POLITECNICO DI TORINO

Master's Degree in Computer Engineering

Politecnico di Torino

Master's Degree Thesis

In collaboration with University of Calgary

Enhancing Security in Smart Buildings: Traffic Classification for Automated Access Control

Supervisors

Prof. Riccardo SISTO

Prof. Lorenzo DE CARLI

Prof. Fulvio VALENZA

Dott. Daniele BRINGHENTI

Candidate

Francesco ROSATI

Academic Year 2023 / 2024

Summary

The advent of the Internet of Things (IoT) has revolutionized the concept of Smart Buildings by integrating various smart sensing and control devices to improve efficiency and user experience. This thesis explores the realm of Smart Buildings, with a focus on the challenges of access control. In particular, the research investigates the applicability and effectiveness of machine learning techniques to classify device activities within the Smart Building environment.
As Smart Buildings continue to evolve, the need to ensure robust security measures becomes paramount. Traditional, intrinsically static access control methods often struggle to adapt to dynamic environments. This study explores the feasibility of using machine learning algorithms to dynamically classify device activities, with the aim of improving access control mechanisms. Research leverages existing datasets to assess the robustness and accuracy of machine learning traffic classification by analyzing the traffic patterns generated by IoT devices. Machine learning models are then applied to classify these patterns into specific activities.
By understanding and classifying different device activities, the system can dynamically adjust access permissions, contributing to a more adaptable and responsive security infrastructure.
Preliminary results demonstrate that this type of approach has considerable potential, indeed, it has been possible to correctly classify about 98% of packets in network traffic from several IoT devices.
The results of this research provide valuable insights into the field of IoT-enabled Smart Buildings, shedding light on the potential of machine learning to promote access control strategies. The implications of implementing such techniques go beyond security, impacting the overall functionality and sustainability of Smart Building environments.

Acknowledgements

More than five years have passed since that first day at Politecnico. Five years since my life changed radically. Today, this thesis is the result of continuous changes, hard work and sacrifices that have led me to become who I am. The Politecnico and university in general, is different from high school. Engineering is not easy at all and here, you are just a number. You must consider that throughout your academic journey, you will have to fight with your mind not to fall too low in moments of difficulty and not to fly too high when everything happens as you expected. Nevertheless, though, if you learn to approach your studies with the right attitude, you realize how everything around you, is full of growth opportunities ready to be leveraged.

I know that in these cases, the first thanks go to all my supervisors, but I have to say that without my parents, I wouldn't have even started this university journey. So, first of all, I want to thank my mom and dad, without whose support, both financially and morally, all of this wouldn't have been possible. Leaving home is never easy, especially if you are nineteen years old. Leaving the environment that raised you, your everyday friends and the habits that have always accompanied you. I have matured and from an extremely shy and introverted boy, I have gradually become a citizen of the world. I hope you are proud of me.
I want to thank all my supervisors for giving me the opportunity to study in Canada, for guiding and supporting me with their valuable, helpful advice and for giving me their precious feedback during these last months.
I want to thank all the professors I had during my journey for teaching me all the new things I know.
I would also like to thank, with great affection and care, the rest of my family and all the friends I have had during these years, in Latina, in Turin and in Calgary. Without you, this important part of my life wouldn't have been the same.
Lastly, I want to thank every other person who, in one way or another, has contributed either actively or passively to this amazing achievement.

I entered Poli as a boy and leave it as a man.

Table of Contents

List of Tables

List of Figures

Listings

Acronyms

AD

Anomaly Detection.

ALG

Application Layer Gateway.

BPF

Berkeley Packet Filters.

CPU

Central Processing Unit.

DBSCAN

Density-Based Spatial Clustering of Applications with Noise.

IDS

Intrusion Detection System.

IoT

Internet of Things.

IP

Internet Protocol.

IPS

Intrusion Prevention System.

LoRaWAN

Long Range Wide Area Network.

MUD

Manufacturer Usage Description.

NGFW

Next-Generation FireWall.

NumPy

Numerical Python.

OAS

OAuth-based Authorization Service.

OAuth

Open Authorization.

OSI

Open Systems Interconnection.

PCAP

Packet CAPture.

PIN

Personal Identification Number.

PyTZ

Python TimeZone.

QoS

Quality of Service.

RFID

Radio Frequency IDentification.

SDN

Software-Defined Networking.

SVM

Support Vector Machines.

TP

Twisted Pair.

UI

User Interface.

UTC

Universal Time Coordinated.

VEREFOO

VErified REFinement and Optimized Orchestration.

VPN

Virtual Private Network.

WeMo

Wireless Motion.

Wi-Fi

Wireless Fidelity.

Chapter 1

Introduction

In the increasingly connected digital era we live in, the importance of IoT (Internet of Things) devices is constantly growing. IoT devices indeed, represent a revolution in connectivity and automation. Thanks to their ability to monitor and collect environmental data, automate processes and control devices remotely, optimize resource usage and offer new personalized services, they are radically transforming the way humans interact with the world around them. Through data analysis, they enable valuable insights that drive smarter, more informed decisions. In summary, IoT devices promise to improve our everyday activities by making environments smarter, more efficient and more connected. Regarding the topic of smart environments, in addition to Smart Homes, Smart Buildings are emerging as fundamental components of our daily lives. Smart Buildings integrate a wide range of IoT devices to improve energy efficiency, optimize resources and provide a more comfortable and secure living or working experience to their occupants. However, with the increasing complexity and connectivity within Smart Buildings, new security challenges arise, particularly regarding the domain of security.

As Smart Buildings become more interconnected and complex, the need for robust Access Control management and IoT traffic protection becomes paramount. Access Control mechanisms indeed, if properly configured, ensure that only authorized users can access sensitive resources, data and services within the building's network. This is crucial in order to safeguard privacy, prevent unauthorized access and mitigate potential security breaches. Current approaches to Access Control management within Smart Buildings, while partially effective, are not without limitations. One of the main issues lies in the diversity and heterogeneity of IoT devices, each with its own communication protocols and security requirements. Traditional Access Control systems often struggle to adequately address this complexity, leading to vulnerabilities and potential security gaps. In addition, the dynamic nature of smart environments, where devices are constantly added, removed or reconfigured, poses

additional challenges to Access Control management. It is also worth noting that existing approaches focus primarily on Smart Homes rather than Smart Buildings, making it even more difficult to adapt these solutions to the wider and more complex ecosystem of Smart Buildings.

To address these challenges, new approaches leveraging machine learning-based techniques for traffic classification and Access Control management have emerged. By analyzing IoT traffic patterns and employing predictive models, these approaches offer more adaptive, intelligent and context-aware solutions for Access Control management within Smart Buildings. Machine learning algorithms can learn from historical data, predict device behaviors and dynamically adjust Access Control policies in order to ensure optimal security while maintaining convenience and flexibility for the user. The evaluation of these innovative approaches has provided promising results, demonstrating their effectiveness in enhancing security, optimizing resource utilization and improving the overall user experience within Smart Buildings. Therefore, by leveraging machine learning algorithms for traffic classification and Access Control management, Smart Buildings can achieve greater resilience, adaptability and efficiency in addressing evolving security threats and operational challenges.

In summary, the integration of machine learning-based techniques holds immense potential to revolutionize Access Control management within Smart Buildings. By addressing the limitations of current approaches and leveraging the power of predictive analytics, these solutions could pave the way for more efficient, secure and user-centric smart environments.

1.1 Thesis Objectives

The primary objective of this thesis is to conduct a comprehensive and in-depth analysis regarding the Access Control management within Smart Buildings, aiming to fully understand the crucial importance of this aspect in ensuring security and efficient management of resources, as well as in improving the overall experience of the occupants.

Another key objective is to carry out a comprehensive review of the existing literature in the field of Access Control management. Through this review, we aim to identify current trends, challenges and the best practices with the aim of developing an in-depth understanding of the context in which our research work fits.

In addition, this thesis aims to develop and propose new methodologies and approaches to enhance the effectiveness and the adaptability of Access Control policies within Smart Buildings. By using advanced techniques such as machine learning-based traffic classification, we intend to develop smarter, more efficient and dynamic Access Control policies capable of adapting in real-time to changing user needs and variations in the built environment. We also focus on exploring strategies for practical implementation of such policies, taking into account limitations and practical considerations.

A crucial aspect of our work is the experimental evaluation of our proposed methodologies. We aim to conduct in-depth experimental studies and simulations to evaluate the effectiveness and efficiency of our proposals, comparing them with traditional solutions and assessing their ability to handle realistic scenarios within Smart Buildings. This evaluation allows us to identify strengths and areas of improvement of the several analyzed approaches, as well as to provide an empirical basis for our conclusions.

Finally, this thesis aims to identify the remaining challenges and outline future research directions in the field of Access Control management within Smart Buildings, in order to contribute to the growth and development of increasingly effective solutions of this rapidly evolving field.

1.2 Thesis Outline

This section provides a brief overview of the discussed topics for each chapter of this thesis.

Chapter 2:

In the second chapter of this thesis, in addition to an introduction to the Smart Building concept and the crucial importance of Access Control in this context, a detailed analysis of the background and related work is conducted. Smart Buildings are presented as a new frontier in the integration of advanced technologies aimed at improving energy efficiency, security and comfort of the occupants. However, the implementation of these advanced technologies brings with it new challenges in terms of managing and securing access to devices, resources and services within the network. This review of existing literature in the context of Access Control within Smart Buildings provides an in-depth overview of the research and studies conducted in this field. Both traditional approaches and new innovative methodologies are explored in order to fully understand the state of the art and the remaining unresolved challenges. This literature review not only provides a solid knowledge

base but also identifies key areas that require further studies and development. In particular, the importance of developing efficient and adaptable Access Control policies to ensure security and effective management of resources within Smart Buildings is discussed, while enabling a better user experience.

Chapter 3:

The third chapter of this thesis explores the landscape of potential threats that could compromise the security of Smart Buildings and countermeasures to address them. Several risks are identified, including excessive access, privilege abuse, privacy violation and others, highlighting the complexity of security management in this technologically advanced context. Three main approaches for addressing these threats are thoroughly examined: the Brownfield approach, which focuses on managing existing vulnerabilities; the Proxying approach, which employs secure devices to filter malicious traffic and finally, Access Control, which regulates user interactions and permissions within the system. After a careful evaluation, the decision is made to focus on Access Control as the preferred approach, in order to improve Smart Building security, given its effectiveness in mitigating a wide range of threats and its ability to perfectly integrate without compromising the operational usability of the building. This choice is supported by a robust analysis of alternatives and is tailored to the specific needs of the research project. This chapter thus provides a solid foundation for addressing security challenges in the context of Smart Buildings, offering a comprehensive overview of threats and strategies to address them.

Chapter 4:

The fourth chapter of this thesis addresses the topic of Access Control within Smart Buildings, focusing on the importance of Access Control policies for the well-being and security of the occupants. Several strategies are explored to ensure the robust and reliable management of such policies. Specifically, the roles, entities and dynamics of Access Control policies are analyzed, considering both human users and device identities. Additionally, usage scenarios and concrete examples of Access Control policy implementation are discussed. Subsequently, different strategies to implement these policies are explored, including cloud-based authentication and authorization, Virtual Private Networks (VPNs), IP address filtering with MUD rules, machine learning-based traffic classification and other techniques. Finally, the possibility of embedding Access Control policies directly into the devices themselves is explored, thereby reducing dependence on external servers and simplifying the overall architecture.

Chapter 5:

The fifth chapter of this thesis presents two distinct approaches for traffic classification in IoT networks: PingPong and the Random Forest classifier. Through the analysis of two different datasets, Bhosale and PingPong, these approaches are tested to evaluate their effectiveness in detecting and classifying IoT device events. Each experiment comprises specific phases, such as data collection and preprocessing, model training and performance evaluation. The results obtained provide a solid basis for comparing the capabilities of both approaches and discussing their practical applications in the context of traffic management and security of IoT networks, with significant implications for the design and management of Smart Building systems.

Chapter 6:

In the sixth chapter of this thesis, the results obtained from the application of two IoT traffic classification approaches, PingPong and the Random Forest classifier, are examined. Through the analysis of two different datasets, Bhosale and PingPong, both approaches show high performance in detecting and classifying IoT device events. However, PingPong demonstrates greater versatility and adaptability across the datasets. The chapter then, provides an in-depth evaluation of the performance of the two classification methods and offers insights for further research in the field of Access Control policy management within Smart Buildings, thus contributing to outline future research directions.

Chapter 7:

The seventh chapter of this thesis examines the implications, limitations and future prospects of the work done. It analyzes the effectiveness of the two IoT traffic classification approaches used and discusses the importance of considering different factors when choosing the best approach for a given analysis context. Possible future directions for research are also outlined, focusing on optimizing Access Control policies based on the past behavior of devices and integrating these policies with device communication protocols in Smart Buildings.

Chapter 8:

The eighth chapter of this thesis is the concluding chapter. It emphasizes the innovative approach in managing Access Control policies within Smart Buildings through the use of machine learning-based traffic classification. The improvement in security and user experience is highlighted, thanks to the customization and responsiveness of Access Control policies. The importance of integration between

machine learning algorithms and device communication protocols is clearly outlined, along with the complexity of the traffic classification system. Finally, the chapter outlines a future vision in which Smart Buildings become adaptable, secure and advanced environments guided by dynamic and responsive Access Control policies.

Chapter 2

Background and Related Work

This Chapter provides an overview related to the basic concepts and a brief review of the literature in the field of Internet of Things. The "Background" Section (2.1), introduces the key theoretical concepts, essential to understand the context of this research. The "Related Work" Section (2.2), reviews the studies and developments that contributed to the current state of the arts.

2.1 Background

In the context of this thesis, the Internet of Things landscape proves to be of fundamental importance, especially in relation to the context of Smart Buildings and the problem of Access Control. Studying and delving into the different facets of these domains, the interconnected nature of IoT devices certainly becomes a central theme. In fact, my research aims to support the hypothesis that leveraging machine learning for traffic classification of IoT devices installed within a smart environment, brings significant benefits and innovation also and especially with regard to the security aspect. Indeed, this approach serves as a key element in enhancing various security measures, highlighting the essential link between Internet of Things advancements and the overall goals of strengthening Access Control within Smart Buildings.

2.1.1 IoT overview

The Internet of Things represents a technological paradigm that, through the use of smart devices such as sensors and other sophisticated tools, is based on the interaction and connection between physical objects. The main goal of IoT is to

create a large interconnected network in which physical devices can collect, process and share data with each other through the Internet infrastructure.

The Internet of Things is characterized by some key aspects:

- ***Intelligent Devices:*** Physical devices are considered "smart" as they are characterized by their computational capabilities and internet connectivity, enabling advanced functionalities. IoT devices then, are able to collect data from the environment in which they have been installed, making an analysis based on the information that has been received and then act accordingly.
- ***Connectivity:*** The several IoT devices communicate with each other through different communication protocols, including Wi-Fi, Bluetooth or Zigbee. Low-power communication protocols such as LoRaWAN can also be used. This allows efficient communication and transmission of data between all devices.
- ***Data Collection and Processing:*** IoT devices collect data from their surroundings according to the purpose each of them serves. This data then, can either be processed and analyzed by the device itself or sent to central servers for conducting more advanced analysis.
- ***Cloud Computing:*** It is possible to use cloud computing services for data storage, processing and analysis. This, allows large amounts of information to be managed, taking advantage of scalable services.
- ***Applications and Automation:*** The IoT paradigm is highly significant to a wide range of sectors and environments, such as Smart Homes, Smart Buildings, industry, health, agriculture, transportation and more. Practical IoT applications range from managing resources to responding to special situations that occur in the different areas where they are installed and from providing personalized services to making industrial processes more efficient.

Smart Thermostat:

An example of IoT device is the smart thermostat, such as the Nest thermostat (Figure 2.1). This device detects environmental conditions such as temperature and humidity and uses that data to automatically regulate the heating or cooling of a room. Users can also control the thermostat via a mobile application, enabling remote management of thermal comfort in their homes.

Smart Lighting:

Another example is the smart light bulb, like the Philips Hue bulb light (Figure 2.2). This type of bulb can be controlled through an IoT connection, allowing users

to adjust brightness, color and even set automatic lighting programs. This not only provides greater control over home lighting but can also contribute to energy efficiency.

