EAVESDROPPING-DRIVEN PROFILING ATTACKS ON ENCRYPTED WIFI NETWORKS: UNVEILING VULNERABILITIES IN IOT DEVICE SECURITY

by

IBRAHIM ALWHBI
M.Sc. in Information Assurance and Cyberecurity- Melbourne, FL (2018 – 2019)
B.Sc. in Computer Information Systems- Tampa, FL (2011 – 2015)

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida

Summer Term
2024

Major Professor: Dr. Cliff Zou

ABSTRACT

Abstract—This dissertation investigates the privacy implications of WiFi communication in Internet-of-Things (IoT) environments, focusing on the threat posed by out-of-network observers. Recent research has shown that in-network observers can glean information about IoT devices, user identities, and activities. However, the potential for information inference by out-of-network observers, who do not have WiFi network access, has not been thoroughly examined. The first study provides a detailed summary dataset, utilizing Random Forest for data summary classification. This study highlights the significant privacy threat to WiFi networks and IoT applications from out-of-network observers.

Building on this investigation, the second study extends the research by utilizing a new set of time series monitored WiFi data frames and advanced machine learning algorithms, specifically xGboost, for Time Series classification. This extension achieved high accuracy of up to 94% in identifying IoT devices and their working status, demonstrating faster IoT device profiling while maintaining classification accuracy. Furthermore, the study underscores the ease with which outside intruders can harm IoT devices without joining a WiFi network, launching attacks quickly and leaving no detectable footprints.

Additionally, the dissertation presents a comprehensive survey of recent advancements in machine-learning-driven encrypted traffic analysis and classification. Given the challenges posed by encryption for traditional packet and traffic inspection, understanding and classifying encrypted traffic are crucial. The survey provides insights into utilizing machine learning for encrypted network traffic analysis and classification, reviewing state-

of-the-art techniques and methodologies. This survey serves as a valuable resource for network administrators, cybersecurity professionals, and policy enforcement entities, offering insights into current practices and future directions in encrypted traffic analysis and classification.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, **Dr. Cliff Zou**, for his guidance, support, and invaluable feedback throughout the process of my PhD journey and writing this dissertation. I am also immensely thankful to my committee members, Dr. David Mohaisen, Dr. Xueqiang Wang, and Dr. Yao Li, for their expertise and insightful suggestions.

I owe a debt of gratitude to my family brothers and sisters for their unwavering support and encouragement. Their belief in me has been a constant source of strength. Special thanks to my mother, **Fotaimah Alaofi** , whose love and encouragement have been my guiding light, and to my daughter, **Sahab**, for being my source of joy and motivation. I am also grateful to my wife for her love, patience, and unwavering support throughout this journey.

I want to express a special thank you to **Dr. David Mohaisen**, who has helped me immensely and has directed us like an older brother. His guidance and support have been invaluable.

I also want to remember my uncle and best friend, **Naif H Alwhbi**, who passed away in the beginning of this year. His wisdom and encouragement will always be cherished.

Last but not least, I want to thank the great man and my back my father, **Abdullah Alwhbi** for his love, sacrifices, and endless encouragement. His unwavering belief in my abilities has been my driving force.

I am truly grateful to all who have contributed to this work and supported me along this journey.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

The proliferation of Internet of Things (IoT) devices connected to WiFi networks has introduced new challenges in maintaining the security and privacy of these networks. With the increasing adoption of IoT devices in various domains such as healthcare, smart homes, and industrial automation, the need to secure these devices against potential attacks has become paramount. Attackers are increasingly targeting IoT devices due to their vulnerabilities, using techniques such as packet sniffing and traffic analysis to gather sensitive information [50].

This dissertation focuses on the privacy issues arising from the potential to fingerprint IoT devices via their WiFi network connections, particularly investigating attacks conducted by external attackers without network access. While existing research has primarily focused on attacks from within the network, such as through compromised devices or malicious insiders, the possibility of attacks from outside the network presents a new and challenging scenario.

In Chapter 2, The purpose of a survey paper is to provide a comprehensive and organized overview of existing research, developments, and trends in a particular field or topic. It only aims to summarize and synthesize existing knowledge, identify gaps or areas needing further research, and provide insights for researchers, practitioners, and policymakers. it's served as valuable resources for those new to the field, as well as for experts looking for a summary of the state of the art and future directions.

The main contribution of this chapter, providing a comprehensive overview and analysis of encrypted network traffic analysis and classification utilizing machine learning techniques, has significant impacts on the field of cybersecurity:

1. **Enhancing network security monitoring and threat detection**: By leveraging machine learning to analyze encrypted network traffic metadata (packet sizes, timing, source/destination, etc.), this work enables the identification of malicious activities, anomalies, and potential threats even in encrypted communications, which traditional methods cannot inspect directly.

2. **Improving encrypted traffic classification**: The chapter explores various machine learning approaches (supervised, unsupervised, reinforcement learning) and techniques (statistical analysis, distance metrics) for accurately classifying different types of encrypted traffic, applications, and protocols, crucial for network visibility and policy enforcement.

3. **Addressing the challenges of encryption**: As encryption becomes more prevalent, obscuring payload contents, the methods discussed in this chapter provide ways to gain insights from encrypted traffic without compromising data confidentiality, addressing a major challenge in cybersecurity monitoring.

4. **Enabling proactive security measures**: By detecting patterns and anomalies in encrypted traffic using machine learning, this work allows for proactive security measures, such as identifying zero-day attacks, advanced persistent threats (APTs), and insider threats that evade traditional security controls.

5. **Applicability across domains**: The chapter explores the application of encrypted traffic analysis using machine learning in various domains, including Internet of Things (IoT) devices, mobile applications, and web applications, demonstrating the broad impact of this research.

6. **Balancing security and privacy**: The chapter highlights the importance of addressing privacy concerns and legal implications when conducting encrypted traffic analysis, promoting a balanced approach to cybersecurity that respects user privacy and adheres to ethical and legal requirements.

Overall, this chapter's main contribution advances the field of cybersecurity by providing effective methods to analyze and classify encrypted network traffic, enhancing threat detection, network visibility, and proactive security measures while addressing the challenges posed by widespread encryption and considering privacy and legal aspects.

we delve into the existing literature surrounding encrypted traffic analysis, machine learning (ML) applications in cybersecurity, encrypted network traffic analysis and classification, Internet of Things (IoT) security, and network traffic processing and inspection. This comprehensive review sets the foundation for our research by highlighting the current state of knowledge and identifying areas where further research is needed. We explore the various techniques and methodologies used in previous studies and identify gaps that our research aims to address [13]. [1]

Chapter 3 is dedicated to the practical aspects of detecting IoT devices by monitoring en-

[1] Ibrahim A. Alwhbi, Cliff C. Zou, and Reem N. Alharbi. "Encrypted Network Traffic Analysis and Classification Utilizing Machine Learning". In: Sensors 24.11(2024).

crypted wireless network traffic. We outline the threat model and assumptions, describe the methodology for capturing out-of-network encrypted WiFi traffic, and detail the process of data processing and profiling using machine learning algorithms. The chapter also includes a discussion on observable data fields in out-of-network monitoring and presents preliminary data analysis results. By focusing on out-of-network monitoring, we aim to shed light on the potential vulnerabilities of IoT devices to external attackers.

The chapter presents findings from profiling 12 real-world IoT devices in different operational states and analyzing the prediction accuracy. Our most effective technique managed to identify devices and their operational states (idle vs. active) with an average accuracy of 95% [14]. [2]

In this work, I was an equal contributor. I was involved in the conceptualization, data collection, experimental setup, analysis, and writing of the paper. Although not the first author, my contributions were pivotal in the research's success.

This chapter is included in my dissertation because it represents a critical portion of my research on the privacy issues associated with fingerprinting IoT devices through their encrypted wireless network traffic. It directly aligns with the core objectives of my dissertation and is not being used in another dissertation.

In Chapter 4, we explore the concepts, techniques, and use cases of time-series data in the context of IoT device detection. Time-series data plays a crucial role in understanding the

[2]Mnassar Alyami, Ibrahim Alharbi, Cliff Zou, Yan Solihin, and Karl Ackerman. "WiFi-based IoT Devices Profiling Attack based on Eavesdropping of Encrypted WiFi Traffic". In: 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC). 2022, pp. 385–392.

behavior of IoT devices over time, allowing us to identify patterns and anomalies that can be used for device detection. We define the problem statement, outline the threat model, and discuss assumptions related to time-series data analysis. The chapter also presents a system architecture for verifying collective movement and details the data processing and analysis techniques used in our study. Results and discussions from this chapter provide insights into the effectiveness of using time-series data for IoT device detection.

The chapter showcases the results from profiling 10 different real-world IoT devices in various operational states and examining the prediction accuracy. Our most successful method was able to identify the devices and their operational states (idle vs. active) with an average accuracy of 94% [9]. [3]

Chapter 5 concludes the dissertation by summarizing the findings from our investigation into fingerprinting IoT devices from outside the network. We discuss the implications of our research on network security and privacy, and outline future research directions in this field. The chapter also reflects on the contributions of the dissertation and its potential impact on the field of cybersecurity and IoT device security. By highlighting the challenges and opportunities in securing IoT devices connected to WiFi networks, we aim to contribute to the development of more secure and privacy-preserving IoT systems.

In summary, this dissertation contributes to the understanding of privacy issues in IoT device security and provides practical insights into detecting and mitigating attacks on IoT devices connected to WiFi networks. The research presented here advances the field of

[3] I.A. Alharbi, A.J. Almalki, M. Alyami, C. Zou, and Y. Solihin. "Profiling Attack on WiFi-based IoT Devices using an Eavesdropping of an Encrypted Data Frames". In: Adv. Sci. Technol. Eng. Syst. J. 7 (2022), pp. 49–57.

cybersecurity by addressing the challenges posed by external attackers without network access, and opens up new avenues for securing IoT devices in an increasingly connected world.

CHAPTER 2: Literature Review, Encrypted Network Traffic Analysis and Classification Utilizing Machine Learning

The main contribution of this chapter, providing a comprehensive overview and analysis of encrypted network traffic analysis and classification utilizing machine learning techniques, has significant impacts on the field of cybersecurity by enhancing network security monitoring and threat detection through the analysis of encrypted network traffic metadata (such as packet sizes, timing, source/destination), enabling the identification of malicious activities, anomalies, and potential threats even in encrypted communications, which traditional methods cannot inspect directly. Additionally, it improves encrypted traffic classification by exploring various machine learning approaches (supervised, unsupervised, reinforcement learning) and techniques (statistical analysis, distance metrics) for accurately classifying different types of encrypted traffic, applications, and protocols, crucial for network visibility and policy enforcement.

The chapter addresses the challenges posed by encryption, which obscures payload contents, by providing methods to gain insights from encrypted traffic without compromising data confidentiality. Furthermore, it enables proactive security measures by detecting patterns and anomalies in encrypted traffic, allowing for the identification of zero-day attacks, advanced persistent threats (APTs), and insider threats that evade traditional security controls. The applicability of these methods across various domains, including Internet of Things (IoT) devices, mobile applications, and web applications, demonstrates the broad impact of this research.

Finally, the chapter highlights the importance of balancing security and privacy by addressing privacy concerns and legal implications when conducting encrypted traffic analysis, promoting a balanced approach to cybersecurity that respects user privacy and adheres to ethical and legal requirements. Overall, this chapter advances the field of cybersecurity by providing effective methods to analyze and classify encrypted network traffic, enhancing threat detection, network visibility, and proactive security measures while addressing the challenges posed by widespread encryption and considering privacy and legal aspects.

2.1 Utilize Machine Learning in Cybersecurity

Machine learning (ML) is a branch of artificial intelligence (AI) that empowers computers to learn and improve from experience without being explicitly programmed. It involves the development of algorithms that enable systems to automatically learn and make predictions or decisions based on data. At its core, ML algorithms leverage statistical techniques to identify patterns and relationships within datasets, allowing them to generalize from past examples and make accurate predictions on unseen data.

In the context of cybersecurity, machine learning plays a crucial role in enhancing threat detection, risk assessment, and anomaly detection. By analyzing vast amounts of data, ML algorithms can identify patterns indicative of malicious activity, enabling early detection and mitigation of security threats. ML techniques are particularly effective in cybersecurity due to their ability to process large-scale data in real-time and adapt to evolving threats.

We summarize the main cybersecurity application areas that people can utilize ML to enhance cybersecurity defense and effectiveness:

- Threat Detection: ML algorithms can analyze network traffic, system logs, and user behavior to identify abnormal patterns indicative of security threats such as malware infections, intrusion attempts, and unauthorized access.
- Anomaly Detection: ML models can learn the normal behavior of systems and users and detect deviations from this baseline, signaling potential security breaches or anomalies. This approach is particularly useful for detecting zero-day attacks and insider threats.
- Predictive Analysis: ML algorithms can predict potential security incidents based on historical data and ongoing trends, enabling proactive risk management and mitigation strategies.
- Vulnerability Management: ML techniques can be used to identify vulnerabilities in software and systems by analyzing code, network configurations, and historical data, helping organizations prioritize and remediate security weaknesses.
- Behavioral Analysis: ML algorithms can analyze user behavior, application usage patterns, and system interactions to identify suspicious activities and detect advanced persistent threats (APTs) that evade traditional security measures

2.2 Machine Learning (ML)

Machine Learning is a subset of artificial intelligence (AI) that focuses on developing algorithms and models that enable computers to learn from data. Instead of being explicitly programmed to perform a task, a machine learning system uses statistical techniques to learn patterns and make predictions or decisions without being explicitly programmed for the task [107, 75]. Machine learning is a specialized area within the study of artificial intelligence (AI) that concentrates on creating algorithms and statistical models that allow computers to carry out tasks without relying on explicit programming. The fundamental concept underlying machine learning is to empower machines to acquire knowledge from data and enhance their capabilities progressively [102].

Machine learning models acquire knowledge from a dataset, which is usually split into two parts: the training set and the testing set. The training data is utilized to instruct the model, whereas the testing data assesses its performance. Features are the independent variables or characteristics that the model utilizes to generate predictions or make judgments [70]. Machine learning algorithms refer to the mathematical models used to analyze data and identify trends. These types can be classified into supervised learning, unsupervised learning, and reinforcement learning Figure 2.1. The model undergoes training using a dataset that is labeled, meaning each input is associated with its corresponding output. The objective is to acquire knowledge of a transformation from given inputs to corresponding outputs [99]. The model is provided with unlabeled data and must identify patterns or structure within it without explicit instructions. The model acquires knowledge through active engagement with an environment and receives feedback in the form of rewards or

punishments.

