Range Extender
38			Zwave Dimmer Plug 700 Series, Dimmable Smart plug Built-in Repeater Range Extender
39		**Fan**	
40		**Coffee Maker**	
41		**Boiler**	
42		**Toaster**	
43			New one Zwave plug N4013, Z-Wave Plug with Energy Monitor, 700 Series Z-Wave Outlet with Electricity Monitoring (774388156030)
44			New one Zwave plug N4013, Z-Wave Plug with Energy Monitor, 700 Series Z-Wave Outlet with Electricity Monitoring (774388156030)
45			New one Zwave plug N4013, Z-Wave Plug with Energy Monitor, 700 Series Z-Wave Outlet with Electricity Monitoring (774388156030)

46			New one Zwave plug N4013, Z-Wave Plug with Energy Monitor, 700 Series Z-Wave Outlet with Electricity Monitoring (774388156030)
47			New one Zwave plug N4013, Z-Wave Plug with Energy Monitor, 700 Series Z-Wave Outlet with Electricity Monitoring (774388156030)
48			New one Zwave plug N4013, Z-Wave Plug with Energy Monitor, 700 Series Z-Wave Outlet with Electricity Monitoring (774388156030)
49			New one Zwave plug N4013, Z-Wave Plug with Energy Monitor, 700 Series Z-Wave Outlet with Electricity Monitoring (774388156030)
50			New one Zwave plug N4013, Z-Wave Plug with Energy Monitor, 700 Series Z-Wave Outlet with Electricity Monitoring (774388156030)
51			New one Zwave plug N4013, Z-Wave Plug with Energy Monitor, 700 Series Z-Wave Outlet with Electricity Monitoring (774388156030)
52			New one Zwave plug N4013, Z-Wave Plug with Energy Monitor, 700 Series Z-Wave Outlet with Electricity Monitoring (774388156030)
53			New one Zwave plug N4013, Z-Wave Plug with Energy Monitor, 700 Series Z-Wave Outlet with Electricity Monitoring (774388156030)
54			New one Zwave plug N4013, Z-Wave Plug with Energy Monitor, 700 Series Z-Wave Outlet with Electricity Monitoring (774388156030)
55			New one Zwave plug N4013, Z-Wave Plug with Energy Monitor, 700 Series Z-Wave Outlet with Electricity Monitoring (774388156030)
56			New One Zwave Plug N4003, Zwave Outlet with Energy Monitoring, Z Wave Device,Z Wave Hub Required, Z-Wave Smart Plug Compatible with Hubitat, SmartThings, Vera, Wink, Fibaro, Homeseer, 2GIG
57			New One Zwave Plug N4003, Zwave Outlet with Energy Monitoring, Z Wave Device,Z Wave Hub Required, Z-Wave Smart Plug Compatible with Hubitat, SmartThings, Vera, Wink, Fibaro, Homeseer, 2GIG