This examples illustrate how IoT devices can be integrated into daily life to improve comfort, convenience and productivity in a variety of home environments and, in essence, underscore that Internet of Things has the enormous potential to transform everything around us, opening up new opportunities for innovation and automation of operations. However, at the same time, it includes many significant challenges related to data security, privacy and managing interoperability between devices.

Figure 2.1: Nest thermostat

Figure 2.2: Philips Hue bulb light

2.1.2 Smart Buildings

Delving into more detail, a Smart Building is a structure that maximizes efficiency, functionality and sustainability through the integration of advanced technologies, including IoT devices. These buildings use sensors, automation and data analysis to optimize energy consumption, improve safety and overall comfort. By incorporating IoT devices, such as temperature sensors and intelligent lighting, Intelligent Buildings gain real-time data collection and analysis, allowing for precise control of environmental factors. This interconnected system not only enables predictive maintenance, early problem identification and remote management but also enhances security measures. Thanks to IoT-powered cameras, access sensors and advanced analytics, Smart Buildings can proactively address security concerns, contributing to a more intelligent, secure and environmentally friendly living or working environment. The synergy between IoT and Smart Buildings results in a perfectly integrated infrastructure that prioritizes efficiency, sustainability, security and comfort for the occupants.

Figure 2.3: Smart Building

The most common features of a Smart Building include:

- ***Automation:*** There are automated systems for controlling lights, air conditioning, curtains or other devices. Automation can be programmed through well-defined rules or, it can adapt dynamically based on environmental conditions or inputs from the surroundings, such as a person entering a room or smoke detection following a fire.

- ***Energy Management:*** Advanced control and monitoring of energy consumption are made possible by energy management tools and systems. Indeed, these systems aim to optimize energy efficiency by reducing energy wastage and balancing consumption as needed.

- ***Predictive Maintenance:*** The use of specific sensors in the different areas of the building, allows for monitoring the status of the several systems. This results in preventive maintenance interventions to extend the lifespan of the equipment and thus, the reduction of operational costs.

- ***Advanced Security:*** IoT devices within a Smart Building, part of integrated security systems such as video surveillance, intrusion detection systems or electronically controlled access, help to improve security by analyzing data from multiple sensors.

Smart Buildings, therefore, are designed to provide all the occupants, an environment that is more energy efficient, environmentally sustainable and that can be adapted to as many needs as possible.

2.1.3 Smart Buildings threats

After introducing the definition and general characteristics of a Smart Building in Subsection 2.1.2, we will now discuss the related threats. Although in fact, Smart Building are facilities that embody innovation and advanced connectivity, security, especially cybersecurity, remains a significant concern and critical priority. Due to the interconnectivity among devices and systems, numerous cyberthreats must be taken into consideration. These threats, can potentially endanger sensitive data (e.g., patient data within a hospital), key functionalities of several systems and user privacy in general. Therefore, it is essential to analyze and understand these risks, in order to try to find different solutions to mitigate them.

Figure 2.4: Smart Building threats

The main critical issues and threats may concern both general aspects and aspects related to network management, communication and management of the different IoT devices.

In particular, the following can be highlighted:

- ***Energy Consumption:*** While Smart Buildings aim to reduce energy consumption, maintaining user comfort and system functionality is paramount.

Poor energy management may result in higher operational costs and environmental impacts. Thus, Smart Building designers must focus on advanced energy management systems and sustainable technologies to mitigate this challenge effectively.

- ***Network Traffic Management:*** Inefficient network traffic management can result in data congestion and potential security vulnerabilities. Poorly managed network traffic can disrupt communication between devices and systems, impacting the reliability and security of Smart Building operations. Implementing robust traffic management solutions is essential to ensure efficient data flow and data security within the building's network infrastructure.

- ***Quality of Service (QoS):*** Inconsistent QoS can lead to disruptions in services like video surveillance, Access Control and automation, potentially creating security vulnerabilities. Ensuring that QoS standards are achieved, is crucial for the reliable operation of critical Smart Building functions.

- ***Authentication and Authorization:*** Weak authentication and authorization mechanisms pose significant security risks to Smart Buildings. Inadequate user or device verification can lead to unauthorized access, potentially compromising sensitive data, control over critical systems and overall building security. Smart Building developers must prioritize robust authentication and authorization protocols to safeguard against unauthorized intrusions.

- ***Unsecured Communication Protocols:*** The use of unsecured communication protocols can leave communications vulnerable to attacks, allowing potential adversaries to intercept, manipulate or compromise data exchanged between devices and systems within the Smart Building. This creates significant risks to data privacy, physical building security and operational continuity.

- ***Diversity of Protocols:*** The diversity of protocols can cause incompatibility, hindering communication and creating operational inefficiencies. Selecting and implementing standardized protocols is essential to ensure effective communication among various building components.

- ***Heterogeneous Configuration:*** Integrating devices and systems from different manufacturers can result in configuration and interoperability challenges. These differences can complicate management and maintenance, potentially leading to compatibility issues and suboptimal performance. Smart Building planners must emphasize standardization and compatibility to mitigate these challenges effectively.

- ***Long-Term Software Deployment:*** The long-term sustainability of Smart Building software involves several intricacies. As technology evolves, maintaining and updating software to remain compatible and secure over extended periods can be complex. Without continuous updates and security patches, Smart Building systems may become vulnerable to cybersecurity threats. This challenge highlights the need for a robust software development strategy that considers long-term support, adaptability and security.
- ***Load Balancing:*** Traffic spikes or energy imbalances can overwhelm certain components, leading to performance degradation or system failures. Effective load balancing is essential to ensure consistent and reliable operation, as it distributes workloads evenly, preventing bottlenecks and maintaining optimal system performance.
- ***Bandwidth Management:*** The rapid proliferation of connected devices within Smart Buildings can strain network bandwidth. This strain can lead to congestion and latency, particularly affecting critical applications such as real-time surveillance and emergency communication. Managing bandwidth effectively is vital to ensure that the diverse data needs within a Smart Building are met without compromising the performance of essential services.

2.1.4 Traffic Analysis

Traffic analysis, in the context of computer networks, is a technique used to examine, analyze and understand the flow of data traveling through the network from one device to another (e.g., from device A to device B and / or vice versa). This type of analysis, can be conducted at different levels and can involve either a very general study of data traffic or a more detailed and meticulous one, carried out on the specific network packet.

Three different types of traffic analysis can be highlighted:

- ***Analysis of Traffic Volume:*** It involves monitoring the flow of data through the network, without examining the specific content of the packets. The goal is to assess the level of network congestion, identify traffic peaks and detect possible general anomalies.
- ***Metadata Analysis:*** It consists of analysis of metadata from various data packets without looking at their content. Indeed, metadata includes important information related to the packets' transmission (e.g., source and destination IP addresses, transmission times, protocols used etc.). This type of analysis can be very useful for identifying communication patterns, data flows and for detecting suspicious behavior within network flows.

- ***Content Analysis:*** It involves examining the actual content of several data packets. It can be useful for identifying the type of data being exchanged (e.g., images, audio, video, text, etc.). This type of analysis, however, requires clear and precise permissions, as reading the actual content of network packets can endanger users' privacy.

In general, traffic analysis is crucial for resource management, performance monitoring and maintaining security in computer networks. However, it is important to always act with respect and in accordance with the privacy of various users.

2.1.5 Action Identification

After introducing the general characteristics of the traffic analysis in Subsection 2.1.4, it is now possible to discuss the concept of action identification.
This concept, for instance in the context of the Internet of Things, can refer to the process of analyzing traffic generated by devices to understand specific actions that are either requested or, directly executed. When different devices communicate with a management platform or with each other, they generate, like all devices that are capable of connecting to the Internet, a stream of data that, when analyzed, provides information regarding the actions (e.g., device status updates, service requests or specific operations) that are or need to be taken.

Action identification can be useful for several purposes:

- ***Behavior Monitoring:*** It means understanding how various devices interact with each other and then monitoring the overall behavior of the IoT system.
- ***Security:*** By understanding and detecting different, suspicious or unauthorized behavior, it helps to ensure the security of the system, preventing and thus avoiding attacks or unauthorized manipulation and accesses.
- ***Issue Diagnosis:*** Action identification can help find and solve any problems or malfunctions in the IoT system.
- ***Resource Optimization:*** Comprehending device actions can be helpful for optimizing the use of network resources and the devices themselves.

In this thesis, traffic analysis and action identification play a key role in improving the security of Smart Buildings, particularly in addressing challenges related to Access Control policy management. The challenge lies in the encrypted nature of traffic generated by IoT devices, which hinders traditional inspection methods. However, machine learning is a powerful tool for classifying encrypted traffic based on characteristics derived from packet length, variance and other metadata.

It is important to note that machine learning does not decrypt packet content, but operates on computed features to classify traffic patterns associated with different IoT device activities. Furthermore, this classification aids in refining Access Control strategies, enabling a more nuanced decision-making process regarding device access within Smart Building networks. The synergy between traffic analysis, action identification and machine learning thus ensures, innovative and effective Access Control management, without compromising the privacy of encrypted data transmitted between the several IoT devices within the system.

2.2 Related Work

Automated Network Security Orchestration and Configuration

In their respective works, Bringhenti et al. [1] and Sisto et al. [2] contribute significantly to the field of automated network security. Bringhenti et al. introduce the VEREFOO (VErified REFinement and Optimized Orchestration) framework, addressing complex challenges in orchestrating security functions within virtual networks. Their focus on automated firewall configurations [3] sheds light on the evolving landscape of network security in dynamic environments. Meanwhile, Sisto et al. provide a comprehensive overview of automation in network security systems, emphasizing its pivotal role in enhancing overall security. Their work not only serves as a valuable reference but also outlines a roadmap for future research, identifying key challenges and potential directions for automating security configuration processes. Together, these contributions form a holistic perspective on advancing automated approaches for securing dynamic network environments.

Advancing Security in Smart Environments: Access Control, Authorization and Threat Modeling

Sikder et al. [4] introduce a sophisticated multi-user and multi-device access-aware system optimized for Smart Home environments. This work outlines the challenges of managing Smart Home access in shared environments, providing valuable insights for securing user interactions. Their construction emphasis on users' skills adds a layer of granularity to the control mechanisms used, increasing the overall level of security of Smart Homes.
Cirani et al. [5] contribute with an OAuth-based authorization service architecture, named IoT-OAS, shedding light on licensing mechanisms in IoT. Their work underscores the significance of secure Access Control, aligning with contemporary authorization best practices.
Ren et al. [6] propose a secure Smart Home authentication system using voice recording and the Internet, focusing on strengthening authentication, particularly

for remote Smart Home device access. The inclusion of voice recording enhances overall Smart Home security.
In a broader context, Ning et al. [7] conduct a comprehensive examination of cyber enterprise security within the Internet of Things. Their research provides valuable insights into complex security challenges and concepts that characterize the vast IoT landscape. By addressing cyber entity security, their work contributes to the understanding of the multiple security dimensions in IoT environments.
Furthermore, Valenza et al. [8] introduce a hybrid threat model designed for intelligent systems. This work provides a comprehensive approach to threat modeling, aimed at strengthening the security posture of smart environments. By combining different threat elements in a hybrid model, their research contributes to the development of robust security algorithms designed for intelligent systems.

Enhancing Cybersecurity in Smart Environments: Anomaly Detection and Packet-Level Signatures

Bringhenti et al. [9] explore the area of cybersecurity personalization within Smart Homes, introducing a threat model aimed at fortifying security measures. Alrashdi et al. [10] present AD-IoT, an anomaly detection system designed for identifying IoT cyberattacks in Smart Cities, enhancing our understanding of threat detection in complex environments. This work also highlights the importance of finding appropriate security models, in order to address the risks related to Smart Home environments and IoT ecosystems.
In addition, De Carli et al. [11] focus on detecting abnormal usage for IoT devices within homes, focusing on detection and response to unusual behaviors. These collective efforts greatly contribute to the development of ongoing initiatives aimed at strengthening the resilience of the IoT ecosystem against cyber threats.
Trimananda et al. [12] introduce packet-level signatures for Smart Home devices and help to understand security policies and measures at the network level. Their work, through the development of a framework called “PingPong”, focuses on developing signatures to identify, manage and respond to potential threats in Smart Homes. The emphasis on packet-level analysis enhances the granularity of threat detection, providing valuable insights for securing Smart Home networks.

Privacy Concerns and Security Solutions in Connected Environments

Privacy issues in Smart Homes are a significant focus, as explored by Acar et al. [13] in their study “Peek-a-Boo”. The research emphasizes the need to address privacy challenges, particularly in encrypted Smart Home environments. It highlights the delicate balance required between Smart Home features and user privacy protection, recognizing evolving privacy concerns in connected environments.

In the wider context of digital security, Taylor et al. [14] introduce AppScanner, an innovative tool designed for automatic fingerprinting and identification of smartphone applications from encrypted network traffic. AppScanner provides a reliable solution for real-time app identification, addressing critical security needs in the dynamic mobile application landscape.
In the realm of IoT security, Miettinen et al. [15] contribute to the field with IoT Sentinel, a system focused on automated device type identification and security application. Leveraging a sophisticated fingerprinting mechanism and machine learning classification, IoT Sentinel demonstrates its strengths in proactive vulnerability assessment and effective mitigation strategies.
These works collectively highlight the multifaceted nature of privacy issues in connected environments and show innovative security solutions that aim to address these challenges effectively.

Behavior Transparency and Control for Smart Home IoT Devices

O'Connor et al. [16] introduce HomeSnitch, which focuses on explicit behavior and controls for Smart Home IoT devices. Their work contributes to the development of tools for greater visibility and control of IoT devices in home environments. The emphasis on transparent behavior coincides with the growing importance and need of user-centric control over Smart Home devices, addressing concerns about device behaviors and potential privacy implications.

Advancements in Smart Building Security and Infrastructure

The research conducted by Wendzel et al. [17] focuses on the security implications of user interactions within Smart Buildings, offering strategies to strengthen the system. By emphasizing the crucial role of user involvement, Wendzel provides valuable insights that help to shape the overall security landscape. To complement this, Ciholas et al. [18] present a systematic review of the literature, providing a comprehensive overview of the current state of the art and exploring future directions for Smart Building security.
A significant contribution to the field comes from Xue et al. [19], who introduce the S^2Net framework. This framework, optimized for software-defined intelligent architecture networks, addresses the unique challenges posed by Smart Building environments. By incorporating modern techniques, it aims to improve the efficiency and security of Smart Building systems. In addition, Zangrandi et al. [20] delve into security aspects, specifically examining threat profiles associated with IoT devices, including behavioral protocols and algorithms provided directly by manufacturers. Both projects emphasize the evolving security landscape of Smart Buildings, highlighting the importance of flexibility and sensitive security measures.

Moving beyond theoretical frameworks, Hernández-Ramos et al. [21] introduce SAFIR, a practical secure access framework for IoT-enabled services. This framework places a strong emphasis on secure access mechanisms, contributing significantly to the development of systems that ensure the integrity and confidentiality of services within Smart Buildings.
In the broader context of Smart Building infrastructure, Verma et al. [22] conduct a comprehensive review of sensing, control and IoT infrastructure. Their work not only provides a detailed overview of the factors that shape the Smart Building ecosystem, but also offers valuable insights into the challenges and opportunities inherent in creating a resilient IoT infrastructure. Together, these studies weave a narrative of continuous progress and adaptation in the pursuit of security and the advancement of Smart Building technology.