A model is a depiction of the acquired patterns from the training data. It could be a mathematical formula, decision tree, neural network, or other structures. Common forms of models include linear regression, decision trees, support vector machines, and neural networks. In the training phase, the model acquires knowledge from the training data by modifying its parameters to reduce the discrepancy between its predictions and the real results. The testing phase assesses the model's capacity to generalize by evaluating its performance on fresh and unseen data. Metrics are utilized to quantify the effectiveness of a machine learning model. Typical metrics used vary depending on the nature of the task, such as classification or regression. These metrics commonly include accuracy, precision, recall, F1 score, and mean squared error. When a model becomes too close to the training data, including noise and outliers, it fails to apply this knowledge to new, unknown data [20, 112]. Underfitting refers to the situation when a model is overly simplistic and fails to accurately capture the underlying patterns in the data. This leads to subpar performance on both the training and testing sets. The process of choosing, modifying, or developing additional features from the raw data to improve a model's performance [82].

Machine learning is utilized in many fields such as image and speech recognition, natural language processing, recommendation systems, healthcare, finance, and numerous other sectors. The area is constantly progressing due to breakthroughs in algorithms, hardware, and the accessibility of extensive datasets.

Machine learning can be classified into three primary categories:

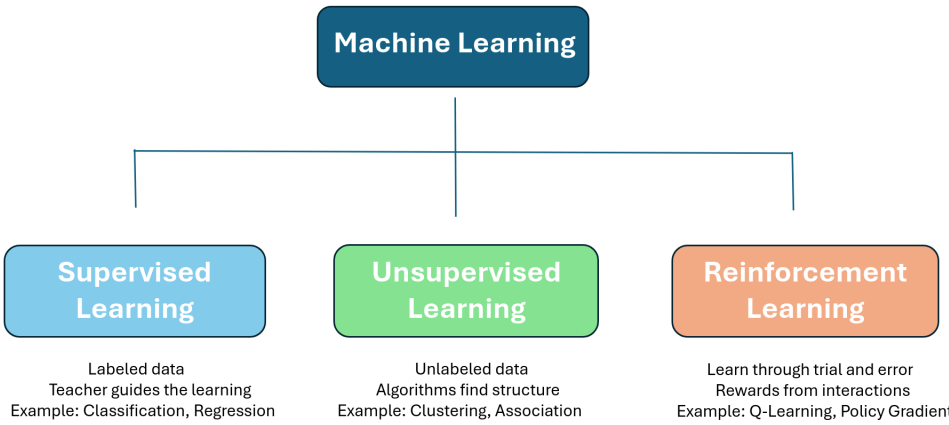

Figure 2.1: Types of machine learning.

- Supervised Learning: Supervised learning involves training an algorithm using a dataset that is labeled, meaning each input is paired with its corresponding accurate output. The model acquires the ability to establish a relationship between the given inputs and outputs, enabling it to generate predictions for novel, unobserved data [36].

- Unsupervised Learning: Unsupervised learning pertains to the analysis of unlabeled data, in which the algorithm must discern patterns or relationships without explicit instructions. Clustering and dimensionality reduction are frequently encountered tasks in unsupervised learning [27].

- Reinforcement Learning: Reinforcement learning entails an autonomous agent ac-

quiring the ability to make decisions through iterative interactions with its environment. The agent is provided with feedback in the form of rewards or penalties, which helps it to learn and adopt optimal behavior [88].

2.2.1 Utilize Machine Learning in Cybersecurity

Machine learning (ML) is a branch of artificial intelligence (AI) that empowers computers to learn and improve from experience without being explicitly programmed. It involves the development of algorithms that enable systems to automatically learn and make predictions or decisions based on data. At its core, ML algorithms leverage statistical techniques to identify patterns and relationships within datasets, allowing them to generalize from past examples and make accurate predictions on unseen data.

In the context of cybersecurity, machine learning plays a crucial role in enhancing threat detection, risk assessment, and anomaly detection. By analyzing vast amounts of data, ML algorithms can identify patterns indicative of malicious activity, enabling early detection and mitigation of security threats. ML techniques are particularly effective in cybersecurity due to their ability to process large-scale data in real-time and adapt to evolving threats.

We summarize the main cybersecurity application areas that people can utilize ML to enhance cybersecurity defense and effectiveness:

- Threat Detection: ML algorithms can analyze network traffic, system logs, and user

behavior to identify abnormal patterns indicative of security threats such as malware infections, intrusion attempts, and unauthorized access.

- Anomaly Detection: ML models can learn the normal behavior of systems and users and detect deviations from this baseline, signaling potential security breaches or anomalies. This approach is particularly useful for detecting zero-day attacks and insider threats.

- Predictive Analysis: ML algorithms can predict potential security incidents based on historical data and ongoing trends, enabling proactive risk management and mitigation strategies.

- Vulnerability Management: ML techniques can be used to identify vulnerabilities in software and systems by analyzing code, network configurations, and historical data, helping organizations prioritize and remediate security weaknesses.

- Behavioral Analysis: ML algorithms can analyze user behavior, application usage patterns, and system interactions to identify suspicious activities and detect advanced persistent threats (APTs) that evade traditional security measures

2.3 Encrypted Network Traffic Analysis And Classification

Traffic analysis, a methodical examination of network activity, plays a vital role in various domains, offering insights into patterns and anomalies within network traffic [4]. However, the advent of network traffic encryption has posed formidable challenges to traditional analysis methods, rendering plaintext payload extraction less effective [10, 18]. Machine

learning emerges as a potent solution, capable of extracting valuable insights from encrypted traffic analysis without accessing content [19]. This paradigm shift allows for the development of sophisticated algorithms that discern patterns and classify encrypted traffic accurately, overcoming the limitations posed by encryption protocols [21].

From the inception of the internet, identifying devices through network traffic classification has piqued interest, with research extending to both WiFi and ethernet traffic. Such studies have demonstrated that traffic classification can pinpoint a variety of information, encompassing IoT devices and mobile application activities [106], with previous investigations primarily focusing on traffic as observed by an insider, capturing TCP/IP packets. This method exposes distinct network features of the device, such as using the destination port number as a unique identifier for different IoT devices, as noted in [94]. Additionally, timing aspects like packet interarrival times have been highlighted as sufficient for deep learning-based device fingerprinting [17]. These techniques, however, are not applicable for adversaries outside the network due to IP traffic being encapsulated at a higher layer, obscuring key network features. Our approach differs by leveraging features accessible to an observer outside the WiFi network. Recent advances in defending against website fingerprinting [1] and data plane intrusion detection [68] have shown promising results, indicating a growing interest in enhancing network security through advanced techniques. Studies on rogue access point detection [56, 57], deep learning-based eavesdropping attacks [31], investigating the effect of traffic sampling on machine learning-based network intrusion detection approaches [11], and a scalable and dynamic ACL system for in-network defense [61] further underscore the importance of continuous research in securing wireless networks. AutoDefense, a reinforcement learning-based autoreactive defense

against network attacks [81], and adversarial learning attacks on graph-based IoT malware detection systems [2] also contribute to the evolving landscape of network security. Additionally, our work is related to hardware fingerprinting and device classification research. Hardware fingerprinting focuses on clock skew measurements for device authentication, as discussed in [111], prioritizing hardware traits over device-specific characteristics. This focus is less ideal for identifying individual devices. Conversely, other efforts aim to categorize devices into broad types [112], such as power or sensor-based devices, which, while useful for understanding device functionality, lack the precision required for differentiating devices with similar operational traits and traffic profiles.

The research most similar to our proposed WiFi profiling attack is by Acar et al. [3], which identified devices, their states, and user activities through traffic analysis across WiFi, Zigbee, and Bluetooth, suggesting traffic spoofing as a defense but they not showing the machine learning results to proof the experiment and ensure the model performance. Their method, however, relied on a rogue access point to gather WiFi data, implying an in-network perspective, akin to other studies [7, 94, 16, 17], which necessitate physical or sophisticated means to access the target's encrypted network. In contrast, we assume a scenario with an out-of-network adversary, who can monitor WiFi traffic without needing to infiltrate the network.

Statistical techniques for detecting cyberattacks on computer networks based on an analysis of abnormal traffic behavior" published in the International Journal of Computer Network and Information Security in 2020. This paper likely discusses various statistical methods used to detect cyberattacks by analyzing abnormal traffic behavior on computer

networks.

In the context of ”Encrypted Network Traffic Analysis and Classification,” this reference could be relevant in several ways. First, it may discuss statistical techniques used in the analysis of network traffic, including encrypted traffic. These techniques could involve identifying patterns or anomalies in encrypted network traffic to detect potential cyberattacks or security threats.

Furthermore, the paper might delve into how the encryption of network encrypted traffic poses challenges for traditional methods of network traffic analysis and classification. Encrypted traffic can obscure important information, making it difficult to detect malicious activities or classify network traffic accurately. Therefore, understanding statistical techniques for analyzing encrypted traffic and detecting abnormalities becomes crucial in ensuring network security.

Overall, this reference could provide insights into statistical methods used for analyzing encrypted network traffic, detecting cyberattacks, and how the insecurity introduced by encryption affects the analysis and classification of network traffic.

An essential component of network security and monitoring is the analysis and classification of encrypted network traffic, particularly since encryption is increasingly used to safeguard sensitive information [6]. Encryption guarantees the secrecy of data during transmission, but it also presents difficulties for network security monitoring as it obscures the payload. Analyzing encrypted traffic is vital for detecting malicious actions, discovering anomalies, and assuring the overall security of a network [22]. Despite the encryption

of the payload, it is still possible to analyze metadata such as packet size, timing, and source/destination information. Identify recurring patterns within encrypted metadata that may suggest specific activities or protocols. In specific situations, organizations implement SSL/TLS interception procedures that utilize a trustworthy certificate to decode and examine encrypted traffic [112]. This strategy gives rise to privacy concerns and may face opposition due to possible ethical and legal complications.

Conduct an analysis of encrypted traffic to identify any variations from the usual patterns of behavior. Utilize machine learning algorithms to detect harmful behavior by analyzing historical data and behavioral patterns. Employ signature-based detection to recognize patterns linked to established threats, even when they are concealed within encrypted data. Formulate heuristics to detect anomalies or suspicious patterns in encrypted traffic. Analyze current traffic patterns in relation to established baselines in order to identify any abnormal behaviors. Endpoint Detection and Response (EDR) refers to the capability of endpoint security systems to monitor and analyze encrypted traffic at the endpoint. This allows for the identification of possible threats and the gathering of valuable information. Utilize behavioral analytics to identify anomalous behaviors on endpoints. Utilize DPI to scrutinize the contents of encrypted packets [111, 58, 103, 8].

Deep Packet Inspection (DPI) can be utilized to detect and identify harmful payloads or suspicious activity occurring within encrypted network traffic. Engage in the exchange of information: Take part in groups focused on sharing threat intelligence to be informed about emerging threats and methods of assault. Engage in collaboration with other organizations to strengthen the joint defense against emerging dangers. It is important to

note that the efficacy of encrypted traffic analysis depends on the use of various methods and technologies. Organizations must diligently maintain a delicate equilibrium between the imperative for security and the preservation of user privacy and adherence to legal requirements. Furthermore, it is crucial to be updated on the most recent advancements in encryption and decryption technology in order to effectively respond to ever-changing security risks [39, 10, 18, 19].

2.4 Encrypted Traffic Analysis for Three Major Applications

In current cyber world, there are many applications involve encrypted network traffic and have needs to be analyzed and classified by their stakeholders. However, from the perspective of technological merit and broader impact, in this paper we focus on providing analysis and survey of three major applications: Internet-of-Things, mobile devices and applications, and web applications.

The IoT has transformed our interaction with the physical world by establishing connections between devices, sensors, and systems over the internet [110]. The interlinked network produces huge quantities of data, which may be utilized and examined using sophisticated technologies like deep learning and machine learning. IoT devices are equipped with a multitude of sensors that provide real-time data. Deep learning models can be utilized to process and analyze the sensor data, deriving significant insights and patterns.

Machine learning algorithms can anticipate the potential failure or maintenance needs of IoT devices by analyzing past data. Implementing this proactive strategy aids in minimiz-

ing periods of inactivity and enhancing the overall effectiveness of IoT implementations. Deep learning models have the capability to analyze user behavior, preferences, and historical data in order to customize website content according to individual users.

This improves the user experience by providing customized suggestions and material to each user. Websites can utilize NLP models to comprehend and provide more natural responses to user inquiries. These applications can be implemented in chatbots, customer support services, and other interactive elements, hence enhancing the user-friendliness of websites [108]. Image and Video Processing: Deep learning demonstrates exceptional performance in the field of image and video recognition. Applications can leverage these functionalities for tasks like facial recognition, object detection, and content tagging, hence improving user engagement and security [89, 67].

Machine learning algorithms have the capability to recognize atypical patterns or behaviors in application data, aiding in the identification of potential security concerns, fraudulent activities, or system faults. Anonymization and Encryption: Machine learning can improve data security by using advanced techniques like anonymization and encryption. This guarantees the security of sensitive information that is exchanged between IoT devices, websites, and applications.

Machine learning models can utilize user behavior patterns to detect anomalous behaviors, facilitating the timely identification of security breaches. The proliferation of AI technology has led to heightened apprehensions over data privacy and ethical implications. Implementing stringent privacy safeguards and strictly adhering to ethical norms are crucial for employing deep learning and machine learning in IoT, websites, and applications [17, 12].

Scalability is crucial in deep learning and machine learning architecture to accommodate the increasing volume of data.

2.4.1 Internet of Things

The IoT encompasses a diverse range of interconnected objects, from industrial systems to household devices, interconnected through embedded electronics and software, facilitating data exchange and analysis among these entities [25]. Emerging IoT applications span various sectors, including healthcare, agriculture, and military, presenting new challenges in device management, data handling, and security [30]. Extensive research addresses architecture, communication protocols, and security mechanisms to ensure the successful commercialization of IoT technologies [32], with the utilization of empowering technologies like cloud computing and edge computing amplifying both opportunities and risks [9, 48], necessitating robust security measures to safeguard against potential threats.

The advent of encrypted network traffic poses significant challenges for analysis and classification, but leveraging machine learning techniques offers promising solutions. The (IoT) plays a pivotal role in this landscape, facilitating vast data generation from interconnected devices and systems. Machine learning, particularly deep learning models, proves invaluable in processing and extracting insights from the sensor data produced by IoT devices, aiding in anomaly detection and predictive maintenance [55]. Furthermore, machine learning algorithms can enhance user experience on IoT-enabled websites by personalizing content based on user behavior and preferences, while natural language processing models improve user interaction [66, 70]. In the realm of image and video processing, deep

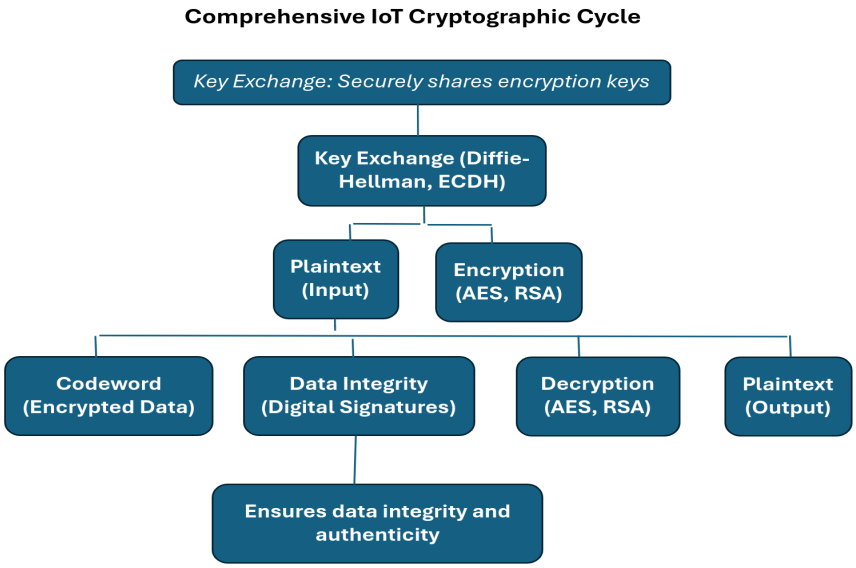

Figure 2.2: IoT crypto system shows how plain text is converted into codeword after using cryptographic algorithms.

learning excels in tasks like facial recognition and object detection, enhancing security and user engagement [74]. Additionally, machine learning techniques such as anonymization and encryption play crucial roles in ensuring data security, particularly in the exchange of sensitive information between IoT devices, websites, and applications [76, 78, 80]. As the volume of data continues to grow, scalability becomes imperative for deep learning and machine learning architectures to effectively analyze encrypted network traffic and classify patterns, ensuring robust security measures in an increasingly encrypted landscape.