58			New One Zwave Plug N4003, Zwave Outlet with Energy Monitoring, Z Wave Device,Z Wave Hub Required, Z-Wave Smart Plug Compatible with Hubitat, SmartThings, Vera, Wink, Fibaro, Homeseer, 2GIG
59			New One Zwave Plug N4003, Zwave Outlet with Energy Monitoring, Z Wave Device,Z Wave Hub Required, Z-Wave Smart Plug Compatible with Hubitat, SmartThings, Vera, Wink, Fibaro, Homeseer, 2GIG
60			New One Zwave Plug N4003, Zwave Outlet with Energy Monitoring, Z Wave Device,Z Wave Hub Required, Z-Wave Smart Plug Compatible with Hubitat, SmartThings, Vera, Wink, Fibaro, Homeseer, 2GIG
61			New One Zwave Plug N4003, Zwave Outlet with Energy Monitoring, Z Wave Device,Z Wave Hub Required, Z-Wave Smart Plug Compatible with Hubitat, SmartThings, Vera, Wink, Fibaro, Homeseer, 2GIG
62			New One Zwave Plug N4003, Zwave Outlet with Energy Monitoring, Z Wave Device,Z Wave Hub Required, Z-Wave Smart Plug Compatible with Hubitat, SmartThings, Vera, Wink, Fibaro, Homeseer, 2GIG
63			New One Zwave Plug N4003, Zwave Outlet with Energy Monitoring, Z Wave Device,Z Wave Hub Required, Z-Wave Smart Plug Compatible with Hubitat, SmartThings, Vera, Wink, Fibaro, Homeseer, 2GIG
64			New One Zwave Plug N4003, Zwave Outlet with Energy Monitoring, Z Wave Device,Z Wave Hub Required, Z-Wave Smart Plug Compatible with Hubitat, SmartThings, Vera, Wink, Fibaro, Homeseer, 2GIG
65			New One Zwave Plug N4003, Zwave Outlet with Energy Monitoring, Z Wave Device,Z Wave Hub Required, Z-Wave Smart Plug Compatible with Hubitat, SmartThings, Vera, Wink, Fibaro, Homeseer, 2GIG
66			New One Zwave Plug N4003, Zwave Outlet with Energy Monitoring, Z Wave Device,Z Wave Hub Required, Z-Wave Smart Plug Compatible with Hubitat, SmartThings, Vera, Wink, Fibaro, Homeseer, 2GIG
67			New One Zwave Plug N4003, Zwave Outlet with Energy Monitoring, Z Wave Device,Z Wave Hub Required, Z-Wave Smart Plug Compatible with Hubitat, SmartThings, Vera, Wink, Fibaro, Homeseer, 2GIG

68			New One Zwave Plug N4003, Zwave Outlet with Energy Monitoring, Z Wave Device,Z Wave Hub Required, Z-Wave Smart Plug Compatible with Hubitat, SmartThings, Vera, Wink, Fibaro, Homeseer, 2GIG
69			New One Zwave Plug N4003, Zwave Outlet with Energy Monitoring, Z Wave Device,Z Wave Hub Required, Z-Wave Smart Plug Compatible with Hubitat, SmartThings, Vera, Wink, Fibaro, Homeseer, 2GIG
70			New One Zwave Plug N4003, Zwave Outlet with Energy Monitoring, Z Wave Device,Z Wave Hub Required, Z-Wave Smart Plug Compatible with Hubitat, SmartThings, Vera, Wink, Fibaro, Homeseer, 2GIG
71			New One Zwave Plug N4003, Zwave Outlet with Energy Monitoring, Z Wave Device,Z Wave Hub Required, Z-Wave Smart Plug Compatible with Hubitat, SmartThings, Vera, Wink, Fibaro, Homeseer, 2GIG
72			New One Zwave Plug N4003, Zwave Outlet with Energy Monitoring, Z Wave Device,Z Wave Hub Required, Z-Wave Smart Plug Compatible with Hubitat, SmartThings, Vera, Wink, Fibaro, Homeseer, 2GIG
73			New One Zwave Plug N4003, Zwave Outlet with Energy Monitoring, Z Wave Device,Z Wave Hub Required, Z-Wave Smart Plug Compatible with Hubitat, SmartThings, Vera, Wink, Fibaro, Homeseer, 2GIG
74			New One Zwave Plug N4003, Zwave Outlet with Energy Monitoring, Z Wave Device,Z Wave Hub Required, Z-Wave Smart Plug Compatible with Hubitat, SmartThings, Vera, Wink, Fibaro, Homeseer, 2GIG
75		**Raspberry PI**	Raspberry P4B, 8GB RAM, #1
76			Raspberry Pi 4B, 8GB RAM, #2
77			Raspberry Pi 4B, 8GB RAM, #3
78			Raspberry Pi 4B, 8GB RAM, #4
79			Raspberry Pi 4B, 4GB RAM, #5
80			Raspberry Pi 4B, 4GB RAM, #6
81			Raspberry Pi 5, 8GB RAM, #7
82			Raspberry Pi 5, 8GB RAM, #8
83			Raspberry Pi 5, 8GB RAM, #9
84			Raspberry Pi 5, 8GB RAM, #10
85			Android Phone, Samsung S22 Ultra
86			IOS Phone, Iphone 12