Advanced Approaches in Traffic Analysis and Anomaly Detection

In the area of improving safety through traffic analysis and anomaly detection, researchers have made significant contributions. Hamza et al. [23] elaborate on the crucial role of profiles MUD for the security of IoT ecosystems. Their work emphasizes the development, validation and use of behavioral profiles, highlighting the importance of MUD data. Complementing this, Ranathunga et al. [24] offer tools designed to automate and validate MUD profiles, providing valuable improvements for security measures in IoT devices.
Shifting the focus to the context of Smart Buildings, Younus et al. [25] explore the complexities of software-defined web-enabled infrastructures. Their comprehensive analysis sheds light on architectural complexities, usability challenges and the broader network security landscape within IoT-enabled Smart Buildings. Understanding these issues is critical to improve stability and security in the evolving landscape of Smart Building infrastructure.
In the domain of Software-Defined Networking (SDN), Fayazbakhsh et al. [26] propose FlowTags, an extension designed to address the challenges posed by dynamic middlebox actions in network architectures. This innovative framework provides essential visibility into middlebox operations, enabling effective policy enforcement and integrity verification within SDN.
Addressing the human element in network profiling, Chuluundorj et al. [27] present a system that exploits user actions to improve network traffic analysis. Their focus on distinguishing between normal and abnormal network activities by monitoring user and application interactions shows the potential of a UI sensor to achieve high accuracy in classifying network traffic based on user-initiated actions. Together, these studies contribute to the continued evolution of advanced approaches in traffic analysis and anomaly detection that are critical to the security of modern network infrastructures.

Chapter 3

Threat Analysis and Mitigation Strategies

This Chapter presents the potential threats within the network of a Smart Building (Section 3.1), as well as possible approaches to mitigate those threats (Section 3.2), while considering the best approach.

3.1 Threat Model

This Chapter describes a threat model which illustrates the potential risks and vulnerabilities that Smart Buildings might face in terms of security. By analyzing and understanding these possible risks, it will be possible to develop effective strategies and solutions to mitigate and manage emerging security challenges in today's Smart Buildings.

Within a Smart Building, several pressing threats require careful consideration to sustain the integrity and security of the network environment. These are the main threats:

1. **Excessive Access:** Automatic assignment of full network access to new devices or users can pave the way for indiscriminate access and unauthorized intrusion. This underscores the need for precise measures to avoid scenarios in which privileges exceed needs.

2. **Privilege Abuse:** Legitimate users may attempt to make unauthorized changes to network settings or introduce unapproved devices. Such actions carry the risk of unintended consequences, ranging from compromising network integrity to exposing the system to potential security vulnerabilities.

Maintaining active monitoring and strict restrictions on user activities emerges as an imperative strategy to effectively mitigate this risk.

3. **Privacy Violation:** If communications between the user's application and the device are not sufficiently protected or the encryption used to protect communications is weak or compromised, there may be a risk of interception by third parties, who could decrypt or manipulate data traffic. The privacy breach could allow an attacker to access sensitive user information.

4. **Persistent Access:** The persistence of access granted to temporary guests or devices without prompt removal from the network accentuates the risk of sensitive information leakage or unauthorized activity, underscoring the critical need for efficient access revocation mechanisms.

5. **Man-in-the-Middle Attacks:** Communications between devices may be subject to "man-in-the-middle" attacks if they are not adequately protected. Such attacks may allow an attacker to intercept or alter communications.

6. **Firmware and Software Vulnerabilities:** Unpatched vulnerabilities in the firmware and software of connected devices may be exploited to gain unauthorized access to the network, emphasizing the importance of regular updates and security patches.

7. **Integration with External Systems:** Connections to external systems, such as cloud services or third-party networks instead, may introduce new threat vectors that require extended security measures.

As an example, we consider the context of a Smart Building system, where an employee with administrator privileges decides to integrate the system with a new cloud service, in order to optimize data management. However, due to an insufficient implementation of cryptography during the integration process, communications between the system and the cloud service become vulnerable to "man-in-the-middle" attacks. An attacker, exploiting this vulnerability, is able to intercept sensitive information transmitted between the system and the cloud service, thereby gaining unauthorized access to user data. In addition, exploiting his privileges, the employee also makes unauthorized changes to network security settings and introduces a new personal device without the necessary approval, configuring the system to allow for indiscriminate access. This action not only compromises the integrity of the network, but also creates a situation of privilege abuse.

Although this is just a simple example, it effectively highlights how each aspect within the network of a Smart Building is crucial for the proper and secure execution of routine activities.

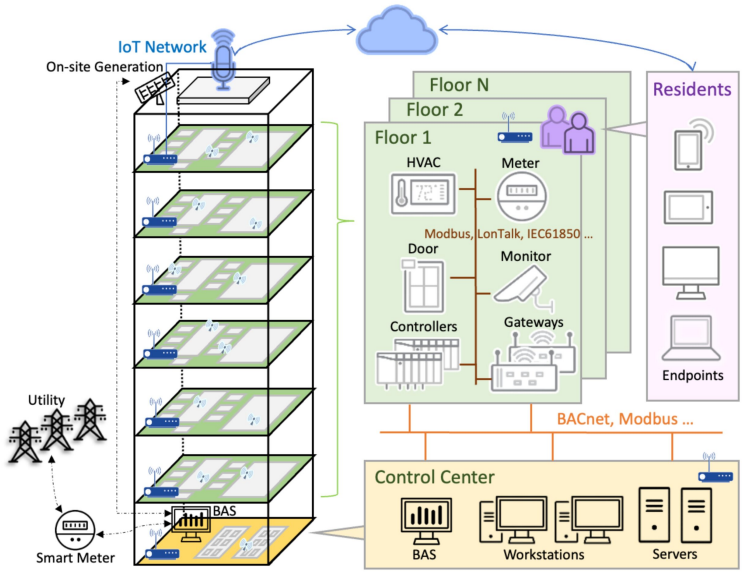

Figure 3.1: Example of a Smart Building Structure

3.2 Approaches for Threat Resolution

During the research work for this thesis, three main approaches were primarily analyzed: Brownfield approach (Subsection 3.2.1), Proxying (Subsection 3.2.2) and Access Control (Subsection 3.2.3). These three approaches differ from each other in terms of characteristics and implementation methods, but all three, prove to be crucial for the enhancement of network security within Smart Buildings.

3.2.1 Brownfield Approach

The Brownfield approach, thoroughly discussed within the paper authored by Miettinen et al. [15], is a strategy that focuses on managing security and vulnerabilities in an existing environment rather than creating a new system or infrastructure from scratch.

When considering the threats highlighted in our threat model (Section 3.1), the Brownfield approach proves to be a relevant solution, offering targeted strategies to address a multitude of security problems. This approach for instance, faces the threat "Excessive Access" directly through a comprehensive verification of existing vulnerabilities, identifying obsolete or unpatched devices. It also allows the isolation of such vulnerable devices within dedicated segments, preventing the spread of vulnerabilities within the network. In addition, the "Firmware and

Software Vulnerabilities" threat, which emphasize the importance of regular updates and patches, finds a natural resolution in the Brownfield approach. As part of the overall process, devices that can be updated are prioritized, ensuring that network components, operating systems, software applications and cryptographic solutions remain up-to-date. This type of solution minimizes the potential exploitation of vulnerabilities, helping to improve the overall security of the system. Finally, in order to mitigate the "Man-in-the-Middle" and "Privacy Violation" threats, continuous monitoring is essential. Recognizing both the importance and the vulnerability of communications among devices, the approach incorporates robust threat detection and active monitoring. This enables a quick identification of suspicious and malicious activities, facilitating timely and effective responses.

The general process typically includes:

1. **Audit of Existing Vulnerabilities:** Before introducing new devices or technologies, it is essential to conduct an audit of existing vulnerabilities within the environment. This may include identifying outdated or unpatchable devices, as well as existing threats.

2. **Isolation of Vulnerable Devices:** If there are legacy devices with known and unpatchable vulnerabilities, it is advisable to isolate them within separate network segments or dedicated subsystems. This practice separates IoT devices, such as sensors and automation equipment, from the main corporate network, limiting the possibility of vulnerabilities from these devices spreading throughout the network.

3. **Updates and Patching of Manageable Devices:** Devices that can be updated should be kept up to date. For instance, network components, operating systems, or software applications should be regularly updated to reduce vulnerabilities.

4. **Continuous Monitoring:** Threat detection and monitoring systems are implemented to identify suspicious activity or intrusions. Continuous monitoring is essential for identifying and responding promptly to threats.

The Brownfield approach is often the reality in many organizations, as it is not always possible to immediately replace all legacy devices. The key to security in this context is a strategic approach that balances the management of existing vulnerabilities with the implementation of new technologies and advanced security practices.

3.2.2 Proxying

Placing a secure device “upstream” of a vulnerable device to filter harmful traffic, is a good strategy to enhance security in an IoT environment or within a Smart Building. This secure device can act as a filter and firewall to protect the vulnerable device.

Taking into account the specific threats outlined in our threat model (Section 3.1), it becomes clear that Proxying offers several solutions for effectively mitigating some of them. To address the “Excessive Access” and “Privilege Abuse” threats, this type of approach proves to be crucial. By integrating a network firewall, this approach filters incoming and outgoing traffic, allowing only authorized access to reach the vulnerable device. In addition, to address the threat “Man-in-the-Middle attacks”, in which communications between devices are subject to interception or alteration, the Proxying approach introduces several types of devices, for instance, a server proxy. This intermediate device performs advanced traffic filtering, ensuring that only traffic which is considered safe is allowed through. This type of strategy not only protects against potential intrusions, but also establishes a secure communication channel between devices. Finally, the use of an Intrusion Detection System (IDS) and / or an Intrusion Prevention System (IPS), directly addresses the "Firmware and Software vulnerabilities" and "Privacy Violation" threats. Indeed, these systems, configured to detect and prevent intrusions or malicious attacks, contribute significantly to the vulnerable device protection, thus strengthening security through continuous monitoring and timely responses.

Several types of solutions can be considered:

- **Network Firewall:** A network firewall is a common device for filtering incoming and outgoing traffic. It can be configured to allow only authorized traffic to the vulnerable device and block everything else. One of the many features of the firewall for instance, is IP filtering. This capability is supported by the presence of an IP packet filter, a type of security device or software that operates at the network level and uses filtering rules to determine which data packets can pass through the device and which should be blocked. This type of device, can assess data packets based on criteria such as IP addresses, source and destination ports, protocols and other attributes.
- **Intrusion Detection System (IDS) / Intrusion Prevention System (IPS):** These systems can detect and prevent intrusions or malicious attacks. They can be configured to protect the vulnerable device from known threats.
- **Proxy Server:** A proxy server can act as an intermediary between the vulnerable device and the outside world. It can perform advanced traffic

filtering based on specific criteria, allowing only traffic considered safe.

- **Application Layer Gateway (ALG):** An ALG is specifically designed to monitor and control application-level traffic. It can be used to protect specific applications on vulnerable devices.
- **Next-Generation FireWall (NGFW):** These devices offer advanced features, including application-based filtering, advanced threat detection and security attack protection.

3.2.3 Access Control

Within a Smart Building, another important aspect can be considered in order to enhance its security: Access Control. This is a fundamental aspect for ensuring security and efficient resource management. It involves implementing security measures to regulate and manage the devices and users connected to the building's network. This ensures that only authorized devices and individuals can access and interact with the building's systems and data, enhancing overall security and preventing unauthorized access and potential cyber threats.

Considering our threat model (Section 3.1), this approach proves to be very effective in mitigating various threats. Thanks to strict identification and authentication protocols, for example, it is possible to mitigate the risk of "Excessive Access", since, only authorized entities are subject to a meticulous verification process, which ensures that privileges are granted only to those who need them, avoiding the risk of indiscriminate access and unauthorized intrusion. The threat "Privilege Abuse" is also effectively mitigated by this type of approach. Indeed, the definition of precise roles and privileges in the system ensures that legitimate users operate within predefined boundaries, reducing the risk of unauthorized changes to network settings or the introduction of unapproved devices. Moreover, with efficient access revocation mechanisms, the threat "Persistent Access" can be limited; in fact, this strategy ensures that access is promptly removed, minimizing the risk of sensitive information leakage or unauthorized activity. The Access Control approach also effectively counteracts the threat "Privacy Violation". Through the implementation of robust authentication processes for both users and devices, along with the utilization of secure encryption methods and ensuring their proper implementation, it guarantees the confidentiality of communications, preventing interception by third parties and thereby safeguarding users' sensitive information. Through continuous monitoring, timely updates and security patches, this approach strengthens the Smart Building network against potential exploits, ensuring that unpatched vulnerabilities are addressed quickly, thus mitigating the "Firmware and Software Vulnerabilities" threat. Regarding the "Integration with External Systems" threat, such as cloud

services or third-party networks, the Access Control approach offers the possibility of policy management within the devices themselves, thereby eliminating the need for external devices and with them, potential unwanted attacks. Such measures serve as a safeguard, protecting devices and systems from potential decryption or manipulation of data traffic.

There are several key aspects to consider regarding this topic:

Identification and Authentication:

Every user should be uniquely identified and authenticated before being allowed to access the Smart Building network. This can be done through the use of credentials such as passwords, PINs, RFID cards, fingerprints, or other biometric authentication methods.

Roles and Privileges:

The roles and privileges of users within the system should be clearly defined. For example, security personnel may have access to more areas than employees or visitors. This helps to limit unauthorized access.

Policy-Based Controls:

Specific policies should be used to define rules and criteria that determine who can access what and under what conditions. Policies should be flexible and customizable based on the specific needs of the building.

Logging:

Logging systems should be implemented to track all user activities. This allows for tracking authorized and unauthorized accesses and provides a starting point in the event of attacks or incidents.

Continuous Monitoring:

Access Control is not static; it should be continuously monitored and adapted to changing needs. Access management should be based on a continuous cycle of evaluation and improvement.

Centralized Management:

Management should be centralized in order to simplify system maintenance and administration. A centralized system also enables more efficient changes and

prompt responses to threats.

Updates and Patches:

Devices and software should be kept constantly up to date in order to address known vulnerabilities and improve security.

3.2.4 The Chosen Approach

Within this thesis, we decided to analyze the management of Access Control policies for improving security within Smart Buildings. This choice was made after a careful examination of the other two approaches: Brownfield and Proxying. Unlike the Brownfield approach, which addresses existing vulnerabilities within legacy systems and Proxying, which involves installing a secure device to filter legitimate from malicious traffic, Access Control policies are more concerned with regulating user interactions and permissions within the system. These policies indeed can serve as a strategic response to threats identified within the network, especially regarding concerns related to excessive access, privilege abuse and unauthorized persistence in the system. Thus, by using robust Access Control measures, it is possible to effectively mitigate these types of vulnerabilities. Furthermore, the user-friendly implementation of these policies ensures that security enhancements do not compromise the operational usability of the building, thereby improving the overall efficiency, security and well-being of both occupants and the infrastructure. In conclusion, although each approach to security brings its own strengths, the benefits, accuracy and adaptability of Access Control policies make them an optimal choice for dealing with a variety of threats in the dynamic context of Smart Buildings.

Chapter 4

Smart Building Access Control: Understanding Dynamics and Implementing Robust Policies

This Chapter will emphasize the importance of Access Control policies within Smart Buildings for the well-being and security of its occupants (Section 4.1). It will then explore potential strategies to ensure the robust and reliable management of various Access Control policies. (Section 4.2).

4.1 Access Control Roles, Entities and Policies Dynamics Explored

In this section, in addition to understanding why Access Control policies are crucial within Smart Building scenarios, various types of roles that can be encountered and different situations that may occur within a Smart Building will be presented, starting with a more general definition (Human Users and Device Identities) and progressively delving into more detail (Subsections 4.1.1 and 4.1.2), concluding with a concrete example of policy enforcement (Subsection 4.1.3).

In the context of Smart Building and Access Control policies, it is essential to categorize users to consider access and determine which entities or devices have access to the systems. Within a Smart Building, devices transcend their role as simple entities to access and become active participants in the system, similar to

users. This category includes a number of sensor devices, cameras, smart locks, light bulbs and thermostats, each of them, perfectly integrated into the infrastructure.

Users, who include both people and devices, interact with the system, seeking access to designated resources, services, or devices. System administrators, employees, visitors and other authorized entities fall into this category of users. Managing different Access Control policies becomes a significant aspect of Smart Building governance.

Within the Smart Building network system, two main identities contribute to the dynamic ecosystem:

Human Users:

By representing individuals as system administrators, employees, visitors and other authorized people, Human users interact with the network to access resources, data, or services. Their roles may include managing network configurations, requesting data, or controlling networked devices. By adhering to defined network access policies, human users use secure authentication strategies, ensuring that only authorized people access the Smart Building network.