2.4.2 Mobile Devices and Applications

Since the invention of iPhone, mobile devices and mobile applications have quickly grasped both our daily lives and the world of business. Mobile devices remove the physical barriers between any people, connecting anyone instantly through calls, texts, video chats, and social media apps, etc. They enable Internet access and information gathering at any places or time, significantly enhancing people's daily life and business transactions. Because of such importance of mobile devices and applications, attackers have developed many ways to extract information and user privacy via captured mobile devices' encrypted traffic. Such attacks become more challenging since attackers can easily eavesdrop the network traffic between mobile devices and the Internet via Wifi or cellular wireless channels.

B. Ahlgren. [5] showcased how encrypted traffic patterns analysis can discreetly deduce sensitive personal information, even over encrypted WiFi networks, without requiring network access credentials . While achieving a commendable recall rate of 86%, indicating potential inadvertent disclosure of personal information by applications, the experiment also reveals room for improvement, with a significant false positive rate. Furthermore, the techniques employed for app identification extend to fingerprinting other encrypted communications like VoIP and website visits, making mobile apps particularly susceptible targets due to their transparent ranking system and simplicity of data collection [95]. Notably, these techniques, demonstrated using standard 802.11g WiFi, are anticipated to be applicable across various wireless communication protocols, including long-distance protocols like 4G LTE used in cellular networks [97]. The scenario examined involves a passive observer seeking to deduce information about users connected to a WiFi Access

Point (AP), highlighting the potential implications of encrypted network traffic analysis utilizing machine learning in uncovering sensitive information without compromising network security.

2.4.3 Website Fingerprinting

López et al. [79] demonstrated that an assailant can discern the specific webpage a client is visiting through the use of a classification algorithm, which processes observed packet sequences. The crux of the classification lies in the concept of distance between packet sequences, where a greater distance suggests that the sequences are less likely to originate from the same webpage. Since then, various metrics for calculating distance have been employed by different researchers, including comparing the frequency of unique packet lengths or adaptations of the Levenshtein distance. The chosen distance metric reflects the approach to utilizing features to differentiate between webpages. These features are derived, whether directly or indirectly, from the packet sequences to enable comparative analysis.

The foundational insight presented by [98] showed that webpage category is inherently multi-modal. Numerous variables can induce variations in a webpage: the network's state, random changes in advertisements and content, updates over time, and the unpredictable sequence of resource loads. Even the client's setup might influence how the page is rendered. To manage such multi-modal datasets, an attacker should amass sufficient data to cover representative samples from each variation. For instance, an attacker might collect data reflecting two different loading scenarios of a webpage: one under low-bandwidth

and another under high-bandwidth conditions. The classifier we employ is tailored to handle multi-modal classes, allowing for the different modes within a category to be unrelated to each other.

Another category of website fingerprinting attack is based on resource size. The HTTP 1.0 specification mandates that individual resources of a web page, such as images, scripts, etc., be fetched over distinct TCP connections. This allows an attacker who can distinguish between these various connections to deduce the total size of each resource. The initial research into this type of vulnerability was conducted in the era of HTTP 1.0, with Cheng et al. in 1998 [29], Sun et al. in 2002 [104], and Hintz in 2003 [52] demonstrating that knowledge of resource sizes could facilitate webpage identification. However, the subsequent HTTP 1.1 standard, which employs persistent connections, alongside advancements in browsers and privacy-protection technologies, have rendered such resource size based attacks less effective against more recent systems.

2.4.4 Procedure of Machine-Learning-Based Classification of Network Traffic

In this section, we systematically introduce the major steps and the procedure in network traffic analysis and classification by utilizing the machine-learning method. The high-level architecture is shown in Figure 3.9 [35, 109, 23, 42, 44, 47]. In the following, we explain each block of the procedure in detail.

$Data\ collection$: The landscape of Internet traffic is constantly changing due to the advent of novel forms of traffic, devices, and applications. Hence, it is necessary to gather well-recognized Internet traffic while taking into account emerging trends in dealing with un-

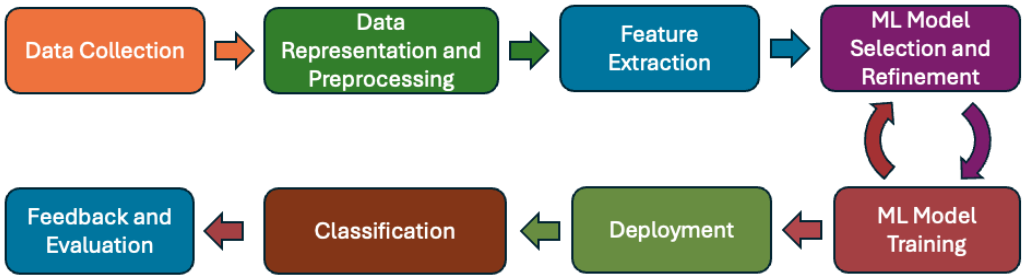

Figure 2.3: Procedure of typical encrypted network traffic analysis and classification based on machine learning.

known traffic. Much earlier work on traffic classification relies on capturing non-encrypted network traffic, which is not suitable for analyzing encrypted traffic for most Internet applications presently.

Lalmuanawma et al. [73] showed that most of the accessible public datasets lack labels or were labeled using unreliable techniques, such as deep packet inspection, which is ineffective with encrypted traffic, or port-based labeling, which is ineffective with dynamically allocated ports.

Data collection could have very different formats depending on where it is collected, such as the data-link frames captured on an encrypted WiFi channel or the encrypted IP packets captured on Ethernet or routers. Data collected could also be complete when capturing 100% of traffic going through or being sampled via various sampling methods.

Data representation and preprocessing: Various techniques have been employed to depict traffic. Moore et al. [83] present an extensive collection of characteristics derived

from the packet size and inter-arrival time, TCP flags, port numbers, and IP addresses. Additional approaches take into account the domain names and the contents of protocol requests. Time series features are utilized for precise classification at a detailed level. An alternative method being suggested is the depiction of the patterns of communication between the entities involved using unconventional formats.

Images have recently been used to depict traffic flows. The selection of data format is determined by the classification algorithm employed. For instance, the time series presentation is employed for classification based on Markov Models, while the image-based representation is utilized for classification based on deep learning, among other methods [77].

Before the network traffic data can be analyzed, it also needs to be preprocessed carefully. Data preprocessing is necessary because it will help clean up messy and/or inconsistent data, reduce data dimension and normalize data scale to improve analysis efficiency, reduce overfitting and bias in training, etc. [54] In summary, data preprocessing acts as an indispensable foundation for successful machine learning. It ensures the data are clean, prepares them for the model's learning process, and ultimately improves the accuracy, efficiency, and generalizability of ML predictions and classifications.

$Feature\ extraction$: In machine learning, feature extraction is the process of transforming raw data into a set of more relevant and informative features. Feature extraction helps identify and extract the key features that actually matter for the prediction task by machine-learning models. It will greatly reduce data complexity and dimensions, selecting only those input dimensions that contain the most relevant information for solving the particu-

lar problem.

Khalid et al. [65] provided a good survey on feature selection and extraction techniques in machine learning. The research conducted by Hakak et al. [45] is an example of work showing the importance of feature extraction techniques in identifying fake news more efficiently.

$ML\ model\ selection\ and\ refinement$: After data have been collected and processed, the next step is to use the right machine-learning model so that we can achieve the best traffic analysis and classification performance. Choosing the right machine-learning model is crucial for making the most of our data. Numerous ML models have been presented so far and utilized in various fields. Factors to consider include problem type, data characteristics, model complexity, interpretability, availability of resources, etc. [91]

ML model selection should be an iterative process that requires us to continuously refine our selected model and its parameter settings to better fit the target application and resources available. Therefore, as shown in Figure 3.9, this step has an iterative loop with the next step of 'ML model training'.

$ML\ model\ training$: Machine-learning training is the process of feeding data into a machine-learning model to help it learn and improve its ability to make predictions or classifications. The data are typically divided into three sets: the training set used to train the model, the validation set used to monitor performance during training and prevent overfitting, and the test set used for final evaluation after training is complete. Two main factors need to be considered in model training. First is the computational resource

needed for training and the resource available for future real-world deployment. Second is the potential bias and fairness issues, especially when ML is used in social-impactful applications.

$Deployment$: Deployment is the process of making a trained machine-learning model accessible and operational in the real world, where it can be used to make predictions or classifications on new, unseen data. For network traffic analysis and classification, it can be deployed in a cloud platform on received real-time monitored data, on the premises if an organization deploys it on its own server due to various concerns, or on edge devices with limited connectivity or for strict real-time requirements [87].

$Classification$ For encrypted network traffic analysis, classification means to discover more insight information based on captured unknown encrypted traffic, such as the IoT device type or even working status, what applications generate the captured traffic, whether the encrypted data are audio, video streaming, or images, etc.

$Feedback\ and\ evaluation$: The classification results given by the machine-learning model can be further evaluated or verified based on either the classification outcome via other ways or human manual inspection. This feedback and evaluation will help users to understand better what are the potential weaknesses or problems with the applied machine-learning system and then redesign or refine all the previous procedure steps in the whole process shown in Figure 3.9 to improve classification performance.

2.4.5 Network Traffic Processing and Inspection

This section provides an overview of common methods and techniques for processing and inspecting network traffic that have been extensively utilized in the literature throughout the years. This section aims to facilitate the reader's understanding of the process of network traffic processing and to highlight the issues that have arisen due to network encryption. Anomaly detection is a versatile technique that may be used in various fields, including network intrusion detection, to identify abnormal or suspicious activity and behavior.

A strategy based on anomaly detection estimates a model of "normal network activity," and evaluates future activity on the network by comparing its probability to the learned model. By employing this method, it becomes feasible to differentiate between a usual and potentially harmful network activity as opposed to one that is harmless. This categorization is founded on specific heuristics or principles and seeks to identify instances of misuse or atypical conduct.

In order for an anomaly-based intrusion detection system to accurately detect suspicious traffic, it is essential for the system to be trained to understand typical system behavior. For instance, there are several ways available for detecting volumes. These techniques are used to monitor the traffic load of a network in order to detect abnormalities that cause major changes in traffic volume, such as flooding attacks.

Feature-based anomaly detection seeks to address the limitations of volume-based solutions by analyzing various characteristics of network data. The bulk of anomaly detection

systems typically involve two phases, which are training and testing [34, 15, 38, 37].

The first part involves constructing a profile of usual behavior, whereas the testing phase involves comparing current traffic with the model generated during the training phase. Machine or deep learning techniques are mostly employed to detect anomalies. Nevertheless, anomaly detection frequently encounters challenges such as (i) elevated rates of false positives, (ii) the challenge of acquiring dependable training data, and (iii) the persistence of training data and the changing behaviour of the system (Fig 5).

Shone et al [101]. assert that the dynamic characteristics of contemporary networks, including the continuous growth in traffic speed, volume, and variety, inevitably lead to a lack of a comprehensive and universally accepted solution for detecting anomalies.

The multitude of protocols and the variety of data in modern connections make it more challenging to distinguish between normal and abnormal behavior. This complexity hinders the establishment of an accurate baseline and expands the potential for exploitation or zero-day attacks. While anomaly-based detection approaches can be highly efficient in the field of network security, signature-based techniques have the capability to analyze and examine a broader range of applications.

Signature-based network inspection is a method used to classify and characterize traffic, primarily through the use of deep packet inspection (DPI) techniques. DPI serves as a fundamental element in various network systems, including traffic monitors, classifiers, packet filters, network intrusion detection, and prevention systems.

Different layers of the OSI architecture utilize Deep Packet Inspection (DPI) in various

network components [41, 43, 46, 49]. The OSI model layers are presented in Fig 6. In contrast to the initial stages of packet inspection, where DPI was limited to examining packet headers (such as proxies and firewalls), the present scenario necessitates the inspection of packet content across all layers of encapsulation due to the increased complexity and obfuscation of protocols.

Governments, Internet and Communications Service Providers (ISPs/CSPs), and other organizations largely depend on Deep Packet Inspection (DPI) technology to accurately monitor and analyze network traffic [51, 53].

Utilizing DPI can enhance the quality of service (QoS) by discerning various types of content and streaming them in a tailored manner. Similarly, through the analysis of the contents of the incoming packets, one can detect and scrutinise suspicious and malicious activity.

In the field of network security, the use of signature-based network inspection techniques can be highly effective if security software developers are already aware of similar assaults. As previously said, DPI is the fundamental process used in typical applications within the field of network security, analytics, and other related areas. The need for advanced traffic analysis and the ongoing rise in network speeds have prompted extensive research to continuously develop innovative Deep Packet Inspection (DPI) methods.

Network intrusion detection systems have evolved into highly effective instruments for network administrators and security specialists in recent decades, aiding in the identification and prevention of a broad spectrum of threats. Snort and Suricata are widely used

network intrusion detection system (NIDS) solutions that employ pattern matching and regular expressions to analyze network data. In contrast, Zeek/Bro utilizes scripts to facilitate automation, making it more convenient. The research community has also made endeavours to enhance the efficiency of NIDS by utilizing either standard hardware, such as graphics processing units (GPUs) and parallel nodes, or dedicated hardware, such as ternary content-addressable memories (TCAMs), application-specific integrated circuits (ASICs), and field-programmable gate arrays (FPGAs).

Nevertheless, these works have the capability to analyze network traffic that lacks encryption, as they extract significant data from the content of network packet payloads. Traffic classification and network analytics play a crucial role in ensuring quality of service (QoS) at both central network traffic intake points and end-host computers. To precisely identify the incoming traffic, a detailed analysis should be conducted, examining both the packet header and payload.

Most traffic categorization methods primarily focus on flow-based strategies, which try to categorize flows based on the application that generates them, rather than individual packets. Several studies have suggested several techniques for determining the application linked to a traffic flow [60, 62, 64, 69, 71].