	User Connection and Control	

87			Tablet, Lenovo Tablet
88			Windows Laptop
89			Windows Laptop
90			Linux Laptop
91			Linux Laptop
92			Linux Laptop
93			Windows PC
94			Windows PC
95		**HUB**	Z-wave.me, z-station hub
96		**RFID Reader (Sniffer tools for z-wave frequency)**	HackRF one
97			Flipper-zero
98			YARD Stick One - a sub-1 GHz wireless test tool controlled by your computer from Great Scott Gadgets
99			RTL-SDR Blog V3 R860 RTL2832U 1PPM TCXO HF Bias Tee SMA Software Defined Radio with Dipole Antenna Kit
100			Nooelec RTL-SDR v5 Bundle - NESDR SMArt HF/VHF/UHF (100kHz-1.75GHz) Software Defined Radio. Premium RTLSDR w/ 0.5PPM TCXO, SMA Input, Aluminum Enclosure & 3 Antennas. RTL2832U & R820T2-Based Radio
101		**Network Switch**	TP-Link Switch
102		**RPi Monitors**	10 inch
103			10 inch
104			10 inch
105			10 inch
106			7 inch
107			7 inch
108			7 inch
109			7 inch
110		**External Hard Disk (Share Folder)**	20 TB

B Appendix 2

Table 7: Attacks schedule.

Attack Category	*Attack Family*	*Target*	*Start time*	*End time*
Layer 4 D/DoS (TCP-based)	***TCP Flood - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/23/2024 16:30	9/2/2024 17:45
Layer 4 D/DoS (TCP-based)	***SYN Flood - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/23/2024 18:00	9/23/2024 22:05
Layer 4 D/DoS (TCP-based)	***CONNECTION - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/24/2024 9:30	9/24/2024 18:45
Layer7 D/DoS (HTTP-based)	***GET Flood - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/24/2024 20:00	9/24/2024 22:00
Layer7 D/DoS (HTTP-based)	***GET Flood - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/25/2024 9:30	9/25/2024 18:50
Layer7 D/DoS (HTTP-based)	***POST Flood - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/26/2024 9:30	9/26/2024 18:05
Layer7 D/DoS (HTTP-based)	***SLOW - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/26/2024 18:50	9/26/2024 22:10
Layer7 D/DoS (HTTP-based)	***BOT - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/27/2024 8:00	9/27/2024 9:40
Layer7 D/DoS (HTTP-based)	***RHEX - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/27/2024 10:00	9/27/2024 11:10
Layer7 D/DoS (HTTP-based)	***STRESS - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/27/2024 11:20	9/27/2024 12:50
Layer7 D/DoS (HTTP-based)	***Downloader- Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/27/2024 13:00	9/27/2024 17:40
Layer7 D/DoS (HTTP-based)	***HEAD - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/27/2024 17:45	9/27/2024 18:45
Layer7 D/DoS (HTTP-based)	***NULL - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/28/2024 12:00	9/28/2024 15:05
Layer7 D/DoS (HTTP-based)	***COOKIE - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/28/2024 15:15	9/28/2024 17:15
Layer7 D/DoS (HTTP-based)	***PPS - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/28/2024 17:25	9/28/2024 21:00
Layer7 D/DoS (HTTP-based)	***EVEN - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/28/2024 21:05	9/29/2024 1:35
Layer7 D/DoS (HTTP-based)	***APACHE - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/29/2024 11:15	9/29/2024 12:15
Layer7 D/DoS (HTTP-based)	***KILLER - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/29/2024 12:25	9/29/2024 14:05