Device Identities:

This type of identity actively participates in the network, which includes intelligent devices integrated into the network, such as sensors, cameras, smart locks, light bulbs, thermostats and other technological devices, like computers, smartphones or tablets. They exchange data, receive commands and interact with other network entities, contributing to the overall functionality. Operating within the network framework, Device identities require careful management through network and packet-level Access Control policies, in order to ensure secure and efficient interactions within the Smart Building network.

4.1.1 Example of Access Control Use Case

In this subsection, as previously mentioned, we will delve further into detail by describing more specifically the potential roles and entities that might be found, the scenarios that may occur and the types of policies that could be enforced within a Smart Building.

User Roles:

- **System Administrator:** The person with the ultimate control over the Access Control system, able to create or remove accounts, define rules and permissions.
- **Regular Users:** Occupants or users of the Smart Building who require access to different resources, devices or services. Regular Users may include employees, security staff or maintenance staff.
- **Visitors:** Individuals who are not regular occupants but need temporary access to specific devices or resources within the Smart Building. Visitors may include clients, guests, or service personnel.
- **Device Identities:** Smart devices or individuals who interact with the Smart Building system network through personal computers, smartphones, or tablets. These users, classified as Device Identities, leverage these devices not merely as tools but as active participants within the system. The devices themselves are considered users, enabling individuals to access and control various resources, devices, or services within the Smart Building environment. Their roles may vary from controlling smart devices to managing preferences and settings through dedicated applications or interfaces.

Entities or Devices:

- **Sensors:** Devices used to detect the presence of people or activities in certain areas.
- **Surveillance Cameras:** Used to monitor activities and detect potential threats or breaches.
- **Smart Devices (such as light bulbs, thermostats, etc.):** May have access restrictions based on user preferences or requirements.

Access Policies:

- **Access Hours:** Definition of specific times when users can access certain resources and other devices.
- **Role-Based Permissions:** Users may have different permissions based on their role (e.g., a standard employee vs. a manager).
- **Time-of-Day Access:** Access permissions vary depending on the time of day, allowing for different security levels during business hours, evenings, or weekends.

- **Emergency Access:** Overrides normal Access Controls during emergency situations.
- **Visitor Access:** Policies governing temporary access for visitors, including time limitations and restricted services.
- **Geofencing:** Access is granted or restricted based on the physical location of a user or device within the building network.
- **Contextual Access:** Access is determined by contextual factors like the purpose of the visit, the user's task, or the current state of the building (e.g., emergency situations).
- **Conditional Access:** Access is granted or denied based on specific conditions, such as the presence of multiple authorized users or the status of connected devices.

Usage Scenarios:

- **Virtual Access:** Controlling access to systems, devices or data through computer networks.

Access Conditions:

- **System Administrator:**
  - **Network Control:** The system administrator has full access and can control all devices connected to the Smart Building network, managing thermostats, light bulbs, audio / video systems and environmental sensors.
  - **Authorization Configuration:** They can configure network access for specific users or devices, define access rules and monitor user activities.
- **Regular Users (Employees):**
  - **Personal Network Control:** Occupants have access only to devices connected to the network within their designated areas. For instance, they can adjust the temperature in their offices or control the lights in their rooms.
  - **Limited Network Access:** They do not have access to critical network devices or sensitive network areas, such as server controls or conference room equipment.

- **Regular Users (Security Staff):**
  - **Access to Surveillance Cameras:** Security personnel can access surveillance cameras to monitor activities inside and around the building.
  - **Access Control:** They may have permissions to control network access to certain areas in response to emergency situations.
- **Regular Users (Maintenance Staff):**
  - **Access to Environmental Sensors:** Maintenance staff have access to environmental sensors connected to the network, in order to monitor air quality and other environmental parameters.
  - **Limited Access to Critical Network Areas:** They may have access to network-connected devices like thermostats and lights but could be restricted in more sensitive network zones.
  - **Access to Devices Under Maintenance:** They may have access to some non-critical network devices that are under maintenance, but only during the period of the maintenance.
- **Visitors:**
  - **Limited Network Access:** Guests may have access only to common-use network devices, such as charging stations or audio systems in public areas.
  - **Temporary Network Access:** Access to certain network devices may be granted only for a limited period during their stay.
- **Device Identities:**
  - **Network Interaction:** Smart devices or individuals using personal computers, smartphones, or tablets can interact with and control devices connected to the Smart Building network. For instance, managing preferences, settings, or accessing smart devices through dedicated applications or interfaces.

4.1.2 Example of Real Access Control Scenarios

We will now outline more detailed network-wide scenarios for a surveillance camera with Access Control functionality, in relation to the use case described in Subsection 4.1.1. In this example, there are three distinct users who want to interact with a surveillance camera, which, in turn communicates with an external server in order to manage Access Control permissions.

In each of these scenarios, network-wide authorization management involves user authentication, sending of valid access tokens and controlling permissible actions based on different types of user's permission levels. In every scenario, encryption can be used to protect data transmission and ensure user privacy.

Regular User (Alice):

Alice is a regular and authorized user who has full access to the surveillance camera. When Alice wants to view the video stream of the camera via the mobile app, the following steps may occur:

1. Alice opens the mobile application and authenticates herself by entering her username and password.
2. The application sends a camera access request to the manufacturer's cloud server.
3. The server checks if the access policy is valid. Since Alice has the necessary authorization, the server authenticates Alice's credentials and sends a valid access token.
4. The Alice mobile app sends a video stream request to the surveillance camera.
5. The surveillance camera, authenticating the access token, responds by sending the video stream to Alice.

An example illustrating this policy can be found in Listing 4.1.

Regular User with limitations (Bob):

Bob is a regular user with limited access who can only view video stream with time restrictions (from 14:00 to 16:00 and from 18:30 to 21:15), but cannot control and configure surveillance camera's settings. When Bob accesses the mobile application to view the video stream within his time constraints and change the settings of the camera, the following steps may occur:

1. Bob authenticates himself in the mobile app.
2. The application sends surveillance camera access and control requests to the cloud server.
3. The server checks if the access policy is valid. Since Bob has only a limited access to the camera, Bob's credentials are authenticated and a limited access token is sent.

4. Bob's video stream request is sent to the surveillance camera.

5. The surveillance camera, authenticating the limited access token, responds by sending the video stream to Bob.

6. Bob, wanting to adjust the camera's settings, attempts to modify configuration parameters through the mobile app.

7. The application sends a request to change settings to the cloud server.

8. The server, recognizing Bob's limited access, denies the request to modify camera's settings.

9. Bob receives a notification informing him that he doesn't have the necessary permissions to change camera's settings.

An example illustrating this policy can be found in Listing 4.2.

Visitor (Eve):

Eve is a visitor trying to view the video stream of the camera, without having any type of authorization. When Eve tries to connect, the following steps may occur:

1. Eve tries to authenticate himself in the mobile app.

2. The application sends an access request to the cloud server.

3. The server checks if the access policy is valid. Since Eve is a visitor and doesn't have the necessary authorization to access surveillance camera's functionalities, Eve's credentials are not authenticated and the access is denied.

4. Eve receives an error message and does not get access to the video stream or other features.

An example illustrating this policy can be found in Listing 4.3.

4.1.3 Demonstration of Access Control Policy Enforcement

We will now present a concrete demonstration of Access Control policy enforcement, linked to the example previously illustrated within Subsection 4.1.2. The following definitions, roles, device types, actions and conditions, do not cover all the possible scenarios within Smart Building systems. The ones provided here are meant to give just an idea of how one or more policies could be constructed.

Definitions:

- **UID (User ID):** A random string of 20 lowercase characters associated with each human user within the system.
- **DID (Device ID):** A random string of 20 lowercase characters associated with each device within the system.
- **RID (Role ID):** Numeric identifier associated with each role within the system.

Roles:

1. **System Administrator:** RID: 1
2. **Regular Users:** RID: 2
3. **Visitors:** RID: 3

Device Types:

- `<device_type_camera>` = `device_type == 'camera'`
- `<device_type_thermostat>` = `device_type == 'thermostat'`
- `<device_type_light>` = `device_type == 'light'`
- `<device_type_smartphone>` = `device_type == 'smartphone'`
- `<device_type_tablet>` = `device_type == 'tablet'`
- `<device_type_computer>` = `device_type == 'computer'`

Actions:

1. connect
2. exchange
3. access
4. control
5. authenticate

Conditions:

1. **Business Hours:** business_hours = access allowed only during business hours
2. **Night Hours:** night_hours = access allowed only during night hours
3. **Custom Hours:** start + end = access allowed only during the hours specified in the condition clause (ex. "start": "8:00", "end": "9:00")
4. **Weekdays:** weekdays = access allowed only on weekdays
5. **Location-Based Conditions:** access allowed only in specific locations of the Smart Building (ex. conference_room)
6. **Priority-Based Conditions:** access is based only on specific priority conditions (ex. fire_emergency)
7. **Device Status Conditions:** access is based only on device status conditions (ex. online)

Operators:

1. AND
2. OR
3. NOT

Effects:

1. allow
2. deny

Example of Policies:

```
{
    "rule": "regular_user_camera_access",
    "constraints": [
      { "source": { "UID": "ilrmvqm1tres6l8ot8u7", "RID": 2
    } },
      AND,
      { "action": "access" },
      AND,
      { "target": { "DID": "303xp7mcdn4m5k4ytqiq", "type": "
    camera", "device_status": "online" } }
    ],
    "effect": "allow"
}
```

Listing 4.1: Example of Alice's Access Control Policy

```
{
    "rule": "regular_user_camera_access",
    "constraints": [
      { "source": { "UID": "2j4mvjabhjsl6l8i3jm1", "RID": 2 }
     },
      AND,
      { "action": "access" },
      AND,
      { "time": [
          { "start": "14:00", "end": "16:00" },
          OR,
          { "start": "18:30", "end": "21:15" }
       ]
      },
      AND,
      { "target": { "type": "camera", "device_status": "
    online" } },
    ],
    "effect": "allow",

    OR,

    "rule": "regular_user_camera_control",
    "constraints": [
      { "source": { "UID": "ype0skeaexmmqbju6ymw", "RID": 2 }
     },
      AND,
```

```
      { "action": "control" },
      AND,
      { "target": { "DID": "303xp7mcdn4m5k4ytqiq", "type": "
    camera", "device_status": "online" } },
    ],
    "effect": "deny"
}
```

Listing 4.2: Example of Bob's Access Control Policy

```
{
    "rule": "visitor_camera_access",
    "constraints": [
      { "source": { "UID": "eyz9sn5jr8y0k8hnrg3c", "RID": 3 }
     },
      AND,
      { "action": "access" },
      AND,
      { "target": { "DID": "303xp7mcdn4m5k4ytqiq", "type": "
    camera", "device_status": "online" } },
    ],
    "effect": "deny"
}
```

Listing 4.3: Example of Eve's Access Control Policy

4.2 Strategies to Implement Access Control Policies

OpenFlow:

OpenFlow is a network communication protocol that enables centralized control and management of network devices, such as switches and routers, in a Software-Defined Networking (SDN) environment. With OpenFlow, network administrators can dynamically manage and configure network traffic flows, making it more flexible and adaptable to changing network needs. Finally, the flexibility given by OpenFlow allows the implementation and the adaptation of Access Control policies in real time, enabling more precise control over network communication.

Cloud Authentication and Authorization:

If the IoT device relies on a cloud service for management and control, authentication and authorization can be managed at the cloud server level. Each user would have

an account with corresponding permissions and requests from the devices would be managed according to those permissions. This centralized approach has positive consequences on the management of Access Control policies, indeed, it simplifies user authorization management and allows the regulation of access to IoT devices based on specific permissions assigned to each user account.

VPNs and Secure Connections:

The use of VPNs and secure connections can help ensure that communications between the IoT device and the management server are protected. In this way, authorization could be managed through secure authentication protocols. In the context of Access Control policies, the secure communication facilitated by the use of VPNs adds an additional layer of security to the authorization process. The encrypted nature of VPN connections in fact, protects against unauthorized access and data breaches, thus making an important contribution to the overall Access Control management strategy.

Token of Authorization:

Sending authorization tokens along with device requests may be one way to manage authorizations. The tokens could be generated and managed at the server level and validated by the device before specific actions are performed. This strategy enables more granular management of Access Control policies. Indeed, tokens, generated and managed at the server level, enable the validation of device requests only if they are associated to authorized tokens, thereby facilitating the immediate exclusion of unauthorized requests.

User Recognition at Application Level:

Some IoT devices may implement user recognition at the application level. For example, a mobile app associated with a device may require user authentication. This way, only authorized users can send commands or access certain features through the application. This type of strategy allows access and actions to be restricted to only authorized users, thus contributing to efficient and more high-level management of Access Control policies.

IP Filtering with MUD

IP Filtering with MUD (Manufacturer Usage Description) rules is a network security technique that allows network administrators to control and manage the network traffic based on the specific behavior and requirements of connected IoT devices, for instance restricting incoming and outgoing connections based on specific IP

addresses and ports associated with authorized users. MUD rules are created by device manufacturers and describe the intended communication and access patterns for their devices. These rules are implemented in network devices, such as routers and firewalls, to enforce policies that permit or restrict network access for IoT devices according to their intended usage. This approach then, allows specific Access Control policies to be defined for each device based on the MUD rules provided by the manufacturers.

Layer 4 Filtering (Vulnerability Protection):

Layer 4 filtering is a network security measure that focuses on safeguarding networks against known vulnerabilities and threats. In this context, filtering occurs at the transport layer of the OSI model, which allows for precise control over data transmission based on attributes like source and destination ports. By implementing Layer 4 filtering techniques, organizations can enhance their network security and optimize performance by blocking or allowing specific types of traffic. In this way it is possible to precisely control traffic based on source and destination ports, contributing to the definition of specific and targeted Access Control policies.

Machine Learning-Based Traffic Classification:

Traffic classification in Smart Buildings involves the use of machine learning algorithms to analyze and classify network activities generated by several devices. By then classifying devices based on their traffic patterns, it becomes easier to modify and improve Access Control policies, ensuring that only authorized devices interact with specific areas or systems. This dynamic approach not only improves security by identifying potential threats, but also contributes to efficient network management.

Device Tagging:

Device tagging for Access Control in Smart Buildings involves assigning specific labels or attributes to devices connected to the network. These labels can include information about the device type, owner, location and security requirements. By categorizing and tagging devices, it becomes easier to enforce Access Control policies, monitor network activity and ensure that each device is compliant to security and standards. This, could help in maintaining a secure and well-managed network environment within the building.

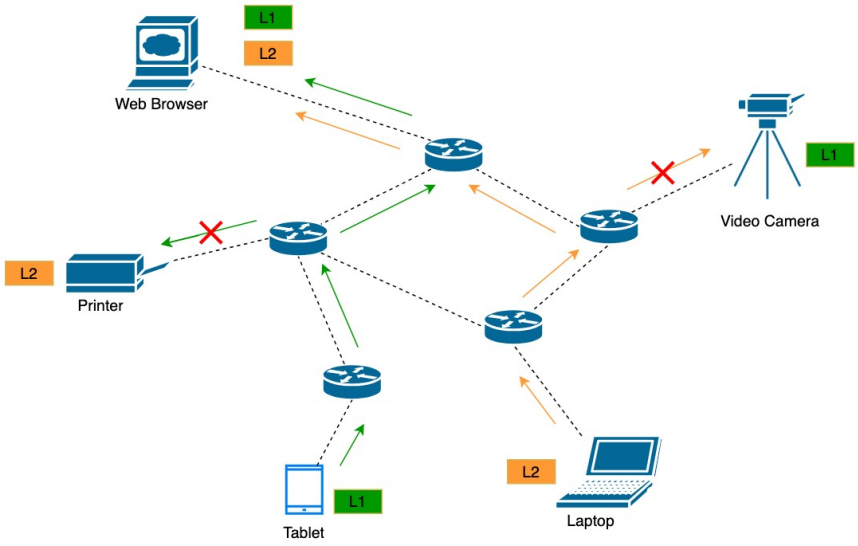

Figure 4.1: Device Tagging

4.2.1 Incorporating Access Control Policies Within the Device

Another strategy to implement Access Control policies could be incorporating them within one or more devices. Indeed, it may be necessary to implement less advanced Access Control policies that relies on systems integrated in the device itself, without the use of external servers or third-party dependencies.