2.5 Strategies in Analyzing and Classifying Encrypted Network Traffic

Encrypting network traffic is crucial for ensuring data privacy and security, but it also poses challenges for analyzing and classifying that traffic for various purposes such as

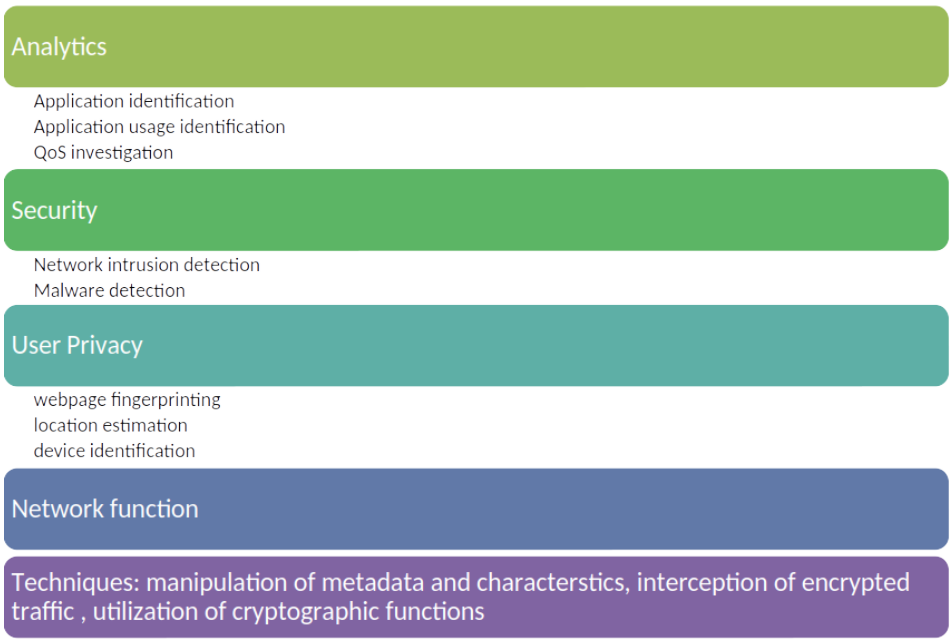

Figure 2.4: Encrypted traffic analysis and inspection phases.

network management, security monitoring, and traffic optimization. However, there are approaches to analyze and classify encrypted network traffic even without decrypting it fully, leveraging machine learning and deep learning techniques.

In the following, we summarize strategies that can be utilized to perform effective analysis and classification of encrypted network traffic for various purposes while preserving data privacy and security.

- Feature Extraction: Even though the payload of encrypted traffic is not accessible, features can still be extracted from the encrypted data. These features could include packet size, inter-arrival times, packet direction,

protocol headers, and statistical properties of encrypted payloads.

- Dimensionality Reduction: High-dimensional feature spaces can be reduced using techniques like Principal Component Analysis (PCA) or autoencoders to capture the most relevant information while discarding noise.
- Model Training: Various machine learning algorithms such as Random Forests, Support Vector Machines (SVM), or neural networks can be trained on the extracted features to classify different types of network traffic.
- Deep Learning Approaches: Deep learning models, especially Recurrent Neural Networks (RNNs) or Convolutional Neural Networks (CNNs), can be effective in learning patterns and representations from sequential or spatial data, which can be applied to encrypted network traffic analysis.
- Transfer Learning: Pre-trained deep learning models on related tasks such as natural language processing or image classification can be fine-tuned for encrypted network traffic analysis, leveraging the learned representations.
- Anomaly Detection: Unsupervised learning techniques can be applied to detect anomalies in encrypted traffic by learning the normal behavior of the network and flagging deviations from it.
- Traffic Classification: Supervised learning techniques can be used to

classify encrypted traffic into categories such as IoT devices, websites, or apps based on their behavioral patterns and communication protocols.

- Ensemble Methods: Combining multiple models or classifiers through ensemble methods like bagging or boosting can improve classification accuracy and robustness.
- Privacy Considerations: It's crucial to ensure that the analysis of encrypted traffic respects user privacy and regulatory requirements. Techniques like differential privacy or secure multiparty computation can be explored to achieve this.
- Continuous Learning: Given the evolving nature of network traffic patterns and security threats, continuous learning mechanisms should be employed to adapt the models to new data and emerging trends.

2.5.1 Limitations

Networking services such as asymmetric routing, NATing, and tunneling have an impact on the performance of traffic classifiers. Hence, it is imperative to account for these network dynamics while constructing a traffic classifier. The features must be selected in a manner that ensures they are not influenced by any of these network functions [84].

The curse of dimensionality refers to the challenge of efficiently classifying data, particularly in real-time processing, when speed is crucial. It is crucial to thoroughly analyse the time and processing complexity aspects of data preparation, data representation or feature

computation, and classification computation overhead. Within this particular framework, three primary factors are crucial: memory capacity, computational intricacy, and processing duration. Developing a resilient yet precise classifier is crucial, yet efficiency is essential in certain scenarios, particularly for time-sensitive network services (such as intrusion detection) [85].

Actual implementation: Despite the substantial research on ML traffic categorization, there are only a limited number of classification frameworks/tools available. Despite privacy issues, DPI continues to be widely employed. Furthermore, technical obstacles such as traffic velocity and large-scale data could potentially hinder the practicality of capturing and analyzing this traffic. Additionally, a constraint in real-world implementation is the need to continuously train the models, as well as adjust for new or unfamiliar traffic. Additionally, the optimization of model parameters, taking into account specific network features like as speed and fragmentation, may impose limitations on the performance of implemented models [96].

Obfuscation: The propriety of classifying and obfuscating information is a subject of controversy. From a privacy standpoint, classification is regarded as an intrusion that undermines user privacy. Nevertheless, in terms of network administration, attackers can employ obfuscation techniques to evade detection of their attacks. Classification is preferred by security and QoS applications in this situation, whereas obfuscation is favored by privacy [90].

Traffic sampling: The process of collecting a representative subset of network traffic data for analysis and monitoring purposes. An obstacle to the implementation of traffic classi-

fication applications is the need for high-speed capabilities in the core network. Due to the impracticality of extracting features from packets at a very high speed, traffic sampling is used as an alternative. Modifying traffic characteristics and statistical data can potentially detrimentally affect the accuracy of classification [93].

As shown in Table 2.1, our survey provides a comprehensive overview of machine learning-based methods for analyzing and classifying encrypted network traffic.

2.6 Summary and Recommendations

Summary: In this chapter, we have provided a comprehensive overview of utilizing machine learning techniques for encrypted network traffic analysis and classification. The growing prevalence of encryption poses significant challenges for traditional traffic analysis methods, necessitating the adoption of advanced techniques capable of extracting insights from encrypted data without compromising privacy and security.

Through this work, we have demonstrated the effectiveness of machine learning approaches in addressing these challenges. By leveraging techniques such as feature extraction, dimensionality reduction, anomaly detection, and traffic classification tailored for encrypted scenarios, it is possible to gain valuable insights into network traffic patterns and behaviors, even in the presence of encryption.

However, it is crucial to acknowledge the limitations and challenges associated with encrypted traffic analysis. The curse of dimensionality, obfuscation techniques, privacy con-

siderations, and the need for continuous learning must be carefully addressed to ensure the practical implementation and long-term effectiveness of these solutions.

Recommendations: Based on the findings of this work, we recommend the following:

1. Continued research and development in machine learning techniques specifically designed for encrypted traffic analysis, with a focus on improving accuracy, efficiency, and scalability.

2. Collaboration between academia, industry, and regulatory bodies to establish guidelines and best practices for ensuring user privacy and data security while enabling effective network monitoring and security.

3. Exploration of privacy-preserving techniques, such as differential privacy and secure multiparty computation, to enable encrypted traffic analysis while maintaining strict privacy guarantees.

4. Adoption of continuous learning mechanisms and adaptive models to keep pace with the ever-evolving landscape of network traffic patterns, security threats, and encryption technologies.

5. Integration of machine learning-based encrypted traffic analysis solutions into existing network security frameworks and tools, enabling more comprehensive and effective threat detection and mitigation.

6. Interdisciplinary research efforts combining expertise from fields such as computer science, cybersecurity, machine learning, and data privacy to holistically address the

challenges of encrypted traffic analysis.

By embracing machine learning techniques and proactively addressing the associated challenges will enable robust, privacy-preserving encrypted network traffic analysis, paving the way for enhanced cybersecurity, network optimization, and secure adoption of technologies involving IoT devices, mobile applications, and website fingerprinting.

Survey Paper	Year	Description
Unencrypted Traffic		
Buczak *et al.* [24]	2016	Introducing data mining and machine learning techniques for cybersecurity intrusion detection
Jing *et al.* [59]	2018	Reviewing security data and analytical methods for detecting DDoS and Worm attacks
Fernandes *et al.* [40]	2019	Summarizing network data types and techniques for anomaly detection
Kwon *et al.* [72]	2019	Examining deep learning methods applied to network anomaly detection
Encrypted Traffic		
Velan *et al.* [105]	2015	Summarizing approaches for analyzing encrypted traffic, mainly focusing on traditional machine learning methods
Rezaei *et al.* [92]	2019	Reviewing deep learning techniques for the classification of encrypted traffic
Conti *et al.* [33]	2018	Reviewing studies that focus on network traffic analysis targeting mobile devices
Pacheco *et al.* [86]	2018	Introducing machine learning solutions for network traffic classification
Shen *et al.* [100]	2023	Providing a comprehensive survey on machine learning-powered encrypted network traffic analysis, including traffic classification, anomaly detection, and feature extraction techniques
Our Survey	2024	Providing a comprehensive survey on machine learning-based methods for analyzing encrypted traffic, covering goals such as network asset identification, network characterization, privacy leakage detection, and anomaly detection in IoT, website, and mobile application

Table 2.1: Comparison of existing surveys with our work.

CHAPTER 3: Detecting IoT Devices by Monitoring Encrypted Wireless Network Traffic

This chapter presents a novel technique for identifying Internet of Things (IoT) devices by passively monitoring encrypted wireless network traffic from outside the target network. The main contribution is the development of an effective method that allows an adversary to profile and detect various IoT devices, as well as discern their operational states (active or idle), without requiring any access to the network itself. This is achieved by capturing and analyzing patterns in the encrypted WiFi traffic metadata, such as packet sizes and flow characteristics, and applying a data summary machine learning algorithms to classify the devices based on their unique traffic signatures.

By leveraging machine learning techniques on encrypted traffic metadata, this work unveils a significant vulnerability in IoT ecosystems, where adversaries can covertly gather sensitive information about the types and operational states of devices within a target network. The findings emphasize the need for enhanced security measures and privacy-preserving mechanisms to mitigate such passive monitoring attacks on encrypted wireless networks.

3.1 Threat Model and Assumptions

Figure 3.1 illustrates how an adversary can passively monitor wireless traffic from a target's WiFi router or access point (AP) without connecting to the network, simply by being

within the AP's transmission range. The adversary might use wardriving, warcycling, or warwalking, employing a non-intrusive sniffing tool to discreetly capture nearby WiFi network traffic. This method ensures no active network penetration.

The adversary's goal is to gather information about connected devices, such as the number, types (e.g., light bulbs, smart TVs, laptops), and operational states (idle or active). This monitoring can unintentionally reveal sensitive information, such as household or business size and economic status. Identifying device types may also uncover vulnerabilities in certain IoT devices, posing potential security threats. This passive surveillance is especially concerning for IoT devices, which often lack robust security measures and may serve as entry points for broader network attacks. Understanding these vulnerabilities is crucial for developing effective cybersecurity strategies.

3.2 Capturing of Out-of-Network Encrypted WiFi Traffic

Illustrated in Figure 3.9 is the attacker's presumed system setup, which is composed of two primary components: the initial training for device profiling (offline phase) and the subsequent device recognition process (online phase). During the offline stage, the attacker connects various IoT devices to a WiFi gateway. Using sniffing tools, the network traffic is captured and the data is subsequently categorized by device name based on their MAC addresses. The collected data undergoes preprocessing to eliminate extraneous information such as irrelevant network traffic, beacon, and broadcast frames, followed by the distillation of pertinent features into a csv format ready for supervised learning (detailed further

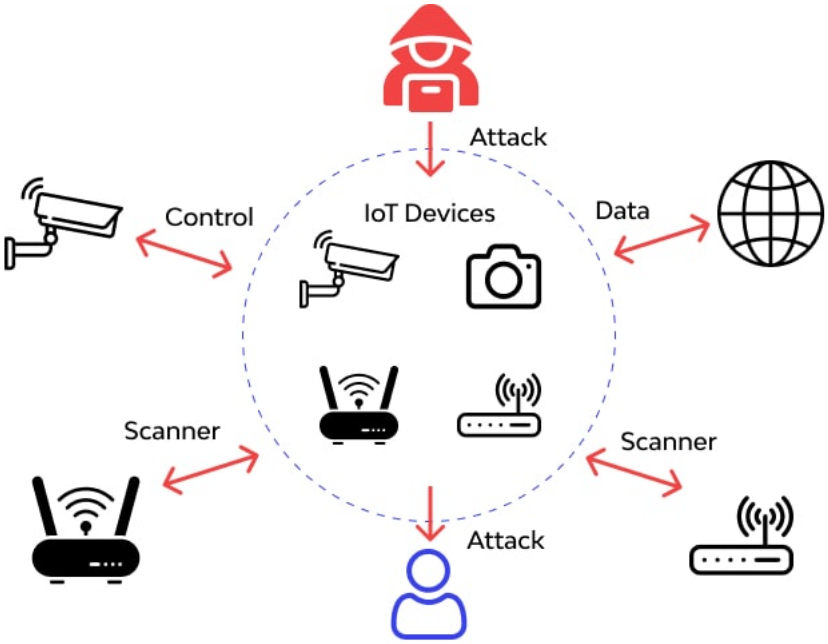

Figure 3.1: Threat model.

in Section 3.3). These features are then fed into various machine learning algorithms to determine the most accurate model for later real-time identification (online phase).

In the online stage, the attacker briefly captures network traffic from the target's AP, typically for a period like 30 seconds, and replicates the preprocessing done during the offline phase. This step, elaborated on in Section 3.2.2, involves no prior knowledge of the devices but employs statistical and standard filtering to remove noise unrelated to data patterns. Features are extracted from the cleaned data using a Python script.

The final step employs the pre-trained model to ascertain the type of each device and its operational status. The forthcoming sections will delve into the details of each phase in

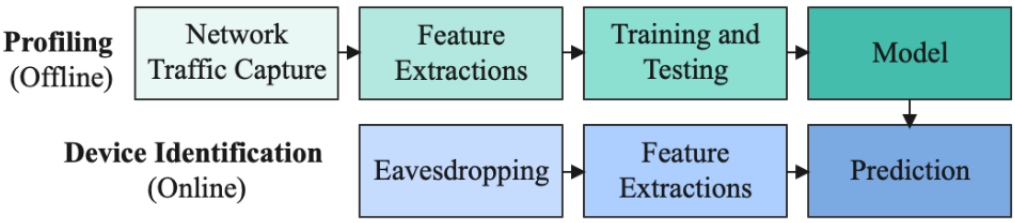

Figure 3.2: IoT device profiling attack system.

this attack strategy.