Layer7 D/DoS (HTTP-based)	***TOR - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/29/2024 14:30	9/29/2024 15:50
Layer7 D/DoS (HTTP-based)	***GSB - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/29/2024 16:00	9/29/2024 17:05
Layer7 D/DoS (HTTP-based)	***DGB - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/29/2024 17:15	9/30/2024 6:00
Layer7 D/DoS (HTTP-based)	***CFB - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/30/2024 8:55	9/30/2024 17:35
Layer7 D/DoS (HTTP-based)	***BYPASS - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	9/30/2024 17:45	10/1/2024 7:00
Layer7 D/DoS (HTTP-based)	***STOMP - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/1/2024 8:15	10/1/2024 9:15
Layer7 D/DoS (HTTP-based)	***CFBUAM - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/1/2024 9:30	10/1/2024 11:20
Layer7 D/DoS (HTTP-based)	***DGB - Method-1***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/1/2024 17:05	10/2/2024 5:15
Layer 4 D/DoS (TCP-based)	***NULL TCP - Method-2***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/2/2024 11:00	10/2/2024 14:30
Layer 4 D/DoS (TCP-based)	***TCP Flood - Method-2***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/2/2024 14:45	10/2/2024 18:50
Layer7 D/DoS (HTTP-based)	***HTTP Flood - Method-2***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/2/2024 19:00	10/2/2024 20:35
Layer7 D/DoS (HTTP-based)	***STRESS - Method-3***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/2/2024 21:50	10/2/2024 23:30
Layer 4 D/DoS (TCP-based)	***TCP Flood - Method-3***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/3/2024 9:30	10/3/2024 10:35
Layer7 D/DoS (HTTP-based)	***AVB - Method-3***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/3/2024 11:15	10/3/2024 18:30
Layer 4 D/DoS (TCP-based)	***SYN-Flood-Method-5***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/4/2024 9:55	10/4/2024 10:40
Layer 4 D/DoS (TCP-based)	***ACK-Flood-Method-5***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/4/2024 10:50	10/4/2024 23:50
Layer 4 D/DoS (TCP-based)	***SYN-URG-Flood-Method-5***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/5/2024 12:50	10/5/2024 13:20
Botnet	***Hammer Botnet***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/5/2024 13:30	10/5/2024 23:30
Layer7 D/DoS (HTTP-based)	***Mixed Get-Post-Head***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/6/2024 12:30	10/8/2024 17:50
Brute Force	***BruteForce***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/8/2024 18:05	10/9/2024 1:05

Information Gathering	***All types***	Z-wave Hub (130.63.241.82)	10/9/2024 8:00	10/9/2024 16:00
Web Attacks	***Repaly Attack***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/10/2024 10:30	10/10/2024 21:15
Layer 4 D/DoS (TCP-based)	***SYN FLood - Method-4***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/10/2024 21:30	10/11/2024 8:15
Layer 4 D/DoS (TCP-based)	***ACK FLood - Method-4***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/11/2024 9:00	10/11/2024 10:00
Layer 4 D/DoS (TCP-based)	***CONNECTION - Method-4***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/11/2024 10:20	10/11/2024 11:00
Layer 4 D/DoS (TCP-based)	***SYN-PSH - Method-4***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/11/2024 11:15	10/11/2024 15:30
Layer 4 D/DoS (TCP-based)	***SYN-URG - Method-4***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/11/2024 15:50	10/11/2024 17:00
Layer 4 D/DoS (TCP-based)	***SYN-PSH-URG-Method-4***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/11/2024 17:10	10/11/2024 18:00
Layer7 D/DoS (HTTP-based)	***Apache-test***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/11/2024 20:00	10/11/2024 22:15
Layer 4 D/DoS (TCP-based)	***Slowloris - Method-4***	Z-wave Hub HTTP/S Service (http://130.63.241.82:8083)	10/11/2024 22:25	10/13/2024 22:40
Websocket Attack	***Malformed - Websocket***	Z-wave Hub Websocket Service (ws://130.63.241.82:8083)	10/14/2024 10:30	10/14/2024 11:35
Websocket Attack	***TBC - Websocket***	Z-wave Hub Websocket Service (ws://130.63.241.82:8083)	10/14/2024 12:25	10/14/2024 14:15
Websocket Attack	***Bruteforce - Websocket***	Z-wave Hub Websocket Service (ws://130.63.241.82:8083)	10/14/2024 17:15	10/15/2024 8:55
Websocket Attack	***Payload-encoding-Websocket***	Z-wave Hub Websocket Service (ws://130.63.241.82:8083)	10/15/2024 9:05	10/16/2024 2:15
MQTT Attack	***Unvalid-Publish-Flood***	MQTT Broker (130.63.241.102:1883)	10/16/2024 10:30	10/16/2024 17:20
MQTT Attack	***Unvalid-Subscribe-Flood***	MQTT Broker (130.63.241.102:1883)	10/16/2024 17:40	10/17/2024 10:45
MQTT Attack	***Valid-Publish-Flood***	MQTT Broker (130.63.241.102:1883)	10/17/2024 11:00	10/17/2024 16:50
MQTT Attack	***Publish-Subscribe-Flood***	MQTT Broker (130.63.241.102:1883)	10/17/2024 17:05	10/17/2024 23:50