The implementation then, could be managed as follows:

1. **Local Authentication:** Implement a local authentication system on the device itself. Users must authenticate directly to the device using specific credentials (e.g., username and password).
2. **Assign Permission Levels:** The device assigns specific permission levels to each authenticated user. For example, there may be a user with full access and a user with limited access only to specific device functionalities.
3. **Package Labeling:** Each packet of data generated or received by the device is labeled with the associated user identifier and its permission level. This label is added to the package metadata.

4. **Local Permission Control:** Before performing a specific action (for example, sending a video stream to a mobile application or changing device settings), the device verifies the package label and associated authorization level locally. Only actions allowed for the authenticated user are performed.

5. **Local Encryption:** Implement local encryption to protect communication between the device and user applications. Encryption ensures security when transmitting data within the network.

6. **Local User Management:** The device manages the list of authorized users and their credentials locally. Administrators can add, remove, or edit users directly through the device interface.

This approach reduces the dependency on external servers, allowing the device to operate independently within the network. However, it is important to note that while this solution may simplify the architecture, it requires effective local management of security and device manufacturer permissions. In addition, the implementation should be carefully designed to mitigate risks such as unauthorized access and firmware vulnerability.

Chapter 5

Machine Learning-Based Traffic Classification: Implementations and Experiments

This Chapter will outline the main features of the implementations and various experiments conducted regarding traffic classification of different IoT devices through machine learning. In particular, Section 5.1 will give a brief overview of the literature review in this field, Sections 5.2 and 5.3 will describe the two implementations (PingPong and Random Forest classifier), Section 5.4 will detail the two datasets used for the experiments while Section 5.5 will illustrate the experiments conducted on these datasets.

Additionally, the main modules and functions that are part of the code of the Random Forest classifier, will be described within Appendix A.

5.1 Literature Review

For the experimental section of this thesis, we found the machine learning-based traffic classification to be of significant interest as a strategy for implementing Access Control policies (for the reasons previously discussed within Section 4.2 of Chapter 4). Machine learning indeed, provides a dynamic and adaptive approach to Traffic Analysis, enabling the real-time identification of patterns and anomalies. This capability not only has the potential to improve the accuracy of device classification, but also allows the system to promptly respond to emerging security

threats. Furthermore, by integrating machine learning with Access Control policy management, we expect a more resilient and agile system that can adapt to the many different network dynamics within Smart Buildings, ensuring both robust security and simplified network management.

As the initial research strategy for the experimental part, we planned to get two to three different implementations presented in literature ([10], [11], [12], [13], [16]). Subsequently, we wanted to test them on at least two common datasets and finally make comparisons and considerations on the various approaches, in terms of accuracy, precision, recall and F1-score of the classification. Unfortunately, this was not possible, as, apart from the implementation outlined in [12], which is public on their research website, all the other implementations could not be obtained.

As a consequence, since [12] provided both a good implementation with innovative features and a diverse well-structured dataset, we decided to implement a generic Random Forest Classifier on Python and then, test both implementations on two different datasets: the PingPong dataset and the one used within [11] (Bhosale dataset). The choice to test both implementations on two different datasets is driven by the need to evaluate the generalizability and performance of the classifiers across different data sources. This strategy indeed, allows for testing and validating the ability of the two approaches to adapt to different contexts and data distributions, thus evaluating their potential for use in real-world scenarios, such as those found within Smart Buildings.

5.2 PingPong

PingPong [12] is an advanced system for the analysis of IoT traffic and represents a significant innovation in the field of cybersecurity and the detection of Internet of Things device events. This system is developed to handle the increasing challenges linked to the analysis of encrypted traffic generated by proprietary protocols, additionally offering an automated approach in order to extract packet-level signatures from IoT devices. This implementation, detects previously unexplored packet-level signatures and introduces an automated approach to extract them from different training datasets. This enables the creation of generalized models, applicable to a wide range of devices and events.

A series of different experiments demonstrate the effectiveness of this approach. These experiments include the analysis of traffic traces from devices such as WeMo Insight smart plug and Blink camera. PingPong, by leveraging its novel method focused on events “fingerprints”, demonstrates its capability to address challenges

such as signature evolution, variability due to specific configurations and encrypted data analysis. Finally, it implements defenses against possible passive confusion attacks (such as addition of dummy bytes), ensuring the robustness of traffic analysis and therefore provides a practical approach to the identification and understanding of the events of IoT devices.

Key Features of the Implementation:

1. **Identification of Package Level Signatures, Clustering and Traffic Analysis:**
   - PingPong employs spatial clustering algorithms like DBSCAN to identify signatures at the packet level within traces of IoT device traffic, enabling the grouping of packets into clusters associated with specific event signatures. It also uses both exact and relaxed matching techniques to compare packet signatures and conducts in-depth traffic analysis to detect distinct patterns and recurring behaviors.
   - Signatures represent unique sequences of packets that correspond to specific events of IoT devices, such as turning a light bulb on or off.
   - This approach enables more detailed traffic analysis, allowing the identification and differentiation of specific events within the overall packet flow.
2. **Extraction Methodology:**
   - The implementation is based on an automatic approach for signature extraction. This allows the creation of a generalized model that can be applied to new devices or similar events.
   - PingPong is designed to deal with encrypted data or data generated by proprietary protocols, allowing for wide coverage in traffic analysis.
3. **Identification of Signature Variations and Evolutions:**
   - PingPong can identify changes in signatures due to firmware updates or different device configurations. This ability to recognize and adapt to variations in signatures makes this implementation a robust system over time.
4. **Passive Defense Analysis:**
   - By addressing possible passive defenses, PingPong implements strategies such as adding dummy bytes to packets or intentionally delaying packet transmission. This helps to maintain the effectiveness of the classification even when there are attempts to obfuscate specific traffic characteristics.

5.3 Random Forest Classifier

This implementation aims to analyze and classify network flows captured from Internet of Things devices. The primary goal is to develop a classification model that can accurately categorize different events or behaviors exhibited by these devices based on network traffic data. The analysis involves processing packet captures (.PCAP files) and associated .timestamps files (every file includes a series of timestamps, each identifying the start of an associated device event, such as the startup of a smart camera) to extract features, which are then used to train a Random Forest classifier.

Working Environment and Used Libraries:

For the implementation of this classifier, we decided to use Python, primarily because it is a simple language to use but also because it offers a vast ecosystem of libraries and frameworks, specifically created for machine learning and data analysis. In this case, libraries such as Scikit-learn, Numpy and Pandas were used, which provide robust tools for building and deploying machine learning models efficiently. These libraries not only simplify the development process but also provide powerful functionalities, such as data manipulation, preprocessing and model evaluation. In addition, this programming language also offers several libraries useful for reading, filtering and analyzing .PCAP files, with Scapy being the foremost among them.

The main characteristics of the used libraries are:

- **PyTZ (version 2023.4):** PyTZ is a Python library for working with time zones. It provides functionality for converting between different time zones, handling daylight saving time adjustments and locating datetime objects. Since the data within both datasets were collected in countries with a different time zone compared to Italy (Los Angeles for the PingPong dataset and UTC - 1 for the Bhosale dataset), this library was useful for aligning all timestamps within the .timestamps files to the same time zone of the timestamps within the analyzed packets.

- **NumPy (version 1.26.3):** NumPy is a very important package for numerical computation in Python. It provides support for multidimensional arrays, along with a collection of mathematical functions for operating on these arrays efficiently. NumPy is widely used for numerical operations and data manipulation. We used this library to build the NumPy feature matrix and the NumPy array of labels, input parameters for the Random Forest algorithm of the Scikit-learn library.

- **Pandas (version 2.2.0):** Pandas is a powerful data manipulation and analysis library for Python. It offers data structures such as Series and DataFrame, which make it easy to work with structured data. Pandas offers functionality for reading and writing data from various file formats, cleaning, reshaping and aggregating data. We used this library to efficiently construct the data structure containing the features computed on the various traffic flows.

- **Scikit-learn (version 1.4.0):** Scikit-learn is a comprehensive machine learning library for Python. It provides implementations of various machine learning algorithms, including classification, regression, clustering and dimensionality reduction. Scikit-learn also provides tools for model selection, evaluation and preprocessing. We used this library to instantiate, train, test and compute the performance of the Random Forest classifier in terms of accuracy, precision, recall and F1-score.

- **Scapy (version 2.5.0):** Scapy is a powerful interactive packet manipulation library for Python. It allows users to create, decode and analyze low-level network packets. Scapy supports a wide range of protocols and can be used for tasks such as network testing, network discovery and security evaluation. We used this library to read and filter packets contained within the several analyzed .PCAP files.

Type of Classifier:

To obtain the best machine learning-based traffic classification performance we opted for a Random Forest Classifier, which proves to be particularly effective in classifying IoT device traffic for several reasons. First of all, IoT devices typically generate data with many characteristics, reflecting the different types of traffic they produce. Random Forests classifiers, as also demonstrated by Bhosale et al. in [11], excel at handling this high-dimensionality data, eliminating the need for complex feature selection or dimensionality reduction techniques. In addition, IoT traffic patterns, often have complex and non-linear relationships between features and classes. Leveraging decision trees, this type of classifier effectively models and classifies complex traffic patterns commonly encountered in IoT environments.

Finally, Random Forest classifiers offer scalability and efficiency, making them very useful for processing large volumes of IoT traffic data. Different tasks can be parallelized across multiple CPU cores and their computational complexity remains manageable even for big datasets. This ensures their applicability in real IoT environments where scalability and efficiency are critical. In conclusion, as highlighted by O'Connor et al. in [16], the robustness, the ability to capture complex relationships, the resistance to unbalanced data, the feature importance

analysis and the scalability of the Random Forest classifier make it an excellent choice for effectively classifying IoT device traffic.

5.3.1 Key Features of the Implementation

This subsection outlines the key features of this implementation, focusing on detailing the methodology employed for feature extraction from packet flows.

This implementation was designed specifically for the analysis of two different types of datasets. The first type of dataset (e.g., Bhosale dataset) comprises a unified set of data that needs to be randomly divided into training and test sets, typically in proportions such as 75% for training and 25% for testing. The second type of dataset (e.g., PingPong dataset), already consists of two distinct sets of data, each prepared for classifier training and testing purposes.

Methodology of Feature Extraction:

The classifier in this implementation categorizes various packet flows by considering the features that can be derived from the packets themselves, in this case, packet lengths. Packet lengths indeed, can provide valuable information about the network traffic patterns and behaviors, as highlighted by Taylor et al. in [14].

These are the main distinctive features that can be derived for precise and comprehensive classification:

- **Average Length:** This feature calculates the average length of packets within a specific time window. It can indicate the typical size of packets transmitted by the device during specific events.
- **Variance:** The variance measures the dispersion of packet lengths around the mean. A higher variance suggests greater variability in packet sizes, which may indicate different types of traffic patterns or behaviors.
- **Standard deviation:** Standard deviation quantifies the dispersion of packet lengths from the mean. It provides insights into the consistency or variability of packet sizes within the traffic data.
- **Minimum and maximum length:** These features detect the range of packet sizes observed during a particular event. They can reveal outliers or anomalies in packet length and it may be indicative of specific events or behaviors.
- **Percentiles:** Percentiles (e.g., 70th, 80th and 90th percentiles) divide the packet length distribution into segments, revealing the distribution of packet

sizes within the data. For instance, the 70th percentile represents the value below which 70% of packet lengths fall. Similarly, the 80th and 90th percentiles represent the values below which 80% and 90% of the packet lengths fall, respectively.

- **Skewness and Kurtosis:** Skewness measures the asimmetry of the packet length distribution, indicating whether the distribution is skewed towards shorter or longer packet lengths. Kurtosis measures the shape of the distribution, providing insights into the concentration of packet lengths around the mean.

Furthermore, it is possible to extract numerous other features such as additional percentiles or the median absolute deviation, to make the characteristics of each packet flow increasingly specific. Our Random Forest classifier is then able to classify different types of device events based on their corresponding network traffic patterns.

5.4 Datasets

The two considered implementations, have been tested on two different datasets: PingPong [12] and Bhosale [11]. Both with similar characteristics, i.e., they consist of traffic traces (.PCAP files) belonging to different devices and for different events, accompanied by a series of timestamps (.timestamps files), each identifying the beginning of a series of packets, related to a specific event to be analyzed. Additionally, data within the two datasets, represent real-world scenarios including traffic generated by devices that use encrypted communication or proprietary protocols. The following Subsections 5.4.1 and 5.4.2, explain the main characteristics of the two datasets.

5.4.1 Bhosale Dataset

The Bhosale dataset consists of several network traffic traces generated by various IoT devices within a Smart Home environment. It proves to be crucial as it is useful for training end testing the classifier in the experiment conducted by Bhosale et al. in [11].

Key Features of the Dataset:

1. **Composition:**
    - The dataset includes many IoT devices that can be found within a Smart Home environment, for instance a smart tv, a smart camera or smart

speakers. Each device has a diverse set of traffic traces that can be used as a rich source of training set for the classifier.

2. **Training and Test Sets:**

   - The Bhosale dataset doesn't consist of distinct sets of data specifically designed for training and testing the classifier, unlike the PingPong dataset. Instead, it contains a unified set of data that must be proportionally and randomly divided in order to derive training and test sets (we used 75% of the data for training and 25% for testing).

3. **Device Events:**

   - The dataset includes traces corresponding to several device events (further information about the type of devices and the analyzed events are provided within Section 6.1 of Chapter 6, precisely Table 6.2), such as turning on / off or playing videos with a smart tv, pronouncing a specific word in front of smart speakers or capturing video streams with the smart camera.
   - Each trace is associated with a specific device event, which facilitates the labeling process performed by the classifier.

4. **Timestamps:**

   - Timestamps are included within the dataset, providing temporal information about when each device event was started. This temporal data is useful to analyze, filter and divide packet flows based on temporal criteria.

In conclusion, the Bhosale dataset collects a large amount of data related to various events from different IoT devices and proves to be an excellent resource for carrying out experiments with our two implementations, in relation to traffic classification through machine learning.

5.4.2 PingPong Dataset

The PingPong dataset consists of numerous network traffic traces generated by various IoT devices in a Smart Home environment. This dataset is a key component within the PingPong research project, as it serves as training and evaluation data for the PingPong implementation, concerning the creation of packet-level signatures and the analysis of device behaviors.

Key Features of the Dataset:

1. **Composition:**

   - The dataset includes several IoT devices commonly found in Smart Homes, including smart plugs, light bulbs and cameras. Each type of device, contributes to the richness and variety of the dataset.
   - The dataset is organized into two main categories: "standalone" and "smarthome", providing separate scenarios for the evaluation of PingPong's capabilities.

2. **Standalone Category:**

   - The "standalone" category involves individual devices operating independently. It enables the analysis of device-specific behaviors without the influence of interactions with other Smart Home devices. Data of this category is exclusively used to train the classifier.

3. **Smart Home Category:**

   - The "smarthome" category captures network traffic in a more realistic network environment, where multiple IoT devices coexist and interact. This setting introduces complexities associated with simultaneous device tasks, providing a holistic view of real Smart Home network dynamics. Data of this category is exclusively used to test the classifier.

4. **Device Events:**

   - The dataset includes traces corresponding to several device events (further information about the type of devices and the analyzed events are provided within Section 6.1 of Chapter 6, precisely Table 6.1), such as turning on / off smart plugs, increasing / decreasing the intensity of smart lights or capturing video streams with smart cameras.
   - Each trace is associated with a specific device event, allowing the training and evaluation of PingPong's ability to identify and classify these events based on packet-level signatures.

5. **Timestamps:**

   - Timestamps are included within the dataset, providing temporal information about when each device event was started. This temporal data is useful to analyze, filter and divide packet flows based on temporal criteria.

In summary, the PingPong dataset is an important collection of network traffic data that reflects the complexity of IoT device communication in Smart Homes and, it serves as the basis for the machine learning-based traffic analysis through our two implementations.