3.2.1 Out-of-Network WiFi Traffic Capturing

We conducted initial tests with various well-known sniffing tools, namely Airodump-ng and Kismet, to capture raw 802.11 wireless frames. Both tools have the capability for either single-channel or multi-channel surveillance, employing frequency hopping to cycle through channels. Airodump-ng differentiates itself by saving captured data in a pcap file format, while Kismet records data within an SQLite3 database. The pcap files generated by Airodump-ng are readily compatible with network analyzers such as Wireshark for packet inspection. Our setup included the use of an Alfa AWUS036ACM external wireless adapter set to monitor mode, as depicted in Figure 3.3.a.

This was essential as built-in WiFi cards typically filter out packets not addressed to them. Upon reviewing the captured data, we noticed a prevalent number of small-sized packets, which is atypical given the commonly observed pattern of internet traffic known as the elephant-mouse phenomenon. In our case, larger packets, like those typically spanning

1,500 bytes including smart TV and camera video packets, were absent.

Our findings indicate that the capture range of both Airodump-ng and Kismet is limited to packets no larger than 472 bytes. Although this range is adequate for certain signal intelligence tasks—for instance, rogue access point (AP) detection as demonstrated by researchers who utilized Kismet to distinguish rogue APs by signal strength—there are constraints. Given the limitations encountered with Airodump-ng and Kismet, we proceeded to evaluate an alternative sniffing tool known as Airtool.

Airtool is a complimentary network traffic analysis tool designed for Mac computers, leveraging the Mac's native network interface card (NIC) for operations, as seen in Figure 3.3.b. This tool can passively monitor WiFi traffic and save the data in pcap format, which is compatible with Wireshark for in-depth traffic analysis. To assess the efficacy of Airtool in capturing comprehensive traffic data, we conducted a parallel traffic capture on a MacBook operating outside of the network and another laptop connected within the network, using Wireshark to record the laptop's bi-directional traffic with the AP.

Our comparative analysis of the traffic data, as summarized in Table 3.1, revealed that Airtool captures a broader range of frames, including control and management frames within the data-link layer, which Wireshark does not typically collect in its in-network traffic logs.

These additional frames, often ranging from 0 to 39 bytes, contribute to Airtool logging more entries. In contrast, Wireshark interprets WiFi data-link layer frames as Ethernet II frames, resulting in a smaller recorded size for the same WiFi packet. Consequently,

Table 3.1: Airtool testing: comparison of passive Airtool capture with Wireshark.

Packets/Frames Size Range	Wireshark	Airtool	
	#IP Packets	#Frames	#Data frames
0-19	0	2428	0
20-39	0	5593	199
40-79	2441	0	0
80-159	260	2890	2883
160-319	108	239	239
320-639	173	194	190
640-1279	241	255	255
1280-2559	13574	13846	13846
Total	16797	25445	17612

many packets that Wireshark categorized in the 40-79 byte size range were recorded in the 80-159 byte range by Airtool. Even after filtering out the control and management frames from Airtool's dataset, our comparison affirmed that Airtool did not significantly miss packets, reinforcing its reliability. Therefore, we have chosen Airtool as the sniffing tool for our experimental testbed.

3.3 Pre-processing of captured WiFi traffic

Following the capture of encrypted WiFi traffic by Airtool software, the collected pcap file format traces are subjected to analysis with Wireshark. The initial stage involves a detailed pre-processing of this raw trace data, which includes several critical steps:

- Extraction of Bidirectional Flows: The focus is on extracting the traffic flows related to the MAC address of the targeted WiFi router, essential due to Airtool's capabil-

(a) Sniffing using Alfa WiFi interface and either Airodump-ng or Kismet software.

(b) Sniffing using MacBook built-in WiFi interface and Airtool software.

Figure 3.3: Two ways of setting up out-of-network capturing.

ity to capture traffic from multiple access points (APs) simultaneously. We narrow down to only data frame types, discarding control and management MAC-layer frames which don't align with the desired data profiling pattern. This is accomplished using a specific Wireshark display filter:

```
<(wlan.sa == "Router's MAC" && wlan.da == "Router's MAC")
&& wlan.fc.type == 2>
```

- Conversion to CSV: The pcap files are converted into CSV format to facilitate further processing in the subsequent steps.

- Noise Frame Removal: Frames produced by some MAC addresses that are identified as noise are eliminated. Such frames usually appear singly and are filtered by retaining only those frames that show bidirectional communication traffic, thus showcasing both transmission and reception activities.

- MAC Address Replacement: In this phase, MAC addresses are replaced with corre-

sponding device names and their operation statuses for easier dataset labeling. This replacement aids in dataset organization but is only conducted during the offline training stage. The process is omitted during the online attack phase to ensure labels are applied appropriately for classification training and testing verification.

- Feature Extraction via Python Script: Utilizing a Python script, we extract and compute statistical features, as will be further detailed in Section 3.3.5. This culminates in the creation of a dataset ready for both offline training and subsequent performance evaluation testing.

These steps systematically refine the raw data, ensuring it's primed for in-depth analysis and aiding in the identification of devices based on their network traffic patterns.

3.3.1 Data Processing and Profiling Based on Machine Learning

In this part of our discussion, we initiate by exploring the data fields that are visible and can be harnessed during out-of-network monitoring activities. Following this, we propose and illustrate two distinct data processing methodologies designed to produce representative datasets that can be efficiently integrated into machine learning (ML) classification frameworks: Time-series data and Summary data.

3.3.2 Observable Data Fields in Out-of-Network Monitoring

In the context of monitoring activities carried out on a secure WiFi network from an external standpoint, the data available for observation is limited due to encryption protocols like WPA-PSK, which secure the content at and above the data-link layer. As a result, the observable components are confined to the MAC-layer frame header, along with the frame's observation timestamp and its signal strength. Within the MAC-layer frame header, valuable information can be extracted such as the source and destination MAC addresses, the frame type, and the frame's size.

Initially, it was hypothesized that signal strength could serve as a beneficial attribute for analysis, potentially indicative of the device's hardware characteristics or its proximity to the access point (AP), distinguishing between mobile and stationary devices. However, subsequent experimental analysis revealed that signal strength is influenced by a multitude of unpredictable elements, including interference from nearby WiFi networks and the impact of environmental factors like object reflections, absorption, and deflection within the vicinity. Due to these variables introducing significant inconsistencies, signal strength was ultimately excluded from our device profiling methodology.

3.3.3 Preliminary Data Analysis

To delve into the nuances of IoT device traffic, we embarked on an analysis of traffic data collected over a five-minute span from three distinct IoT devices selected for demonstration purposes: a smart light bulb, a smart plug, and a WiFi-enabled printer. Mid-

way through this observation period, we altered the operational states of these devices by switching off the light bulb and plug, and ceasing printing operations, rendering the printer inactive.

The investigation focused on two primary types of header-based characteristics: (a) flow-related features and (b) volume-related features, observable within a predetermined time-frame. Flow-related characteristics are concerned with the patterns of data transmission, including frequency and duration, while volume-related attributes measure the data traffic in terms of bytes. Despite the seemingly restricted scope of available data, it provides ample information to create distinctive profiles or signatures, highlighting the unique statistical variances across different device categories. For instance, the printer demonstrated a notable difference in the quantity of packets received compared to the light bulb and plug, as depicted in Figure 3.4. Conversely, the smart light bulb and plug showed particular packet size tendencies, predominantly sending packets of 82 bytes and 92 bytes respectively. This differentiation in packet sizes is further illustrated in Figure 3.5 through a word cloud, suggesting the potential for adversaries to develop precise device signatures based on these statistical indicators.

(a) Smart light bulb

(b) Smart plug

Figure 3.4: Word cloud of sent packet sizes from two devices.

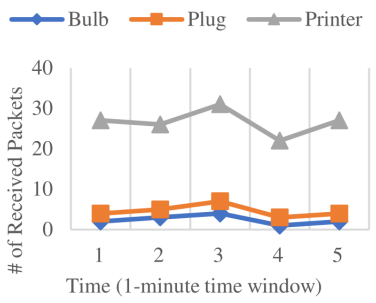

Figure 3.5: Traffic flow of three devices.

Moreover, the analysis allowed for the detection of changes in device operational statuses. A slight increase in data traffic, as shown in Figure 3.4, was observed across all devices, attributed to the AP transmitting commands to deactivate the bulb and plug, and halt the printer. This pattern represents a short-lived network fluctuation, perceivable only momentarily, which contrasts with the more prolonged statistical features derived from extended observation windows (exceeding 20 seconds). Consequently, inferring the operational states of devices with minimal traffic variation proves challenging. Conversely, devices with more advanced network functionalities and storage capacities, such as Alexa, smart TVs, and WiFi cameras, demonstrate significant traffic reductions when transitioning to an idle state. For example, an idle Alexa device receives far fewer packets since it's not engaged in processing voice commands or performing internet searches, such as checking weather forecasts or delivery statuses.

3.3.4 Machine Learning Algorithms

We selected a variety of machine learning (ML) models for our study, including Random Forest (RF), Support Vector Machine (SVM), and Naïve Bayes (NB), to conduct both learning and inference tasks. RF has been highlighted in the literature [63, 40] for its exceptional performance in classifying network traffic across a comparison of 11 ML models. We also opted for SVM and NB, considering the diversity in the sequence sizes within our time-series dataset. SVM is recognized for its efficacy in handling multidimensional datasets, which aligns well with IoT devices characterized by high traffic volume.

Conversely, NB is known to perform well with smaller datasets, making it an ideal choice for devices that generate less frequent and smaller volumes of network traffic [111]. In our approach to classifying IoT devices, we include an additional category named 'unknown'. This category encompasses all devices for which the classifier cannot assign a class with a certain level of predefined confidence.

3.3.5 Device Profiling based on Summary Data

In this methodology, we classify IoT devices based on traffic characteristics that are captured within a predefined time frame. As depicted in Figure 3.6, we segment the monitored traffic, which spans n seconds, into intervals or windows of W seconds each. This process begins with an initial time window $[t_{\text{start}} \dots t_{\text{end}}]$, and we systematically advance both the start and end of this window by W seconds, provided that $t_{\text{end}} + W \leq n$. Consequently, from n seconds of observed data, we can derive $\frac{n}{W}$ sample points.

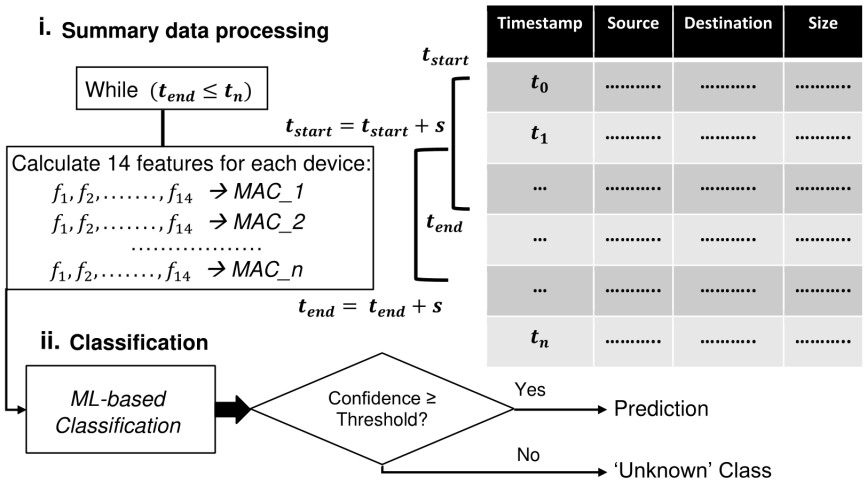

Figure 3.6: Summary data processing and classification.

To further augment the dataset with additional sample points, we employ a sliding window technique with a step size of $s = \frac{W}{2}$, meaning we shift the window $[t_{\text{start}} \ldots t_{\text{end}}]$ by s seconds instead of the full window size W. This adjustment effectively doubles the number of data points to $\left(\frac{n}{W} \times 2\right) - 1$. Identification of each device within these time windows is achieved through its MAC address, which facilitates the subsequent extraction and labeling of features. Upon compiling our dataset, we proceed to construct three classifiers utilizing the selected ML algorithms: SVM, NB, and RF.

We list all 14 extracted features from monitored packets within each time window:

- The number of packets sent from the device to AP.
- The number of packets received by the device from AP.

- The variance of inter-arrival time.
- The average number of consecutively sent packets before seeing a received packet.
- The average number of consecutively received packets before seeing a sent packet.
- Total number of bytes in sent packets.
- Total number of bytes in received packets.
- Number of different sizes in sent packets.
- Number of different sizes in received packets.
- Maximum packet size.
- Mode of sent packet lengths (i.e., the packet size that appeared most in the monitoring window).
- Mode of received packet lengths.
- The variance of sent packet size distribution.
- The variance of received packet size distribution.

3.4 Evaluation

In this section, we first discuss our testbed to collect the dataset and our evaluation metrics. Then, we present our evaluation results on the testbed trace.

3.4.1 Testbed Setup and Evaluation Metrics

We established a testbed consisting of 10 different IoT devices and 2 non-IoT devices (a smartphone and a laptop), all connected to the Internet via a WiFi router. To gather ground truth data, we recorded the network activity of all devices over a monitoring period of one hour. Table 3.2 presents the devices and their operational modes used in our dataset capture setup. While these scenarios do not cover the full range of device functionalities, they represent a variety of common usage patterns. From the captured raw packets, we created two datasets (in time-series and summary data formats as detailed in the previous section) and randomly divided each dataset into two groups, allocating approximately 75% of the instances for training and 25% for testing.

We evaluate our classification models using the following aspects: Accuracy, Precision, Recall and F1 Score. Let us denote true prediction as T, broken further into true positives TP and true negatives TN. Likewise, false prediction is denoted as F, broken into false positives FP and false negatives FN. Accuracy is measured as $\frac{T}{T+N}$, Precision is measured as $\frac{TP}{TP+FP}$, Recall is measured as $\frac{TP}{TP+FN}$, and F1 is measured as $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$.