MQTT Attack	***Wildecard-Subscribe-Flood***	MQTT Broker (130.63.241.102:1883)	10/18/2024 10:15	10/19/2024 15:15
MQTT Attack	***Bruteforce***	MQTT Broker (130.63.241.102:1883)	10/19/2024 17:00	10/19/2024 23:05
MQTT Attack	***Connect-Disconnet-Flood***	MQTT Broker (130.63.241.102:1883)	10/19/2024 23:15	10/21/2024 1:00
MQTT Attack	***Connect-Disconnet-Flood***	MQTT Broker (130.63.241.102:1883)	10/20/2024 22:55	10/21/2024 10:00
MQTT Attack	***Very Large Message***	MQTT Broker (130.63.241.102:1883)	10/20/2024 10:15	10/20/2024 16:15
MQTT Attack	***Encoded-Payload***	MQTT Broker (130.63.241.102:1883)	10/20/2024 16:45	10/20/2024 22:30
MQTT Attack	***PingReq-Flood (Short Keep Alive)***	MQTT Broker (130.63.241.102:1883)	10/21/2024 10:30	10/21/2024 22:35
MQTT Attack	***Slow Publish***	MQTT Broker (130.63.241.102:1883)	10/21/2024 22:55	10/24/2024 11:00
Z-wave Attack	***Replay/DoS***	From Plug (#95) to Z-wave Hub (#1)	10/25/2024 17:10	10/25/2024 20:40
Z-wave Attack	***Replay/DoS***	From Plugs (#81, #54) to Z-wave Hub (#1)	10/25/2024 20:43	10/25/2024 23:50
Z-wave Attack	***Replay/DoS***	From Hub (#1) to Z-wave Device (#26) - Door Lock	10/25/2024 23:57	10/26/2024 12:30
Z-wave Attack	***Replay/DoS***	From Door Sensor (#6) to Z-wave Hub (#1)	10/26/2024 12:45	10/26/2024 19:45
Z-wave Attack	***Replay/DoS***	From Hub (#1) to Z-wave Device (#27) - Plug lamp	10/26/2024 19:50	10/26/2024 23:59
Z-wave Attack	***Replay/DDoS***	From Hub (#1) to 30 different devices	10/27/2024 0:10	10/27/2024 13:30
Z-wave Attack	***Replay/DDoS***	From 40 different devices to From Hub (#1)	10/27/2024 15:00	10/27/2024 23:59
Z-wave Attack	***Jamming***	Using YARD Stick on 908.42 MHz	10/28/2024 0:10	10/28/2024 12:25
Z-wave Attack	***Jamming***	Using YARD Stick on 915 MHz	10/28/2024 12:50	10/28/2024 22:20