5.5 Experiments

This section describes the phases of the four experiments carried out on the two datasets, namely "Random Forest on Bhosale", "Random Forest on PingPong" (Subsection 5.5.1), "PingPong on Bhosale" and "PingPong on PingPong" (Subsection 5.5.2). Each experiment aims to achieve the classification of bidirectional, incoming and outgoing traffic flows, in relation to the various IoT devices within the Smart Home environment. At the end of the experiments, the performance of the classifiers is evaluated in terms of accuracy, precision, recall and F1-score. Classification results, comparison of the PingPong and the Random Forest classifiers, further discussions and considerations will be provided within Chapters 6 and 7 of this thesis.

5.5.1 Random Forest Experiment

The experiments "Random Forest on Bhosale" and "Random Forest on PingPong" were practically carried out in the same way (the two datasets have the same file structure). The only difference between them is that in the former case, classification is performed on a unified set of data, divided into 75% for the training set and 25% for the test set, whereas in the latter case, classification is performed on a dataset already divided into distinct sets of data for training and testing the classifier.

Data Collection and BPF Filtering:

We began the experiment by reading from the two datasets the .PCAP files containing network traffic data of IoT devices. Each .PCAP file was associated with a timestamp file documenting device events, such as powering on and off, playing a video or adjusting bulb intensity. To filter relevant traffic, we extracted the current analyzed IoT device's IP address(es) from a previously populated dictionary and used a BPF filter based on IP address filtering. This filtering process resulted in a list of packets where the IP address(es) of the device appeared as either the source or the destination address of the packets.

Data Preprocessing:

Once we obtained the list of filtered packets, we proceeded with data preprocessing. For each .PCAP file, we extracted and inserted into a list all the packet flows belonging to a 20-second time window around each corresponding event timestamp. This ensured that we only used the necessary and sufficient traffic from the device events for the upcoming classification. Subsequently, from the initial list containing bidirectional traffic, we derived two additional lists: one containing only outgoing packets (in which the device's IP address appears as the source in the packet) and another containing only incoming packets (in which the device's IP address appears as the destination in the packet). Then, for each packet flow, after mapping each packet within the three lists to its respective length, we computed various statistical features, comprising about 40 distinctive characteristics (as outlined by Taylor et al. in [14]), such as mean, minimum and maximum packet sizes, variance, standard deviation and several percentiles. Finally, this set of features was stored in a structured NumPy matrix.

Flow Labeling:

Alongside feature extraction, we assigned a label to each device event for each IoT device under consideration. Each event was then associated with a specific integer via a previously populated dictionary. These labels were eventually added to a separate NumPy array, ensuring alignment with the corresponding extracted features. Indeed, the feature NumPy matrix is characterized by a number of columns equal to the number of features extracted from each flow and a number of rows equal to the number of analyzed flows. As each flow has an associated label, the number of elements within the array of labels is also equal to the number of analyzed flows.

Training and Testing:

Following the "Data Preprocessing" and "Flow Labeling" phases, we proceeded with the training of the Random Forest classifier. Leveraging its functionalities (previously illustrated within Section 5.3), along with the features and labels of the newly extracted and associated flows, the classifier was able to learn the various characteristics and peculiarities related to the events of the IoT devices. Finally, we evaluated the performance of the trained classifier using a distinct set of test data. This evaluation phase aimed to assess the classifier's ability to classify different types of IoT device events with varying degrees of precision, based on the analyzed packet flows.

In conclusion, through this comprehensive experiment, we aimed to demonstrate

the effectiveness of using a Random Forest classifier to classify the traffic of different IoT devices within a smart environment. By integrating packet data with device event timestamps and using advanced feature extraction techniques, we have tried to extract valuable insights into the results that can be achieved with this type of classification, using this type of classifier.

5.5.2 PingPong Experiment

As for the previous two experiments (outlined in Subsection 5.5.1), the experiments "PingPong on Bhosale" and "PingPong on PingPong" are also nearly identical and differ only in some minor details, primarily aimed at making the structure of the Bhosale dataset as similar as possible to that of the PingPong dataset.

Adapting Bhosale Dataset for Compatibility with PingPong Implementation:

Since the PingPong implementation can be considered as a "black box", the only way to make it work with a dataset other than its native one was to take an external dataset (Bhosale in this case) and modify it in order to make it as similar as possible to the structure of the PingPong dataset. The most challenging part of the modification was transforming the Bhosale dataset from a unified set of data into two distinct sets of data already prepared for training and testing the classifier. Since many .PCAP files within the dataset were related to the same event of the same device, we were able to solve this issue by equally dividing these files into two separate folders, with the same purpose of the "standalone" and "smarthome" categories of the PingPong dataset. The only thing that couldn't be changed was that the conditions of the two Smart Home environments of PingPong and Bhosale were slightly different (and thus without a real distinction between "standalone" and "smarthome"), but still compatible and the results proved it. Finally, small modifications were made to the names of the folders and .timestamps files, in order to make them reflect the PingPong "style".

Execution of PingPong Implementation with Different Datasets:

After the previous phase and following the instructions provided to us, we executed the PingPong implementation first with its dataset and then with the Bhosale dataset, obtaining output results that would then be compared with those from the Random Forest Classifier. This part was quite straightforward, since, as mentioned earlier, PingPong is a "black box" and we only had to concern ourselves with providing input data and not with what was done during the execution of the implementation. Nevertheless, the various steps of the PingPong execution are the same as those illustrated by Trimananda et al. in [12].

Management of Results:

At the end of the experiment, we obtained the results in the form of True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN). Using these values, we were able to calculate the classifier's performance, namely: accuracy, precision, recall and F1-score.

In summary, with these two experiments on PingPong, we aimed to test an existing solution based on a different classification approach from that used in the Random Forest classifier experiments. Furthermore, these experiments were very useful as they allowed us to obtain an excellent set of results for comparison between the two approaches, enabling us to delve into an interesting discussion regarding strengths and weaknesses of these techniques and the future implications to ensure that such strategies can be widely used in the context of Access Control policy management within Smart Buildings.

Chapter 6

Results

This Chapter will present the results of several experiments conducted by running two different implementations: the PingPong and the Random Forest Classifier, on two different datasets discussed in Chapter 5. Furthermore, many considerations regarding the two different approaches will be made, addressing two questions:

- What is the accuracy achieved by the Action Identification methods?
- Which approach gives the best results?

6.1 Events of Analyzed Devices

The Table 6.1, shows all the IoT PingPong devices and the corresponding events that were analyzed. The PingPong dataset includes a wide range of devices, ranging from smart bulbs and cameras to thermostats and sprinkler systems. A notable trend is the prevalence of "on/off" events, which are a common event among multiple devices such as Amazon Plug, D-link Plug and Smart Things Plug. This suggests a fundamental and frequently encountered functionality related to the activation and deactivation of the device. Additionally, specific types of devices show specific features corresponding to their functionality. For instance, cameras like the Arlo camera have "stream on/off" events, which emphasize the monitoring aspect, while irrigation systems like Blossom and Rachio have events related to different types of modes.

Device	Events
Amazon plug	on/off
Arlo camera	stream on/off
Blossom sprinkler	mode - quickrun
D-link plug	on/off
D-link siren	on/off
Ecobee thermostat	hvac
Kwikset door lock	lock/unlock
Nest thermostat	fan on/off
Rachio sprinkler	mode - quickrun
Ring alarm	arm/disarm
Roomba vacuum robot	robot mode
Sengled light bulb	on/off - intensity
Smart Things plug	on/off
TP-link bulb	on/off - intensity - color
TP-link plug	on/off
WeMo Insight plug	on/off
WeMo plug	on/off

Table 6.1: PingPong IoT devices and related analyzed events

Table 6.2 provides information about events associated with several devices within the Bhosale dataset. A notable trend here, as in the PingPong dataset, is the prevalence of "startup" events ("on/off" on PingPong dataset) on all devices. The diversity of events shows the different capabilities and functions of the devices; for example, Arlo and Omna cameras include events such as 'night vision' and "video stream", emphasizing their surveillance and video-related features. Echo and Google Home, being smart speakers, are characterized by events such as "weather", "wake word" or "volume adjust", highlighting their ability to provide

weather information and customization of specific settings. Samsung TV instead, in addition to the "startup" event, is characterized by the "play video" event, which indicates a specific action related to media playback.

Device	Events
Arlo camera	startup - night vision - video stream
Echo	startup - weather
Echo Dot	startup - wake word - volume adjust
Google Home	startup - weather - wake word - volume adjust
Omna camera	startup - night vision - video stream
Samsung TV	startup - play video

Table 6.2: Bhosale IoT devices and related analyzed events

6.2 Performance of Random Forest Classifier

Random Forest Classifier on Bhosale Dataset:

The classification results obtained by the Random Forest Classifier on the Bhosale dataset (Table 6.3), indicate high performance across many devices:

- **Arlo camera:** This device achieved perfect score (100%) across accuracy, precision, recall and F1 score, demonstrating excellent classification.
- **Echo, Echo Dot, Google Home and Omna camera:** These devices present similar outstanding performance compared to the previous one with 99% accuracy, precision, recall and F1 score.
- **Samsung TV:** This device, while still good, shows slightly lower performance with 94% accuracy, 96% precision, 90% recall and 93% F1 score.

Device	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
Arlo camera	1,00	1,00	1,00	1,00
Echo	0,99	0,99	0,99	0,99
Echo Dot	0,99	0,99	0,99	0,99
Google Home	1,00	1,00	1,00	1,00
Omna camera	1,00	1,00	1,00	1,00
Samsung TV	0,94	0,96	0,90	0,93

Table 6.3: Classification results obtained by the Random Forest Classifier run on the Bhosale dataset

Random Forest Classifier on PingPong Dataset:

The classification results obtained by the Random Forest Classifier on the PingPong dataset (Table 6.4), demonstrates outstanding performance across a diverse set of devices:

- **Majority of devices:** These devices achieved perfect scores (100%) across accuracy, precision, recall and F1 score, indicating highly reliable classification.
- **Sengled light bulb:** This device shows comparatively lower performance with 56% accuracy, 77% precision, 55% recall and 44% F1 score, indicating an important misclassification.
- **TP-link bulb:** This device displays 92% across all metrics, indicating good but not perfect classification.

In summary, the Random Forest Classifier generally performs exceptionally well across both datasets, with only a few devices within the PingPong dataset, presenting lower classification metrics. Further in-depth investigations into the specifics of these cases have revealed a potential issue with misclassification. Specifically, concerning events such as “intensity” and “color”, unique for the Sengled light bulb and the TP-link bulb, regardless of the temporal delta that is set, the number of packets within the considered network flow proves to be significantly low. This, leads to a lower specificity of the features extracted and computed from the flow, consequently making it more challenging for the classifier to correctly distinguish these types of events from more general events or those with more specific features.

Device	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
Amazon plug	1,00	1,00	1,00	1,00
Arlo camera	1,00	1,00	1,00	1,00
Blossom sprinkler	0,98	0,98	0,97	0,97
D-link plug	1,00	1,00	1,00	1,00
D-link siren	1,00	1,00	1,00	1,00
Ecobee thermostat	1,00	1,00	1,00	1,00
Kwikset door lock	1,00	1,00	1,00	1,00
Nest thermostat	1,00	1,00	1,00	1,00
Rachio sprinkler	1,00	1,00	1,00	1,00
Ring alarm	1,00	1,00	1,00	1,00
Roomba vacuum robot	1,00	1,00	1,00	1,00
Sengled light bulb	0,56	0,77	0,55	0,44
Smart Things plug	1,00	1,00	1,00	1,00
TP-link bulb	0,91	0,93	0,91	0,91
TP-link plug	1,00	1,00	1,00	1,00
WeMo Insight plug	1,00	1,00	1,00	1,00
WeMo plug	1,00	1,00	1,00	1,00

Table 6.4: Classification results obtained by the Random Forest Classifier run on the PingPong dataset

6.3 Performance of PingPong Implementation

PingPong Implementation on Bhosale Dataset:

The PingPong implementation on the Bhosale dataset (Table 6.5) demonstrates great classification performance:

- **Arlo camera, Echo, Echo Dot, Google Home, Omna camera:** These devices achieved high scores across accuracy, precision, recall and F1 score, with values ranging from 94% to 100%.
- **Samsung TV:** This device shows slightly lower performance compared to the other devices, with 98% accuracy, 99% precision, 98% recall and 98% F1 score.

Device	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
Arlo camera	0,96	0,97	0,94	0,95
Echo	0,97	0,99	0,94	0,97
Echo Dot	1,00	1,00	1,00	1,00
Google Home	0,99	0,99	0,98	0,98
Omna camera	1,00	1,00	1,00	1,00
Samsung TV	0,98	0,99	0,98	0,98

Table 6.5: Classification results obtained by the PingPong implementation run on the Bhosale dataset

PingPong Implementation on PingPong Dataset:

The PingPong implementation on its own dataset demonstrates robust classification (Table 6.6) across various devices:

- **Majority of devices:** These devices achieved perfect scores (100%) across accuracy, precision, recall and F1 score, indicating highly reliable classification.
- **TP-link bulb:** This device achieves high scores with 98% accuracy, 99% precision, 98% recall and 98% F1 score.
- **Ecobee thermostat and Nest thermostat:** These devices display slightly lower performance compared to the other devices, with accuracy and F1 score around 94 / 95%.

In summary, the PingPong implementation generally performs well across both datasets, with only a few devices showing slightly lower classification metrics.

Compared to the previous analysis, we can observe how this "fingerprints"-based approach is less affected by the issue of a low number of packets available for analysis. This is very positive as it allows for more accurate classifications even when less information is available.

Device	**Accuracy** (%)	**Precision** (%)	**Recall** (%)	**F1 score** (%)
Amazon plug	1,00	1,00	1,00	1,00
Arlo camera	0,97	0,98	0,97	0,97
Blossom sprinkler	0,95	0,96	0,94	0,95
D-link plug	1,00	1,00	1,00	1,00
D-link siren	1,00	1,00	1,00	1,00
Ecobee thermostat	0,95	0,98	0,91	0,94
Kwikset door lock	1,00	1,00	1,00	1,00
Nest thermostat	0,99	0,99	0,98	0,99
Rachio sprinkler	1,00	1,00	1,00	1,00
Ring alarm	1,00	1,00	1,00	1,00
Roomba vacuum robot	1,00	1,00	1,00	1,00
Sengled light bulb	1,00	1,00	1,00	1,00
Smart Things plug	1,00	1,00	1,00	1,00
TP-link bulb	0,98	0,99	0,98	0,98
TP-link plug	1,00	1,00	1,00	1,00
WeMo Insight plug	0,96	0,99	0,94	0,96
WeMo plug	1,00	1,00	1,00	1,00

Table 6.6: Classification results obtained by the PingPong implementation run on the PingPong dataset

6.4 The Best Action Identification Approach

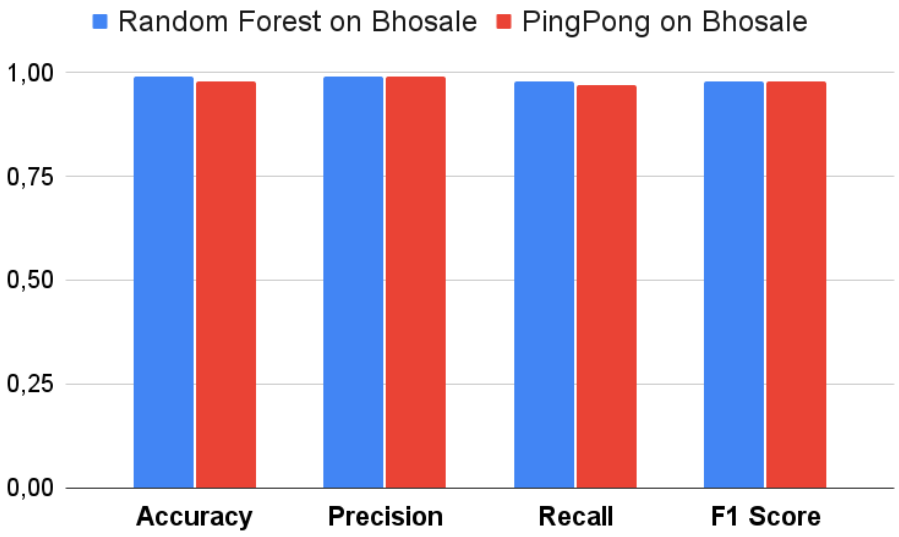

Figure 6.1: Classification results for the Bhosale dataset

Random Forest Classifier run on the Bhosale Dataset:

The Random Forest Classifier shows outstanding performance on the Bhosale dataset, with an accuracy of 99%, precision of 99%, recall of 98% and an F1 score of 98%. These high metrics suggest that the Random Forest model can accurately identify actions within the dataset. Accuracy and recall scores above 98% indicate a well-balanced performance both in terms of minimizing false positives and false negatives.