3.4.2 Model Accuracy Results

Table 3.2 presents the comparative accuracies of different ML models using time-series versus summary data with a 30-second time window applied in the evaluation. The test incorporates non-IoT devices such as a laptop and an iPhone. Across all scenarios, the evidence indicates that the summary data consistently outperforms the time-series data

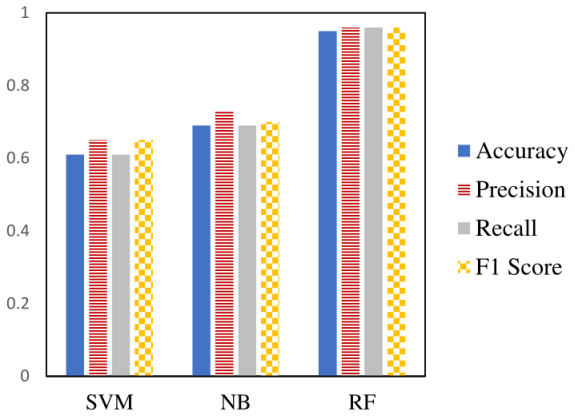

Figure 3.7: Overall performance of summary data-based profiling.

in terms of accuracy. Furthermore, the Random Forest (RF) model surpasses the other two ML models in predictive precision. It is postulated that time-series patterns require an extensive duration of trace observation to be effectively discerned, potentially over a 30-minute period, to execute an attack. Nonetheless, such prolonged surveillance is considered impractical for realistic attack situations. As illustrated in Table 3.3, the summary data profiling method provides more precise outcomes than the time-series approach for all models and devices. This greater accuracy can be attributed to the more discriminative feature set in the summary data, which effectively profiles a diverse array of devices.

For example, the consistent packet transmission of a TV is readily identifiable, unlike the variable Google Home traffic, which the time-series data may not capture, but which becomes evident when analyzed through packet size in the summary data.

Table 3.2: Accuracy of ML models.

Data Type	**SVM**	**NB**	**RF**
Time Series Non-IoT	0.34	0.25	0.41
Time Series IoT	0.57	0.74	0.68
Summary Data Non-IoT	0.51	0.41	0.94
Summary Data IoT	0.65	0.77	0.96

Additionally, the data reveals that the RF algorithm demonstrates superior performance compared to the SVM and NB algorithms across all tests using summary data. Figure 3.7 further delineates these evaluation metrics, reinforcing the dominance of RF: it excels beyond SVM and NB across all metrics, attaining a minimum of 95% in each.

3.4.3 Working Status Detection and Detection Speed

Our analysis extends to assess whether the RF model can discern the operational status of devices that have dual working modes—namely, busy versus idle. The findings on this aspect are conveyed in Figure 3.8, which depicts classification accuracies exceeding 90%, with certain exceptions such as the iPhone in idle mode and Amazon Echo in both active and idle modes.

Notably, the Amazon Echo experiences the lowest accuracy rate; 28.1% of the time, an Echo in busy mode is mistakenly classified as being in idle mode. This discrepancy primarily stems from the quick execution of certain commands (like illuminating a light), which generates minimal traffic. Consequently, the resulting traffic pattern bears resemblance to that of an idle state.

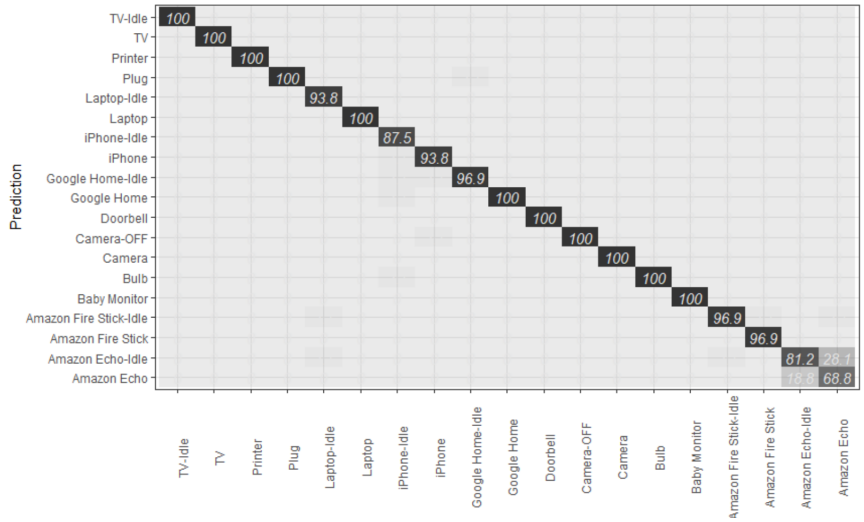

Figure 3.8: Confusion matrix of our IoT devices classification based on summary dataset using RF machine learning algorithm.

As elucidated in Section 3.3.2, certain devices, including printers, plugs, doorbells, smart bulbs, and baby monitors, do not exhibit significantly different traffic patterns when transitioning between working states. Therefore, in our experimental setup, each of these IoT devices is represented by a single model, without differentiation between operational states.

Additionally, we examined the influence of time window duration on accuracy by adjusting the window from 20 to 60 seconds, as illustrated in Figure 3.9. A notable observation was a 3% improvement in accuracy with the RF model when the window was extended to 30 seconds; however, the accuracy plateaued for windows ranging from 30 to 60 seconds. Hence, we designated the 30-second window as the standard for our evaluations.

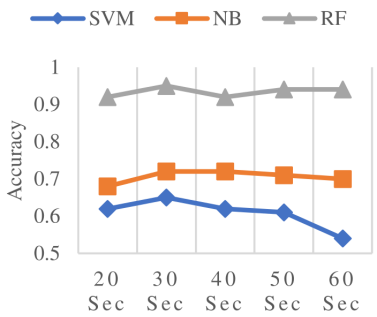

Figure 3.9: Impact of time window size on accuracy.

3.4.4 Discussion

Our study has confirmed that it is indeed feasible for an attacker to profile IoT devices from outside the network without needing to gain access. By examining elements like the quantity of data packets, the intervals between their arrivals, the dimensions of the packets, and the pattern of their size distribution, we can accurately identify the types of devices and often discern their operational states. This method of attack is straightforward (it requires neither network access nor specialized tools), swift (yielding over 95% accuracy within a mere 30 seconds), and stealthy (leaving no traces that can be tracked).

Implications: of such an attack are considerable, particularly in terms of privacy risks. For one, it paves the way for clandestine corporate espionage. A potential attacker could simply drive by a commercial vicinity to deduce information about the level of business activity, the economic status of customers, potential earnings, or even identify devices for subsequent cyber attacks. On a more advanced level, the technique can be used to monitor

and follow mobile devices like cars, drones, or phones. Deploying a fleet of drones across a broad area could enable the classification and tracking of various signal-emitting devices, providing insights into their interactions and movements.

The scalability of this out-of-network profiling attack remains an open question. While the approach is proven effective within our experimental setup, its applicability in a real-world, uncontrolled environment is yet to be explored and will be a subject of our future research.

As for defensive strategies, constructing a solid barrier against such device fingerprinting, particularly at the data-link layer, comes with significant implementation costs. For instance, encrypting the MAC header information could mask the identities of devices and decrease the likelihood of a successful attack, but it's not a complete solution and the associated overheads for key management and encryption could render it unfeasible [67]. Defenses designed for network or application layers do not translate effectively to the data-link layer. One alternative defense could be to employ obfuscation using virtual identifiers, such as virtual wireless clients (VWC)[1, 26], which generate multiple virtual network interfaces for each physical network card. By distributing a device's traffic across several virtual clients, either randomly or by a set of rules, we might introduce enough randomness to thwart an attacker's attempt to correlate different MAC addresses with a single WiFi device.

3.5 Summary and Recommendations

Summary: This chapter has detailed the methodology for detecting IoT devices through passive monitoring of encrypted wireless traffic. By capturing and preprocessing traffic data, extracting relevant features, and employing machine learning models by using data summary and random forest model, it is possible to profile and identify connected IoT devices without directly accessing the network. This approach highlights the potential privacy risks posed by passive surveillance techniques and underscores the need for enhanced security measures in IoT deployments.

Recommendations:

1. Developing robust defensive mechanisms to obfuscate IoT device traffic patterns and prevent device fingerprinting attacks. Potential strategies include encrypting MAC header information, employing virtual wireless clients (VWCs) to distribute traffic across multiple virtual interfaces, or introducing controlled randomness in traffic patterns.

2. Enhancing security measures and privacy-preserving mechanisms in IoT device communications, such as implementing end-to-end encryption and limiting the exposure of device metadata.

3. Raising awareness among IoT device manufacturers, network administrators, and end-users about the risks posed by passive monitoring attacks and the importance of implementing appropriate countermeasures.

4. Conducting further research to assess the scalability and effectiveness of this attack technique in real-world, uncontrolled environments, and exploring potential mitigation strategies.

5. Collaborating with regulatory bodies and industry stakeholders to establish guidelines and best practices for secure IoT device deployment and network traffic monitoring.

While the technique presented in this chapter highlights a significant vulnerability in IoT ecosystems, it also underscores the need for proactive measures to enhance the security and privacy of wireless networks and connected devices. By addressing these challenges through rigorous research, innovation, and collaboration, we can pave the way for a more secure and trustworthy IoT landscape.

CHAPTER 4: Time-Series Data: Concepts, Techniques, and Use Cases

This chapter investigates the feasibility of fingerprinting and identifying Internet of Things (IoT) devices by eavesdropping on their encrypted WiFi network traffic from outside the network. It presents a detailed analysis of capturing and processing time-series data from the wireless traffic of various IoT devices. By extracting statistical features and applying machine learning techniques like XGBoost on the time-series data, the proposed approach demonstrates the ability to accurately classify and profile different types of IoT devices based on their network traffic patterns, even when observing the traffic from outside the encrypted WiFi network. The results show promising accuracy of up to 94% in identifying and distinguishing between active and idle states of 10 different IoT devices, highlighting potential privacy concerns and the need for enhanced security measures in IoT device communications.

4.1 Problem Statement, Threat Model, and Assumptions

In this section, we first define the problem statement that serves as a strong motivation for the proposed research. Afterward, we present the threat model that needs to be covered by the proposed study. Finally, we discuss the considered assumptions while conducting the proposed research.

4.1.1 Problem Statement

The TCP/IP paradigm is used for communicating devices on networks. The IP address at the Network Layer and the MAC address at the Data Link Layer can be used to identify the devices on the network. Spoofing identities have been used to get around these identifying mechanisms and access restricted resources. Using WiFi traffic analysis, an attacker can "fingerprint" devices to determine private user behavior [58]. For instance, by continuously watching the camera's bitrate, the attacker could ascertain the movements of objects inside a building [111]. Moreover, the attacker can predict which vulnerabilities are available to exploit depending on the type of IoT devices in the network. However, extensive research has been conducted on fingerprinting the IoT devices using eavesdropping from the inside network [103]. This research performs a detailed investigation and proves that fingerprinting the IoT devices eavesdropping from outside the network is not only possible but also a straightforward process. The developers of the IoT devices must need to consider some extra security constraints to overcome those privacy threats.

4.1.2 Threat Model

In the proposed research, we consider that the attacker aims to target the information of IoT devices using a targeted WiFi network. Moreover, the attacker is also interested in the number and type of unique IoT devices, e.g., Laptop, Smart TV, Light Bulb, etc. Moreover, the attacker is also interested in getting the mode of those devices, such as idle or active. The attacker intends to gain maximum sensitive information by gathering

the devices' data. For example, the device type may reveal potential vulnerabilities to software/hardware status. The number of devices may reveal the customers in business, the number of employees, or the family size. The type and number both can reveal the status of socioeconomic.

Considering this as a potential threat model, we aim to perform a detailed investigation that fingerprinting the IoT devices eavesdropping from outside the network is a straightforward process. This threat should be considered in the first place.

4.1.3 Assumptions

In the proposed research investigation, we assume that the attacker continuously observes the network traffic outside the network using the targeted WiFi or access point. We also believe that the attacker is physically in the signal range of the access point, so he can perform eavesdropping using a sniffing tool and gather the nearby WiFi network traffic. We assume that the attacker can't join or break the network. To this end, in the proposed research investigation, we prove that fingerprinting the IoT devices from the outside of the network eavesdropping is possible. Moreover, the existing research focuses on the IoT devices operated at 2.4GHz; we consider the same. However, the proposed study can be applied to 5GHz as well.

4.2 Verification of Collective Movement

In this section, we first present the system architecture, and then we discuss how the attacker captures the network traffic from outside of the network. After that, we present the pre-processing of the captured data.

4.2.1 System Architecture

In the proposed investigation, we consider the system architecture illustrated in Figure 2 where the attacker uses to access the WiFi network. The system architecture consists of two stages, offline and online. The first stage (offline) is the attacker's profiling model training and building stage, where an attacker uses his computer and many IoT devices to conduct experiments in order to build the profiling model of each IoT device. On the other hand, the second stage (online) is the attacking stage, where the attacker monitors a WiFi network, trying to identify all IoT devices in the WiFi network based on monitored data and profiling models built in the offline stage. In the first stage (offline), the attacker configures maximum IoT devices which are connected to the nearby WiFi gateway. The attacker accesses the network traffic using a sniffing tool, where the traffic data would be labeled as the device name using the MAC address. The collected data is then pre-processed, where we removed the noise (e.g., network traffic gathered from nearby WiFi networks, data link layer broadcast frames, WiFi protocol beacon frames), and dumped the valuable features into a CSV file for applying the machine learning techniques. In particular, we apply several machine learning algorithms and achieve accuracy up to 95%

for device identification.

In the second stage (online), the attacker applies a sniffing tool and targets the victim's access point for a short period of, for example, 30 seconds, and stores the traces for pre-processing. To this point, we never require prior knowledge of the IoT devices for pre-processing, which we will explain in detail later in the following subsection. Precisely, we use standard and statistical filtering techniques to eliminate the noise from the data frames which do not represent the patterns of data. After that, we use Python scripting to extract the features from pre-processed data. Finally, we were able to predict the type of devices and their activity.

4.2.2 Traffic Capturing From Outside of the WiFi Network

In order to capture the data frames from outside of the WiFi network, we first test to use the two most popular sniffing tools, Kismet[1] and Airodump-ng[2]. Kismet stores the network traces as SQLite3 database, whereas Ariodump-ng dump the traces into a capture file format such as pcap[3].

The output of pcap is used to perform packet inspection as the output is in a compatible format, where the packet inspection can be done via a network analyzer such as Wireshark[4]. We use those sniffing tools because of their capability to sniff raw 802.11 frames. Besides, both of them are able to monitor single-channel and multi-channel using frequency hopping. For the hardware, we use an external wireless adapter (Alfa AWUS036ACM) as the built-in WiFi cards don't serve our purpose because they are programmed to accept the

data packets which are particularly addressed to the machine's interface card.

Once the captured traffic is analyzed, we observe that a significant proportion of captured packets contradicts the elephant-mouse internet traffic phenomenon [16]. The elephant flows of 1500 bytes were unable to see for all the available devices, including the video packets captured from a Camera or a smart TV. The proposed investigation shows that the aforementioned tools can only capture a limited range of packets in terms of their sizes.

For example, they are able to capture the packet up to the frame size of 472 bytes; however, this size is enough for particular applications such as signal intelligence. Due to such limitations of Kismet and Airodump-ng, we consider another sniffing tool called Airtool[5]. The Airtool sniffer is a MAC's built-in sniffing tool that can passively sniff WiFi traffic and store the traces in a pcap format which can be further analyzed using Wireshark.

We simultaneously run Wireshark on a different laptop connected to the network to record its own incoming/outgoing network traffic to the AP in order to confirm the accuracy of the traffic caught by Airtool which is running on an out-of-network MacBook. The traffic between the second laptop connected to the network and the AP was considered for comparison of those two traces.