Z-wave Attack	***Jamming***	Using HackRF on 915 MHz	10/28/2024 22:30	10/29/2024 11:20
Z-wave Attack	***Jamming***	Using YARD Stick on 908.42 MHz, and HackRF on 915 MHz	10/29/2024 12:00	10/29/2024 22:00
Z-wave Attack	***Jamming***	Using YARD Stick on both 908.42 MHz and 915 MHz	10/29/2024 23:30	10/30/2024 17:15
Z-wave Attack	***Slow DDoS to Hub***	From 30 different devices to From Hub (#1)	10/30/2024 18:00	10/31/2024 20:00
Z-wave Attack	***Slow Ack Flood to Door Sensor***	From Hub (#1) to Z-wave Device (#6) - Door Sensor	10/31/2024 20:05	11/1/2024 21:15
Z-wave Attack	***Slow Ack Flood to plug lamp***	From Hub (#1) to Z-wave Device (#94) - Plug lamp	11/1/2024 21:35	11/2/2024 19:45
Z-wave Attack	***Slow open lock Flood to door lock***	From Hub (#1) to Z-wave Device (#64) - Door Lock	11/2/2024 20:00	11/3/2024 21:30
Z-wave Attack	***Slow Turn On to kitchen plug***	From Hub (#1) to Z-wave Device (#48) - Plug coffee maker	11/3/2024 22:20	11/4/2024 22:30
Z-wave Attack	***Motion detected Slow DoS to hub***	From Motion Sensor (#17) to Z-wave Hub (#1)	11/4/2024 23:00	11/5/2024 23:15
Z-wave Attack	***Fuzzing - Src ID fuzzing***	Source ID Fuzzing to Z-wave Hub (#1)	11/5/2024 23:45	11/6/2024 16:00
Z-wave Attack	***Fuzzing - Dst ID fuzzing***	Destination ID Fuzzing from Z-wave Hub (#1)	11/6/2024 16:10	11/7/2024 2:25
Z-wave Attack	***Fuzzing - Src ID and Dst ID fuzzing***	Source ID and Destination ID Fuzzing	11/7/2024 10:15	11/7/2024 20:30
Z-wave Attack	***Fuzzing - Payload fuzzing to hub***	Payload Data Fuzzing to Z-wave Hub (#1)	11/8/2024 8:30	11/8/2024 23:45
Z-wave Attack	***Fuzzing - all parameter fuzzing with correct CRC***	All parameter fuzzing with correct CRC	11/9/2024 0:15	11/9/2024 16:20

Z-wave Attack	***Fuzzing - all parameter fuzzing with incorrect CRC***	All parameter fuzzing with incorrect CRC	11/9/2024 21:30	11/11/2024 13:15

Vita

Candidate's full name: MohammadMoein Shafi

University attended (with dates and degrees obtained):
Bachelor of Computer Engineering at University of Tehran, 2017 - 2022

Publications:
MohammadMoein Shafi, Arash Habibi Lashkari, Arousha Haghighian Roudsari, **"NTLFlowLyzer: Towards generating an intrusion detection dataset and intruders behavior profiling through network and transport layers traffic analysis and pattern extraction"**, published in the Computers & Security journal (Elsevier).

MohammadMoein Shafi, Arash Habibi Lashkari, Hardhik Mohanty, **"Unveiling malicious DNS behavior profiling and generating benchmark dataset through application layer traffic analysis"**, published in the Computers and Electrical Engineering journal (Elsevier).

MohammadMoein Shafi, Arash Habibi Lashkari, Arousha Haghighian Roudsari, **"Toward Generating a Large Scale Intrusion Detection Dataset and Intruders Behavioral Profiling using Network and Transportation Layers Traffic Flow Analyzer (NTLFlowLyzer)"**, under the second revision at Journal of Network and Systems Management (Springer).

Arash Habibi Lashkari, MohammadMoein Shafi, Yongkun Li, Abhay Pratap Singh Sengar, Ashley Barkworth, **"Unveiling Evasive Malware Behavior: Towards Generating a Multi-Sources Benchmark Dataset and Evasive Malware Behavior Profiling Using Network Traffic and Memory Analysis"**, submitted to the Cybersecurity journal (Springer).