PingPong Implementation run on the Bhosale Dataset:

The PingPong implementation, when applied to the Bhosale dataset, maintains strong performance. Although the accuracy is a little bit lower than the Random Forest Classifier, it still stands at an impressive 98%. The accuracy of 99% indicates a minimum rate of false positives, while the recall of 97% suggests an effective identification of the action. The F1 score of 98% further underlines the overall robustness of the PingPong implementation on the Bhosale dataset.

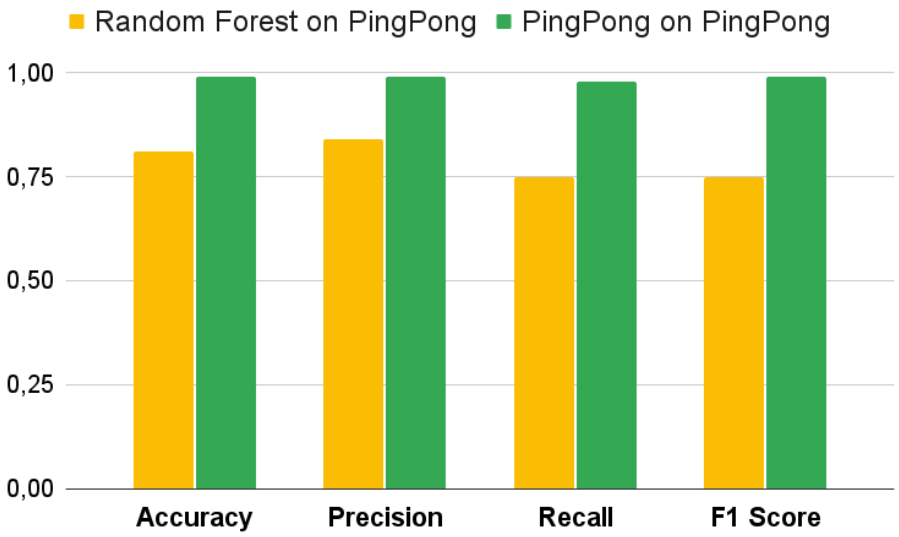

Figure 6.2: Classification results for the PingPong dataset

Random Forest Classifier run on the PingPong Dataset:

Shifting focus to the PingPong dataset, the Random Forest Classifier shows a noticeable drop in performance compared to its performance on the Bhosale dataset. The accuracy, precision, recall and F1 score are considerably reduced (76% on average), indicating challenges in traffic classification on some devices of the dataset. The reasons for this discrepancy, were previously explained within the Section 6.2 of this thesis.

PingPong Implementation run on the PingPong Dataset:

In contrast, the PingPong implementation excels on its native dataset, achieving outstanding and impressive metrics. With an accuracy, precision, recall and F1 score greater than 98%, this implementation shows a great effectiveness and adaptability. High scores indeed, suggest that PingPong is able to correctly classify almost all the events from the traffic flows of devices within in its own dataset.

In conclusion, the analysis reveals that the PingPong implementation demonstrates robust performance for both the PingPong and Bhosale datasets. Its ability to maintain high metrics on its native dataset and to function reasonably well on the Bhosale dataset suggests a great versatility, efficiency and generalization. On the other hand, while the Random Forest Classifier excels on the Bhosale dataset, its reduced performance on the PingPong dataset raises concerns about its adaptability across different datasets. Furthermore, the two implementations significantly differ in terms of internal approach: for traffic classification, the first uses the unique "fingerprints" derived from the events of each device, whereas the second uses features such as mean and variance associated to the length of packets captured from the network flow. However, the decision of choosing one of the two implementations might depend on many different factors, such as the specific characteristics of the dataset to be analyzed or the desired balance between precision, recall, F1 score and overall accuracy. The selection process should consider the nature of the data, their distribution and the potential impact of false positives or false negatives. In addition, consideration should be given to the computational resources required by each implementation, especially if the different devices used have limitations on processing power or memory. Furthermore, the choice could be influenced by the interpretability of the model's decisions. For instance, the 'fingerprint' approach of the PingPong implementation allows to understand the basis of classification decisions for each device. On the other hand, the Random Forest Classifier, relying on statistical features, might be perceived as a 'black box' with less interpretability. All these factors, together with the nature of objectives and limitations, contribute to the selection of the most suitable approach.

Chapter 7

Discussion

Building upon the findings presented in Chapter 6, this Chapter analyzes the implications (Section 7.1) of the results of the two approaches, taking into consideration all the associated limitations (Section 7.2). This allows to provide a more detailed understanding of the research field. In addition, important reflections are made regarding possible future work (Section 7.3), suggesting potential avenues for further research.

7.1 Implications

We will now deepen what was previously discussed in Chapter 2, specifically addressing fundamental principles related to Traffic Analysis (Subsection 2.1.4) and exploring the core principles of Action Identification (Subsection 2.1.5). In the context of Access Control policy management in Smart Buildings indeed, the connection and use of advanced techniques such as Action Identification and Traffic Analysis, supported by machine learning and packet length evaluation, play a crucial role.

Connection Dynamics Learning:

The use of machine learning to analyze packet lengths in data streams from IoT devices enables the learning of specific connection dynamics. Each device, through its activities, creates unique "fingerprints" and Action Identification, leveraging machine learning, can recognize and classify these distinctive features. This level of detail is pivotal, since it allows to create an accurate and precise picture of each device's activity, thus making possible the identification of any deviations from the normal behavior.

The PingPong implementation for instance, relies specifically on those "fingerprints"

(called “signatures”) unique to each activity. This approach involves training a machine learning model in order to classify traffic based on different signatures. The effectiveness of this methodology is highlighted in the results presented in Tables: 6.5 and 6.6 as well as in Figures: 6.1 and 6.2. These illustrate the performance of the PingPong implementation on two different datasets with different devices among them, demonstrating a high and promising performance of the classifier, in total about 98% to 99%, taking into consideration metrics such as accuracy, precision, recall and F1 score.

In addition, the process of learning connection dynamics enables more effective management of Access Control policies. For instance, an IoT device with environmental monitoring functions might be authorized to access only specific areas and specific data, thus avoiding the risk of unauthorized handling. Action Identification, then, ensures that each action is consistent with the role assigned to that device, helping to create more targeted and secure Access Control policies.

Identifying Actions via Packet Length:

The evaluation of packet length offers considerable insight into Action Identification. This approach not only identifies the type of action, but can also distinguish specific details within an action. For example, the distinction between an authorized access request and an unauthorized one, can be based on small variations in packet length. This level of detail allows the implementation of highly customized Access Control policies, adapting the system’s response based on the specific nature of the identified action.

The Random Forest Classifier, for example, takes advantage of this type of feature to classify the diverse packets sent and received by devices within the two datasets. Analyzing Figures 6.1 and 6.2, the performance of this classifier stands out. Specifically, the classifier achieves about 99% accuracy for devices in the Bhosale dataset and approximately 84% accuracy for devices in the PingPong dataset. Going into more detail and considering the Tables 6.5 and 6.6 with a focus on individual devices, it becomes evident that the significant difference in measurements between this and the PingPong implementation comes mainly from two devices (especially the Sengled light bulb). This classifier indeed (as already explained within the Section 6.2), compared to the other classifier based on the event “fingerprint”, presents several classification challenges when the features associated with an event are few and not very specific. In considering which method is most promising between these two classifiers, it is essential to weigh their respective strengths and challenges. Based on these considerations, it is necessary to analyze the types of objectives and the computational resources that are available and then make a

choice based on what is better to prioritize. For example, having the same amount of computational resources available, prioritizing a more reliable and accurate solution across a wider range of devices, even if slightly slower, may lead to choosing the implementation based on event "fingerprints". Vice versa, if a slightly less reliable solution but still highly accurate and faster is acceptable, the approach based on the Random Forest Classifier would be preferred.

However, let's consider an example where the system detects many anomalous packet lengths from a device within a Smart Building. Action Identification can distinguish whether this change is due to a normal firmware update, which can be allowed, or if it indicates an attempt to hack the device. Based on this distinction, the system can apply Access Control policies that allow or block the action, contributing to ensure the security of protected network areas.

Recognition of Usage Trends and Adaptability to Changes in the Network Environment:

Traffic Analysis, supported by machine learning, can recognize specific usage trends of devices. For instance, analyzing the traffic of a device in an office on a normal day, during working hours it could be associated to a "typical" behavior, whereas during the night it could be linked to different activities. Therefore, the recognition of usage trends may help to figure out which are the ordinary activities related to each device within the network. Moreover, in the context of Access Control policy management, we can imagine a situation where an IoT device with facility control functions, shows an unexpected usage trend, such as a significant increase in access requests during a non-business time period. Traffic analysis, identifying this abnormal behavior, can trigger an alert and apply Access Control policies, thus limiting the action of the device during that period and then reducing the potential risks of unauthorized access or malfunction.

In reviewing the results comprehensively, although the activities of two devices were not classified correctly by the Random Forest Classifier's classifier, but, considering a broad range of devices analyzed, it is possible to conclude that the overall performance of the models, turns out to be promising. Such a conclusion holds significant importance, since it indicates the potential for classifying the activities of virtually all current and future market devices in near real-time. This capability allows the immediate management of other aspects and behaviors that were previously unmanageable or very challenging to manage.

Therefore, variations in daily activities or the introduction of new devices will be able to be detected and integrated into Access Control policies, ensuring dynamic and

efficient management. This kind of adaptability could not only assure continuous security, but also allow for flexibility, resulting in constant optimization of Access Control policies. However, the presence of false positives reduces the overall classification accuracy and introduces an additional dimension of complexity, that must be carefully considered. In this context, false positives, consist of the incorrect identification and classification of one or more device events within the traffic flow, potentially leading to a misconfiguration of the Access Control policies. For instance, if the system generates not negligible rate of false positives, it may trigger unnecessary alarms, interventions or unplanned behavior of devices, interrupting the normal flow of network activities within the Smart Building. This can lead to inconvenience for occupants, inefficient use of resources or misinterpretation of security threats.

7.2 Limitations of the Approach

Despite the positive and promising results that this approach, based on two different implementations has highlighted, it is now important to discuss the potential limitations related to the machine learning-based traffic classification. This discussion is particularly paramount, also in anticipation of the use of these techniques for the management of Access Control policies within a Smart Building.

Device-Specific Limitations:

Both implementations exhibit slightly lower performance for certain devices (e.g., Samsung TV in the Random Forest Classifier and Echo and Ecobee thermostat in the PingPong implementation). This highlights the challenge of creating a unique model that excels for all devices due to inherent differences in their communication patterns. Indeed, different devices can present specific communication models, making it difficult to come up with a universal classification approach.

Dataset Specificity:

The models are trained and evaluated on specific datasets (Bhosale and PingPong). While these datasets provide valuable insights, the challenge consists of generalizing models to the wide range of devices and network scenarios present in several Smart Building environments. Models then, might not capture the complexity of new devices or unique usage patterns.

Imbalanced Classes:

In both implementations, some classes have a more significant representation than others, such as the Sengled light bulb within the PingPong dataset. Unbalanced

classes might lead to a bias in the model towards the majority class and may affect its ability to accurately classify the less represented devices.

Static Network Conditions and Behavioral Changes:

Models may be sensitive to specific network conditions prevalent in training data. However, in real-world scenarios, such as in Smart Buildings, network configurations are dynamic and devices might be added or removed. Moreover, Smart Buildings involve human interactions and user behaviors may evolve over time. Changes in user habits or the introduction of new devices could not be promptly reflected in the models. Models then, can struggle in order to adapt to changing conditions, thus affecting their reliability in dynamic environments.

Limited Feature Set:

Both implementations rely on specific features such as event "fingerprints", packet lengths, mean, variance, etc. This approach may not capture the whole complexity of communication patterns, especially with the evolution of Smart Building technologies. Therefore, considering more sophisticated features or exploiting deep learning techniques, could provide a more nuanced understanding of communication patterns within Smart Buildings.

Security and Adversarial Attacks:

Models may be vulnerable to adversarial attacks that aim to manipulate network traffic patterns. Therefore, ensuring the robustness of the models against intentional malicious behavior or attacks, is crucial for the security of Access Control systems within Smart Buildings.

Real-Time Constraints:

Assessing the real-time applicability of the implementations is very important, especially within Smart Building contexts, where Access Control decisions require low latency. Delays, particularly in authentication and verification processes, can result in unauthorized access, thereby compromising the overall security of the Smart Building. Furthermore, such delays in Access Control can significantly impact emergency response times, increasing the risk for occupants. For these and other reasons, evaluating the computational efficiency of models and their ability to provide real-time responses is essential for practical deployment.

Privacy Concerns:

Network traffic analysis for device classification introduces privacy concerns. Finding a balance between accurate classification and user privacy is paramount. The implementation of privacy protection mechanisms, such as anonymization or federated learning, is crucial to comply with data protection regulations.

In conclusion then, addressing these limitations is crucial for implementing effective and reliable Access Control systems in the dynamic and diverse environment of a Smart Building. Continuous model refinement, adaptation to evolving device configurations and a strong focus on privacy and security considerations are key elements for a successful implementation and operation of IoT devices within Smart Building environments.

7.3 Future Work

In the context of managing Access Control policies in a Smart Building, a promising future perspective is to deepen and enhance the machine learning-based approach for traffic classification. This involves using advanced algorithms such as deep neural networks, ensemble methods like Gradient Boosting and clustering techniques such as k-means or hierarchical clustering to learn and dynamically adapt to the communication patterns of devices within the Building. A key research area could focus on customizing Access Control policies based on specific contexts. By using machine learning algorithms, like Support Vector Machines (SVM) or decision trees, it is possible to analyze traffic data in real time and adapt the policies in response to changing conditions within the Smart Building. This approach would allow for a more flexible and adaptable management of Access Control policies while ensuring high-security standards. Another important aspect concerns the implementation of Access Control policies focused on the past behavior of devices. Through the application of machine learning algorithms capable of recognizing patterns and anomalies in network traffic, a system could be developed to automatically adapt to changes in device usage patterns, thus improving the overall effectiveness of these policies.

The interaction between machine learning-based Access Control policies and device communication protocols in Smart Buildings is another important point. Research could explore how to optimize the integration of these policies across different types of devices, thereby ensuring an accurate classification and a consistent management of authorizations. Furthermore, a critical aspect concerns the robustness of the traffic classification system. Implementing machine learning techniques that are resilient to changes in traffic patterns is essential to ensure long-term Access

Control policy accuracy, especially considering the continuous evolution of smart environments. Finally, it is also crucial to address the challenges associated with false positives. A comprehensive management strategy must be integrated into the system. This could include further verification steps or setting flexible thresholds that consider the unique characteristics of Smart Buildings. Finding the right balance between accuracy and adaptability is very important in order to ensure that Access Control policies remain effective, responsive and adapted to the evolving dynamics of the intelligent construction environment. In summary, future work should focus on the evolution and optimization of the machine learning-based approach for traffic classification to create more flexible, intelligent and adaptable Access Control policies within the Smart Building environments.