4.2.3 Data Pre-Processing on Captured Data

Once the encrypted WiFi traffic data is captured using the Airtool software, the output in pcap format is analyzed using the Wireshark tool. In particular, the following steps are taken to analyze the captured data:

1. We start with the traffic broadcasting in both directions to the MAC address of the WiFi network under investigation. This is required since Airtool could potentially monitor WiFi traffic from many neighboring APs. Only data frame types are kept since all other control, and management MAC-layer frames do not adequately depict the profiling data pattern.

2. We export the pcap files into the csv files for the following steps number 3 and 4.

3. We eliminate noisy frames that some MACs produced. Since they often only appeared as a single frame, these noise frames are simple to filter out. By just keeping traffic frames with bi-directional communication traffic, they are filtered away.

4. To make dataset labeling easier, we swap out the MAC addresses for the relevant device names and their operational status. This step is included only in the offline training stage.

5. The dataset required for both offline training and performance testing is eventually obtained using a Python script that extracts and calculates statistical characteristics.

4.3 Data Processing and Analysis

In the section, we first discuss the observable Data Fields, and then we discuss data analysis. Finally, we discuss device profiling on time-series data.

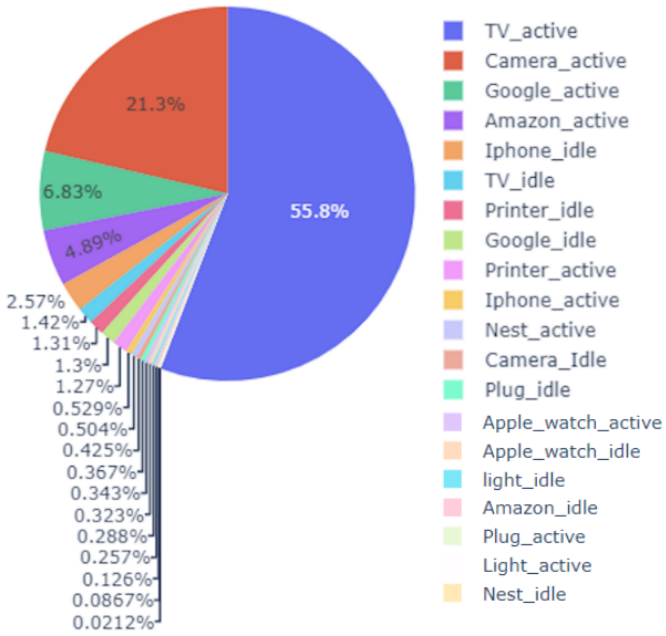

Figure 4.1: The total number of data packets captured with respect to the IoT device. Many devices' names show whether they are in active or in an idle state.

4.3.1 Useful Data from Network Monitoring

As we have discussed earlier, the process of out-of-network monitoring. Here, we present the observable data fields on a secured WiFi network where everything above the data-link layer is encrypted because of WiFi protocol *WPA-PSK*. Because of this encryption, the only observable data is the MAC-layer frame header, signal strength, and the observation timestamp. The frame header provides the source and destination's MAC addresses, frame size, and frame type. However, the signal strength cannot be used as it has been affected by several factors such as neighboring WiFi networks, deflection, absorption, and reflections of the surrounding objects. Therefore, in the proposed investigation, we neglect the signal

strength and only use the MAC layer frame header.

4.3.2 Data Analysis

As discussed earlier, we consider 10 different IoT devices and collected their data through out-of-network monitoring. we show the total percentage of captured data from each device. To provide insights into the monitored traffic, we measure the working and idle status of all the IoT devices. We can observe that the number of received packets from TV is more than the other IoT devices such as the camera and Google. All of the other devices' captured data are comparatively way less than those three devices. Consequently, the camera and google both display different behavior with regard to packet sizes. Specifically, the camera and google both appear to send the majority of their packets at a fixed size of 170 bytes and 140 bytes, respectively. We will discuss this part in the following section.

We believe that the attacker can easily build the signature. Moreover, the attacker can also change the status of those IoT devices. As we can see, there is a huge difference between the active and idle states of the devices which means the AP initiated communication to send an off signal to those devices. On the other hand, due to the notable decline in flow when switching to the idle state, the working condition of other devices with better network capabilities and memory storage, such as iPhones, printers, and Amazon, is impressively noticeable. For instance, Amazon will only get a small number of packets when it is idle because the user is not searching the Internet. Similarly, we show a detailed transmission of data packets from each IoT device. In particular, we show the total number of packets

sent from the access point to the IoT device, which is represented by **1**, and the total number of packets sent from IoT devices to the access point, which is represented by **0**. Moreover, the figure depicts both the total data captured in the active and idle states of all the devices.

4.3.3 Machine Learning Techniques

In order to execute our investigation, we choose several classification methods, and a popular machine learning model XGBoost [28]. We consider XGBoost because of its superior performance, especially for the problems of network classification among other popular machine learning models. Because our data is captured in two different sequence sizes, so we consider precision, recall f-1 score, and support vector machine (SVM) as performance metrics to tackle the time series data.

4.3.4 Profiling on IoT Devices using Time-series Data

Processing time-series data is simple. The captured traces from the Data-link layer are converted into a string of three-feature items. The monitored frame is then converted into three numeric values: the size of packet P, the direction of packet X, and the arrival time Y, where **0** represents the transmitted packets and **1** represents the received packets by an IoT device. Following those numeric values, we can obtain the series of data such as $\{P_0, X_0, Y_0\}, \{P_1, X_1, Y_1\}, ..., \{P_n, X_n, Y_n\}$.

This method's main disadvantage is that it needs a lot of packets to supply all the data

points needed for categorization or machine learning training. In our case, we are dealing with a heterogeneous system monitored inside a specific time window; certain devices (such as Smart TV) generate a large volume of data packets while others only produce very sparse packets (such as smart light bulb). For instance, if we compare how long it takes the light and TV to collect a 100-packet series, the TV just needs one second of visible data while the light needs approximately 30 minutes.

To overcome this challenge, we use a two-level categorization technique, starting with a traffic intensity threshold. Devices are divided into two groups in the first level according to whether there is a high or low volume of traffic. Then, in accordance with the volume of device traffic, we use an appropriate sequence size. Using ML algorithms, the second level determines the prediction probability. A prediction is made if the probability rises above a certain threshold; else, the data is labeled as an ”unknown” device. After you finish your AirTool data capturing, suppose you have N IoT devices. Split the packets into N files (or N tabs in Excel). Each file contains packets that all associate with one IoT device’s MAC (either as source MAC, or destination MAC). Just keep the data in the following format:

Timestamp	**From/To AP**	**Packet length**
t_1	0	n_1
t_2	1	n_2
t_3	1	n_3
t_4	0	n_4
t_5	0	n_5
t_6	1	n_6

Table 4.1: We use 1 to denote the packet is sent from AP to the IoT device, 0 to denote the packet is sent from the IoT device to the AP. Timestamp is according to time order.

Once the dataset is created, we extracted the following features from the monitored traffic in each time window:

1. Single feature data sequence: The simplest data for classification is a single sequence of packet length. There are three ways for such data sequence. We use the above example table to illustrate:

   a) From-AP only packet length data: (n_2, n_3, n_6, ...).

   b) To-AP only packet length data: (n_1, n_4, n_5, ...).

   c) Both-directional packet length data: (n_1, n_2, n_3, n_4, n_5, n_6, ...).

2. Two-feature data sequence:

   a) We use both the direction and the packet length as two-feature dataset:

$t_2 - t_1$	$t_3 - t_2$	$t_4 - t_3$	$t_5 - t_4$	$t_6 - t_5$	$t_7 - t_6$
n_1	n_2	n_3	n_4	n_5	n_6

   b) Another way is to ignore direction, but use packet interarrival time and packet length as two-feature dataset (remove the first packet since there is no interarrival time for it):

1	1	0	0	1
n_2	n_3	n_4	n_5	n_6

3. Three-feature data sequence: We use interarrival time, direction, and packet length for the three-feature dataset.

Justification for Feature Selection

Profiling IoT devices using time-series data involves extracting and utilizing features that best capture the distinctive patterns of device communication. The three primary features used in this analysis are packet length, direction, and interarrival time. Each of these features contributes uniquely to the classification and profiling of IoT devices, providing a comprehensive understanding of their traffic behavior.

Packet Length: Packet length is a critical feature because it reflects the amount of data being transmitted in each packet. Different IoT devices have varied functionalities and purposes, leading to distinctive packet sizes. For instance, a smart TV streaming video content will have consistently larger packet sizes compared to a smart light bulb, which transmits minimal data for simple commands. By analyzing packet lengths, we can identify patterns specific to each device type, making it a valuable feature for classification.

Direction (From/To AP): The direction of packets (whether they are sent to the Access Point (AP) or received from it) provides insights into the communication flow of the IoT device. Devices have different communication behaviors based on their operational states and roles. For example, a security camera might frequently send data to the AP (uploads) while occasionally receiving configuration commands (downloads). By distinguishing between packets sent to and from the AP, we capture this directional flow, which aids in differentiating devices based on their communication patterns.

Interarrival Time: Interarrival time, the time gap between consecutive packets, offers valuable information about the device's activity and usage patterns. Devices with periodic

updates, such as environmental sensors, have regular interarrival times, whereas devices with sporadic usage, like a smart thermostat, might have irregular patterns. Interarrival time helps in understanding the temporal dynamics of the traffic, providing an additional layer of differentiation among devices.

Combining Features for Enhanced Profiling:

- *Single-feature data sequence (Packet Length):* This simple approach focuses solely on packet length, categorized by direction, providing a baseline understanding of the traffic patterns.
- *Two-feature data sequence (Direction and Packet Length):* Combining direction with packet length enhances the classification by incorporating the flow of communication along with the data size, making it easier to distinguish devices with similar packet lengths but different communication patterns.
- *Two-feature data sequence (Interarrival Time and Packet Length):* Ignoring direction but using interarrival time and packet length captures the temporal and size-related aspects of the traffic, useful for identifying devices with distinct temporal patterns.
- *Three-feature data sequence (Interarrival Time, Direction, and Packet Length):* This comprehensive approach leverages all three features, providing a robust dataset that captures size, temporal dynamics, and directional flow. This rich feature set enhances the machine learning model's ability to accurately profile and classify IoT devices.

The heatmap of feature correlations further justifies the selection, showing significant relationships among these features, which collectively contribute to a more accurate and reliable classification model. The combination of these features ensures a thorough analysis, addressing the inherent heterogeneity and diverse communication patterns of various IoT devices.

4.3.5 Training and Testing

Let us denote the dataset sequence of one IoT device as: $(d_1, d_2, d_3, d_4, \ldots)$ For single-feature dataset, $d_i = n_i$. For two-feature dataset, such as (2.a) dataset, $d_3 = (t_4 - t_3, n_3)$, which is a two-entry vector. For three-feature dataset, $d_3 = (t_4 - t_3, 1, n_3)$, which is a three-entry vector. We utilize the initial portion of the dataset (either the first half based on the number of packets or the first half based on time, either is acceptable) as the training dataset to train a machine learning algorithm. During the training stage, we assume prior knowledge of the IoT device's identity. In the testing stage, the testing data (the latter half of the dataset) is denoted as $(d_m, d_{m+1}, d_{m+2}, d_{m+j}, \ldots)$, where j represents the size of the testing window. We aim to classify the IoT device using only these $j + 1$ data entries, allowing our system to profile and classify a monitored IoT device within that specified time window.

Initially, a large value can be chosen for j, as long as the testing data permits. If the classification is successful, the testing window can be reduced (i.e., the value of j is decreased). A smaller value of j indicates that our profiling process is faster.

	precision	recall	f1-score	support
amazon_active	0.99	0.97	0.98	16948
apple_watch_active	0.93	0.94	0.93	16589
Camera_active	0.94	0.94	0.94	16875
google_active	0.97	0.92	0.94	16783
Iphone_active	0.94	0.98	0.96	16864
Light_active	0.91	0.98	0.95	16941
Nest_active	0.96	0.98	0.97	16777
Plug_active	0.81	0.97	0.88	16917
TV_active	0.98	0.96	0.97	16737
Printer_active	1.00	1.00	1.00	16979
amazon_idle	0.97	0.95	0.96	17047
apple_watch_idle	0.96	0.96	0.96	16927
Camera_idle	0.86	0.88	0.87	16765
google_idle	0.89	0.84	0.86	17107
Iphone_idle	0.90	0.91	0.91	16792
Light_idle	0.89	0.89	0.89	17102
Nest_idle	0.98	0.99	0.99	16801
Plug_idle	0.97	0.76	0.85	16919
TV_idle	0.97	0.97	0.97	16887
Printer_idle	1.00	1.00	1.00	16835
accuracy			0.94	337592
macro avg	0.94	0.94	0.94	337592
weighted avg	0.94	0.94	0.94	337592

Figure 4.2: Machine learning model accuracy on Precision, Recall, and F1-Score.

4.4 Results and Discussion

In this section, we first present the testbed settings and the evaluation metrics. Afterward, we show the IoT devices' packets transmission and their reception in terms of packet length, and time. Finally, we show the evaluation results and prove that the outsider intruder can significantly harm the IoT devices without joining the WiFi network.

4.4.1 Testbed and Evaluation Metrics

With the help of a WiFi router, we built up a testbed with 10 distinct IoT devices. We use AirTool to capture the WiFi data frames be- tween all IoT devices and the WiFi router for an appropriate amount of time in order to collect enough data. Once the data is captured, we use a time-series format and randomly split the dataset into two groups, 20% for testing and 80% for training. The following metrics are used to assess our classification models: Precision, Recall, F1 Score, and Accuracy. Let's use the abbreviation T to stand for true prediction, further subdivided into true positives and true negatives. The letters F stand for false prediction, which is further divided into false positives and false negatives. The following equations are used to measure the Precision, Recall, F1 Score, and Accuracy, respectively. Accuracy, Precision, Recall and F1 Score. Let us denote true prediction as T, broken further into true positives TP and true negatives TN. Likewise, false prediction is denoted as F, broken into false positives FP and false negatives FN. Accuracy is measured as $\frac{T}{T+N}$, Precision is measured as $\frac{TP}{TP+FP}$, Recall is measured as $\frac{TP}{TP+FN}$, and F1 is measured as $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$.

4.4.2 Results

We chose a 30-minute time window size for evaluation. We believe that long-term observed traces can teach us more about time-series patterns than short-term ones, which call for much longer time observations to carry out the attack. As we were facing a challenge in working with imbalanced data, for example, data captured from some devices

are extremely high such as TV or Camera, whereas other devices are barely showing any record for example Nest, Light. To balance such data, we use SMOTE analysis to prove the significance of the results. Figure 4.3 shows the confusion matrix of prediction accuracy using a SMOTE analysis on the XGBoost model. The Figure shows that the IoT device ”Printer active” captures maximum true labels, whereas the IoT device ”google idle” captures minimum true labels.