Chapter 8

Conclusions

The research performed in this thesis illustrates a robust and innovative approach in the management of Access Control policies within Smart Buildings. The adoption of machine learning-based techniques for traffic classification is an important step towards creating intelligent, adaptable and secure environments. The research also showed how the customization of Access Control policies according to specific contexts could be made possible through the dynamic analysis of network traffic, thus leading to a more flexible and responsive management. This approach not only enhances security, but also the overall user experience within the Smart Building. Indeed, Access Control policies can be tailored to the diverse needs of different users, across various areas of the building and at different times of the day. This allows to create a user-centric system while maintaining a high level of security. Furthermore, a key feature that has emerged is the importance of applying machine learning algorithms to model and predict device behaviors. This not only allows for more accurate classification of network traffic, but also provides a dynamic basis for continuous Access Control management. The introduction of a predictive element, based on learned models, helps to improve the resilience of the system and additionally, allows it to anticipate and address not only current challenges, but also future ones, as it is able to predict and adapt dynamically to changes in user behaviors and environmental dynamics. Essentially, the use of predictive models based on machine learning is a key element to ensure the preparation and the effectiveness of the Access Control policy management system in the face of a constantly evolving landscape. Moreover, the interaction between machine learning-based algorithms and device communication protocols represents an important area of convergence. The research has emphasized the importance of providing consistent permission management in a heterogeneous environment and demonstrates the importance of optimizing integration between policies and devices, ensuring accuracy and consistency in classification. The complexity of the traffic classification system has also emerged as a key factor. The implementation of machine learning techniques

capable of adapting to changes in traffic patterns, is essential to ensure the security of Access Control policies throughout the years, especially in smart environments characterized by continuous evolution. In conclusion, this research not only provides an in-depth review and analysis of current developments in Access Control policy management, but also outlines a path for future development. The integration of machine learning-based techniques is crucial for the transformation of Smart Buildings into adaptable, secure and intelligent environments and also shapes the future where Access Control policies efficiently adapt to the changing needs and dynamics of users.

Appendix A

Random Forest Classifier: Code Modules and Main Functions

In this appendix, the different modules with their respective main functions within the project are described. These functions are responsible for the entire flow of operations, from preprocessing the data within the dataset to evaluating the Random Forest classifier.

Utilities Module (utilities.py):

This module provides utility functions for various tasks, including converting timestamps, filtering packets based on timestamps and computing statistical features from packet lengths.

The function *compute_statistical_features* is the main function of this module. It calculates several statistical characteristics based on packet lengths. The computed characteristics include the maximum, skewness, variance, standard deviation, kurtosis, median absolute deviation and percentiles (e.g., 90th, 80th, 70th, 60th, 50th, 40th, 30th, 20th and 10th percentiles) for both the complete set of packets and the separate sets of incoming and outgoing packets.

- **Parameters:**
  - **packets:** The list of packets including the bidiriectional traffic from and to the current analyzed IoT device.
  - **incoming_packets:** The list of packets including only incoming traffic to the current analyzed IoT device.

  - **outgoing_packets:** The list of packets including only outgoing traffic from the current analyzed IoT device.

- **Return Value:**

  - A NumPy array containing the calculated statistical characteristics for further processing and analysis.

```
def compute_statistical_features(packets, incoming_packets,
    outgoing_packets)
```

Listing A.1: Prototype of *compute_statistical_features* function

Dataset Formatting Module (dataset_formatter.py):

This module is responsible for reading dataset files, including .PCAP and .timestamps files. It extracts features from packet flows, combines them with corresponding labels and prepares the data for training the classifier. The module includes functions to read .timestamps files, read .PCAP files and format the overall dataset.

The function *read_dataset_files* is the main function of this module. It iterates through all files in the directory and subdirectories, searching for .PCAP files. For each .PCAP file found, it reads the packets and filters them based on timestamps obtained from a .timestamps file. Statistical features are computed for each flow of packets using the *compute_statistical_features* function. This function also retrieves the label associated with the flow using the *get_flow_label* function. If a label is found, it is appended to the list of labels. Finally, the function returns the computed features and labels as NumPy arrays.

- **Parameters:**

  - **folder_path:** The path to the folder containing .PCAP and .timestamps files.
  - **folder_name:** The name of the folder containing the dataset.

- **Return Value:** A tuple of NumPy arrays representing data and target for the classifier.

```
def read_dataset_files(folder_path, folder_name)
```

Listing A.2: Prototype of *read_dataset_files* function

Classifier Module (classifier_module.py):

The classifier module implements a Random Forest classifier. It includes a function to train and evaluate the classifier both for a "single" and a "double" dataset. The module also showcases the use of cross-validation and scaling features.

The function *train_and_evaluate_classifier* is the main function of this module. It is used to train the classifier using the training set and evaluate its performance using the test set. It first initializes a Random Forest classifier with specified parameters such as the number of estimators, random state and criteria for splitting nodes. Then, it trains the classifier using the training data (X_train and y_train). Next, it makes predictions on the test data (X_test) and evaluates the model's performance using the true labels (y_test). The accuracy of the model and the classification report are computed and returned as a tuple.

- **Parameters:**
  - **X_train:** Array-like or matrix of shape (n_samples, n_features) representing the features from the training set of the dataset.
  - **y_train:** Array-like of shape (n_samples) representing the class labels associated with the features in X_train from the training set of the dataset.
  - **X_test:** Array-like or matrix of shape (n_samples, n_features) representing the features from the test set of the dataset.
  - **y_test:** Array-like of shape (n_samples) representing the class labels associated with the features in X_test from the test set of the dataset.
- **Return Value:** A tuple containing the model's accuracy and the classification report.

```
def train_and_evaluate_classifier(X_train, y_train, X_test, y_test)
```

Listing A.3: Prototype of *train_and_evaluate_classifier* function

Flow Labeling Module (flow_labeling.py):

The flow labeling module provides functions for classifying events based on labels and indices obtained from timestamps. It assigns integer labels to different IoT device events, allowing for easier interpretation and analysis of the classification results.

The function *get_flow_label* is the main function of this module. It uses a dictionary, called *label_dictionary*, which maps each event to its corresponding integer label. It then iterates over the dictionary and checks if the event is present as a key. If found, it returns the corresponding label. If no matching label is found, it returns the integer -1.

- **Parameters:**
  - **activity:** The activity (event) coming from the type of captured flow.
- **Return value:** The integer label associated with the event or the integer -1.

```
def get_flow_label(activity)
```

Listing A.4: Prototype of *get_flow_label* function

IP Addresses Module (ip_addresses.py):

The IP Addresses module provides a function to map device names to their corresponding IP addresses. This is useful for identifying the source and destination of network traffic flows.

The function *get_ip_address* uses a dictionary, called *device_ip_dictionary*, which maps each device name to its corresponding IP address(es). It then iterates over the dictionary and checks if the device name is present as a key. If found, it returns the corresponding IP address(es). If no matching IP address is found, it returns the integer -1.

- **Parameters:**
  - **device_name:** The name of the device.
- **Return Value:** The IP address associated with the device name or the integer -1.

Bibliography

[1] Daniele Bringhenti, Guido Marchetto, Riccardo Sisto, Fulvio Valenza, and Jalolliddin Yusupov. «Towards a fully automated and optimized network security functions orchestration». In: *2019 4th International Conference on Computing, Communications and Security (ICCCS)*. IEEE. 2019, pp. 1–7 (cit. on p. 15).

[2] Daniele Bringhenti, Guido Marchetto, Riccardo Sisto, and Fulvio Valenza. «Automation for Network Security Configuration: State of the Art and Research Trends». In: *ACM Comput. Surv.* 56.3 (Oct. 2023). DOI: 10.1145/3616401 (cit. on p. 15).

[3] Daniele Bringhenti, Guido Marchetto, Riccardo Sisto, Fulvio Valenza, and Jalolliddin Yusupov. «Automated Firewall Configuration in Virtual Networks». In: *IEEE Transactions on Dependable and Secure Computing* 20.2 (2023), pp. 1559–1576. DOI: 10.1109/TDSC.2022.3160293 (cit. on p. 15).

[4] Amit Kumar Sikder, Leonardo Babun, Z. Berkay Celik, Hidayet Aksu, Patrick McDaniel, Engin Kirda, and A. Selcuk Uluagac. «Who's Controlling My Device? Multi-User Multi-Device-Aware Access Control System for Shared Smart Home Environment». In: *ACM Trans. Internet Things* 3.4 (Sept. 2022). DOI: 10.1145/3543513 (cit. on p. 15).

[5] Simone Cirani, Marco Picone, Pietro Gonizzi, Luca Veltri, and Gianluigi Ferrari. «IoT-OAS: An OAuth-Based Authorization Service Architecture for Secure Services in IoT Scenarios». In: *IEEE Sensors Journal* 15.2 (2015), pp. 1224–1234. DOI: 10.1109/JSEN.2014.2361406 (cit. on p. 15).

[6] Honglei Ren, You Song, Siyu Yang, and Fangling Situ. «Secure smart home: A voiceprint and internet based authentication system for remote accessing». In: *2016 11th International Conference on Computer Science & Education (ICCSE)*. 2016, pp. 247–251. DOI: 10.1109/ICCSE.2016.7581588 (cit. on p. 15).

[7] Huansheng Ning, Hong Liu, and Laurence T. Yang. «Cyberentity Security in the Internet of Things». In: *Computer* 46.4 (2013), pp. 46–53. DOI: 10.1109/MC.2013.74 (cit. on p. 16).

[8] Fulvio Valenza, Erisa Karafili, Rodrigo Vieira Steiner, and Emil C. Lupu. «A Hybrid Threat Model for Smart Systems». In: *IEEE Transactions on Dependable and Secure Computing* 20.5 (2023), pp. 4403–4417. DOI: 10.1109/TDSC.2022.3213577 (cit. on p. 16).

[9] Daniele Bringhenti, Fulvio Valenza, and Cataldo Basile. «Toward Cybersecurity Personalization in Smart Homes». In: *IEEE Security & Privacy* 20.1 (2022), pp. 45–53. DOI: 10.1109/MSEC.2021.3117471 (cit. on p. 16).

[10] Ibrahim Alrashdi, Ali Alqazzaz, Esam Aloufi, Raed Alharthi, Mohamed Zohdy, and Hua Ming. «AD-IoT: Anomaly Detection of IoT Cyberattacks in Smart City Using Machine Learning». In: *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*. 2019, pp. 0305–0310. DOI: 10.1109/CCWC.2019.8666450 (cit. on pp. 16, 43).

[11] Vishwajeet Bhosale, Lorenzo De Carli, and Indrakshi Ray. «Detection of Anomalous User Activity for Home IoT Devices.» In: *IoTBDS*. 2021, pp. 309–314 (cit. on pp. 16, 43, 46, 48).

[12] Rahmadi Trimananda, Janus Varmarken, Athina Markopoulou, and Brian Demsky. «Packet-Level Signatures for Smart Home Devices». In: *Network and Distributed Systems Security (NDSS) Symposium* 2020 (). DOI: 10.14722/ndss2020.24097 (cit. on pp. 16, 43, 48, 53).

[13] Abbas Acar, Hossein Fereidooni, Tigist Abera, Amit Kumar Sikder, Markus Miettinen, Hidayet Aksu, Mauro Conti, Ahmad-Reza Sadeghi, and Selcuk Uluagac. «Peek-a-Boo: I See Your Smart Home Activities, Even Encrypted!» In: *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. WiSec '20. Linz, Austria: Association for Computing Machinery, 2020, pp. 207–218. DOI: 10.1145/3395351.3399421 (cit. on pp. 16, 43).

[14] Vincent F. Taylor, Riccardo Spolaor, Mauro Conti, and Ivan Martinovic. «AppScanner: Automatic Fingerprinting of Smartphone Apps from Encrypted Network Traffic». In: *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. 2016, pp. 439–454. DOI: 10.1109/EuroSP.2016.40 (cit. on pp. 17, 47, 52).

[15] Markus Miettinen, Samuel Marchal, Ibbad Hafeez, N. Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. «IoT SENTINEL: Automated Device-Type Identification for Security Enforcement in IoT». In: *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. 2017, pp. 2177–2184. DOI: 10.1109/ICDCS.2017.283 (cit. on pp. 17, 21).

[16] TJ OConnor, Reham Mohamed, Markus Miettinen, William Enck, Bradley Reaves, and Ahmad-Reza Sadeghi. «HomeSnitch: Behavior transparency and control for smart home IoT devices». In: *Proceedings of the 12th conference on security and privacy in wireless and mobile networks*. 2019, pp. 128–138 (cit. on pp. 17, 43, 46).

[17] Steffen Wendzel. «How to Increase the Security of Smart Buildings?» In: *Commun. ACM* 59.5 (Apr. 2016), pp. 47–49. DOI: 10.1145/2828636 (cit. on p. 17).

[18] Pierre Ciholas, Aidan Lennie, Parvin Sadigova, and Jose M. Such. *The Security of Smart Buildings: a Systematic Literature Review*. 2019. arXiv: 1901.05837 [cs.CR] (cit. on p. 17).

[19] Nian Xue, Xin Huang, and Jie Zhang. «S2Net: A Security Framework for Software Defined Intelligent Building Networks». In: *2016 IEEE Trustcom/BigDataSE/ISPA*. 2016, pp. 654–661. DOI: 10.1109/TrustCom.2016.0122 (cit. on p. 17).

[20] Luca Morgese Zangrandi, Thijs Van Ede, Tim Booij, Savio Sciancalepore, Luca Allodi, and Andrea Continella. «Stepping out of the MUD: Contextual Threat Information for IoT Devices with Manufacturer-Provided Behavior Profiles». In: *Proceedings of the 38th Annual Computer Security Applications Conference*. ACSAC '22. , Austin, TX, USA, Association for Computing Machinery, 2022, pp. 467–480. DOI: 10.1145/3564625.3564644 (cit. on p. 17).

[21] José L. Hernández-Ramos, M. Victoria Moreno, Jorge Bernal Bernabé, Dan García Carrillo, and Antonio F. Skarmeta. «SAFIR: Secure access framework for IoT-enabled services on smart buildings». In: *Journal of Computer and System Sciences* 81.8 (2015), pp. 1452–1463. DOI: https://doi.org/10.1016/j.jcss.2014.12.021 (cit. on p. 18).

[22] Anurag Verma, Surya Prakash, Vishal Srivastava, Anuj Kumar, and Subhas Chandra Mukhopadhyay. «Sensing, Controlling, and IoT Infrastructure in Smart Building: A Review». In: *IEEE Sensors Journal* 19.20 (2019), pp. 9036–9046. DOI: 10.1109/JSEN.2019.2922409 (cit. on p. 18).

[23] Ayyoob Hamza, Dinesha Ranathunga, Hassan Habibi Gharakheili, Theophilus A. Benson, Matthew Roughan, and Vijay Sivaraman. «Verifying and Monitoring IoTs Network Behavior Using MUD Profiles». In: *IEEE Transactions on Dependable and Secure Computing* 19.1 (Jan. 2022), pp. 1–18. DOI: 10.1109/tdsc.2020.2997898 (cit. on p. 18).

[24] Ayyoob Hamza, Dinesha Ranathunga, Hassan Habibi Gharakheili, Matthew Roughan, and Vijay Sivaraman. «Clear as MUD: Generating, Validating and Applying IoT Behavioral Profiles». In: *Proceedings of the 2018 Workshop on IoT Security and Privacy*. IoT S&P '18. Budapest, Hungary: Association for Computing Machinery, 2018, pp. 8–14. DOI: 10.1145/3229565.3229566 (cit. on p. 18).

[25] Muhammad Usman Younus, Saif ul Islam, Ihsan Ali, Suleman Khan, and Muhammad Khurram Khan. «A survey on software defined networking enabled smart buildings: Architecture, challenges and use cases». In: *Journal of Network and Computer Applications* 137 (2019), pp. 62–77. DOI: https://doi.org/10.1016/j.jnca.2019.04.002 (cit. on p. 18).

[26] Seyed Kaveh Fayazbakhsh, Vyas Sekar, Minlan Yu, and Jeffrey C. Mogul. «FlowTags: Enforcing Network-Wide Policies in the Presence of Dynamic Middlebox Actions». In: *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*. HotSDN '13. Hong Kong, China: Association for Computing Machinery, 2013, pp. 19–24. DOI: 10.1145/2491185.2491203 (cit. on p. 18).

[27] Zorigtbaatar Chuluundorj, Curtis R. Taylor, Robert J. Walls, and Craig A. Shue. «Can the User Help? Leveraging User Actions for Network Profiling». In: *2021 Eighth International Conference on Software Defined Systems (SDS)*. 2021, pp. 1–8. DOI: 10.1109/SDS54264.2021.9732164 (cit. on p. 18).