Finally, we show the accuracy of all 10 IoT devices in each active and idle state with respect to Precision, Recall, and F1-Score. In this chapter, we investigated the feasibility of fingerprinting IoT devices by eavesdropping on encrypted WiFi traffic from outside the network. We defined the problem statement, presented the threat model, and discussed our assumptions. We then described our system architecture, how we captured traffic from outside the network, and pre-processed the data. Our analysis of the captured data revealed insights into the behavior of different IoT devices in terms of packet transmission and reception. We used machine learning techniques, including XGBoost and time-series classification, to classify the devices based on their traffic patterns. The results showed promising accuracy, with the XGBoost model. In Figure 4.3, we show that the model achieves up to 94% accuracy, demonstrating the potential for identifying and profiling IoT devices based on their WiFi traffic characteristics.

In this chapter, we explored the feasibility of fingerprinting IoT devices by monitoring their encrypted WiFi traffic from an external attacker's perspective. The training and testing methodology involved dividing the dataset into sequences based on time or the number of packets, using initial portions for training and the latter portions for testing. This setup

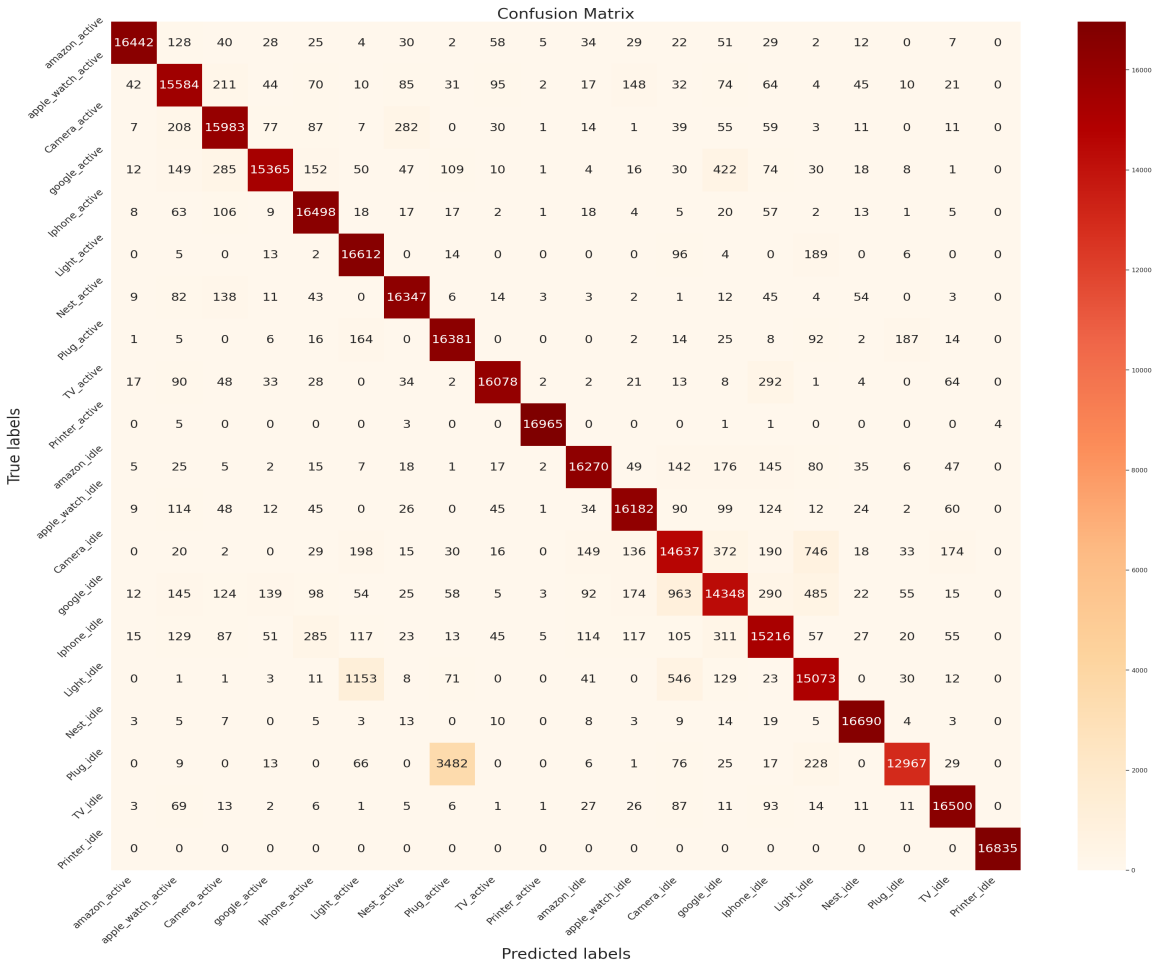

Figure 4.3: Predicted labels of each IoT device.

allowed for evaluating the performance of machine learning algorithms, particularly focusing on how accurately and quickly an IoT device could be profiled within a specified time window. The training process assumed prior knowledge of the device's identity, while the testing process aimed to classify the device using a minimal number of data entries to expedite profiling.

The results demonstrated that the machine learning models, especially XGBoost, could achieve high accuracy in classifying IoT devices based on their encrypted traffic patterns. The evaluation metrics used, including Precision, Recall, F1 Score, and Accuracy, provided a comprehensive view of the model's performance. The analysis of the confusion matrix revealed the variability in prediction accuracy among different IoT devices, highlighting that some devices were easier to classify accurately than others. This variability underscored the importance of addressing imbalanced data, which was mitigated using the SMOTE technique to ensure a more balanced representation of device activity in the training dataset.

Overall, the chapter's findings underscore significant privacy risks associated with the ability of an external attacker to profile IoT devices by passively eavesdropping on their WiFi traffic. This research highlights the necessity for device manufacturers to implement measures that obfuscate traffic patterns and for users to adopt best practices to mitigate these risks. Additionally, the study suggests that traditional encryption protocols like WPA2 are insufficient to prevent such attacks, as they do not hide metadata visible at the link layer. These insights pave the way for further research and development of more robust countermeasures to enhance the security and privacy of IoT ecosystems.

4.5 Summary and Recommendations

Summary: In this chapter, we investigated the feasibility of fingerprinting and profiling IoT devices by externally eavesdropping on their encrypted WiFi traffic. Through

capturing time-series data, extracting statistical features, and applying machine learning techniques like XGBoost, we demonstrated the ability to accurately classify different IoT devices and their active/idle states based solely on their network traffic patterns observed from outside the network.

Recommendations: The key findings and recommendations from this work are as follows:

1. The results highlight significant privacy risks, as an external attacker can potentially identify the types and number of IoT devices in a target network, as well as detect their operational modes, simply by sniffing the wireless traffic.

2. Device manufacturers should implement additional measures to obfuscate and randomize the network traffic patterns of their IoT devices, making it more difficult for external observers to fingerprint them based on time-series traffic analysis.

3. The use of encrypted WiFi protocols like WPA2 is not sufficient to prevent such fingerprinting attacks, as they only encrypt the payload data and not the metadata visible at the link layer.

4. Network-level anomaly detection systems should be developed to identify potential eavesdropping and fingerprinting attempts based on suspicious sniffing activities around the network perimeter.

5. IoT device users should be made aware of these privacy risks and adopt best practices, such as enabling MAC address randomization, using wired connections where possible, and minimizing the exposure of IoT devices to external networks.

6. Further research is needed to develop more robust countermeasures against such fingerprinting attacks, as well as to investigate the feasibility of extending this approach to other network types and protocols used in IoT ecosystems.

By highlighting these risks and providing recommendations, we aim to raise awareness and drive the development of more secure and privacy-preserving IoT systems, which are becoming increasingly prevalent in our daily lives.

CHAPTER 5: Conclusion

This dissertation delves into the privacy vulnerabilities posed by out-of-network eavesdropping on encrypted WiFi traffic. Through a comprehensive investigation involving 22 IoT devices, including 3 non-IoT devices, we demonstrate that eavesdropping on IoT devices is not only feasible but also a relatively straightforward process.

Leveraging WiFi frame timing and header information, we employ machine learning techniques to infer and fingerprint the presence of IoT devices in a network, as well as their operational status. Our models exhibit exceptional accuracy, with data summary achieving up to 95% accuracy and time series up to 94% accuracy in identifying devices and their working status. These results highlight the significant threat posed by outside intruders, who can exploit IoT devices without joining a WiFi network and launch attacks swiftly and stealthily.

The survey presented in this dissertation explores various classification approaches for encrypted network traffic from IoT devices, websites, and mobile applications. The classification methods for IoT devices primarily focus on device fingerprinting and OS identification, which are instrumental in device management and security. Similarly, website classification methods aim to detect privacy leaks and malicious activities, such as financial fraud, through techniques like web fingerprinting and user action identification. Mobile application classification methods center around network traffic analysis, QoE metric measurement, and anomaly detection, aiding in service optimization and security enhancement.

While these classification methods offer valuable insights, they also raise concerns regarding network security and user privacy. Countermeasures to mitigate these risks were discussed, emphasizing the importance of balancing security and privacy considerations. In conclusion, the classification of IoT devices, websites, and mobile applications is crucial for ensuring network security and providing personalized services. However, it necessitates careful consideration of its implications for security and privacy. Future research should focus on developing innovative approaches to address these challenges and ensure the safe and efficient operation of IoT devices, websites, and mobile applications.

In summary, this dissertation sheds light on the vulnerabilities of IoT devices to out-of-network attacks and provides valuable insights into classification approaches for encrypted network traffic. By addressing these challenges and implementing appropriate security measures, we can enhance the security and privacy of IoT devices and networks, as well as websites and mobile applications, ensuring a safer and more secure digital ecosystem.

5.1 Discussion and Future Work

Our findings demonstrate the feasibility of an outside-of-network attacker successfully identifying IoT devices without connecting to a WiFi network. This highlights significant vulnerabilities in current network security practices. The ability to determine device types and operational modes using characteristics such as packet count, inter-arrival time, and packet size distribution underscores the sophistication of potential attacks. Moreover, the ease with which this attack can be executed without leaving any discernible traces raises

serious privacy concerns.

The implications of such a profiling attack are profound. For instance, an attacker could drive near a business to gather information about its economic activity, clientele, and revenue trends. This information could be used for malicious purposes, such as targeting businesses with weak security measures or tailoring attacks to exploit specific vulnerabilities. Furthermore, this attack could be extended to track mobile devices in complex scenarios, such as monitoring the movement of vehicles or drones. The ability to map out device communication and movement patterns poses significant threats to privacy and security.

Given the importance of our experimental findings in real-world scenarios, it is crucial to further develop and implement security measures to mitigate the risks posed by out-of-network profiling attacks. One approach is to enhance encryption and security protocols to protect against such attacks. This could involve developing new encryption methods or enhancing existing ones to make them more resistant to profiling attacks. Additionally, implementing stricter access control measures and authentication protocols can help prevent unauthorized access to network traffic.

Privacy-preserving techniques are another area of focus for future work. These techniques aim to obfuscate or anonymize IoT device traffic to prevent identification and profiling. For example, using techniques like data masking or encryption can help protect sensitive information from being intercepted and analyzed by attackers. Furthermore, developing new methods for detecting and mitigating profiling attacks can help organizations better protect their networks and devices.

Regulatory considerations are also crucial in addressing the risks associated with profiling attacks. Advocating for regulatory measures to enforce stronger security practices for IoT devices and networks can help mitigate the risks posed by such attacks. This could involve implementing standards and guidelines for IoT device manufacturers to adhere to, as well as providing incentives for organizations to adopt more secure practices.

Continued research is essential to further explore the full extent of privacy risks posed by out-of-network profiling attacks. This includes conducting additional experiments and simulations to better understand the capabilities and limitations of such attacks. Additionally, developing more effective countermeasures and security solutions will be crucial in mitigating the risks associated with these attacks.

In conclusion, our research highlights the serious privacy and security risks posed by out-of-network profiling attacks on IoT devices. By expanding on our findings and exploring new avenues for research and development, we can better protect against these threats and ensure the security and privacy of IoT devices and networks.

List of References

[1] Ahmed Abusnaina, Rhongho Jang, Aminollah Khormali, DaeHun Nyang, and David Mohaisen. "DFD: Adversarial Learning-Based Approach to Defend Against Website Fingerprinting". In: *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 2459–2468.

[2] Ahmed Abusnaina, Aminollah Khormali, Hisham Alasmary, Jeman Park, Afsah Anwar, and Aziz Mohaisen. "Adversarial Learning Attacks on Graph-based IoT Malware Detection Systems". In: *ICDCS*. 2019, pp. 1296–1305.

[3] 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*. 2020, pp. 207–218.

[4] K.D. Adejuwon. "Internet of things and smart city development: is Nigeria leveraging on emerging technologies to improve efficiency in public service delivery?" In: *J Public Admin Finance Law* (2018), p. 13.

[5] B. Ahlgren, M. Hidell, and E. Ngai. "Internet of things for smart cities: interoperability and open data". In: *IEEE Internet Comput* 20.6 (2016), pp. 52–56. URL: https://doi.org/10.1109/MIC.2016.124.

[6] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: KDD '19. Anchorage, AK, USA: Association for Computing Machinery, 2019, 2623–2631. ISBN: 9781450362016. DOI: 10.1145/3292500.3330701. URL: https://doi.org/10.1145/3292500.3330701.

[7] Katie Boeckl et al. *Considerations for managing Internet of Things (IoT) cybersecurity and privacy risks*. Tech. rep. US Department of Commerce, National Institute of Standards and Technology, 2019.

[8] Muhammad Waqas Nadeem et al. "Internet of Things for Green Building Management: A Survey". In: *Role of IoT in Green Energy Systems*. IGI Global, 2021, pp. 156–170.

[9] I.A. Alharbi, A.J. Almalki, M. Alyami, C. Zou, and Y. Solihin. "Profiling Attack on WiFi-based IoT Devices using an Eavesdropping of an Encrypted Data Frames". In: *Adv. Sci. Technol. Eng. Syst. J.* 7 (2022), pp. 49–57.

[10] B. Ali. "Leach robust routing approach applying machine learning". In: *IJCSNS* (2019), p. 19.

[11] Jumabek Alikhanov, Rhongho Jang, Mohammed Abuhamad, David Mohaisen, Daehun Nyang, and Youngtae Noh. "Investigating the Effect of Traffic Sampling on Machine Learning-Based Network Intrusion Detection Approaches". In: *IEEE Access* 10 (2022), pp. 5801–5823.

[12] Ali Almalki and Pawel Wocjan. "Accuracy analysis of Educational Data Mining using Feature Selection Algorithm". In: (2021).

[13] Ibrahim A. Alwhbi, Cliff C. Zou, and Reem N. Alharbi. "Encrypted Network Traffic Analysis and Classification Utilizing Machine Learning". In: *Sensors* 24.11 (2024). ISSN: 1424-8220. DOI: 10.3390/s24113509. URL: https://www.mdpi.com/1424-8220/24/11/3509.

[14] Mnassar Alyami, Ibrahim Alharbi, Cliff Zou, Yan Solihin, and Karl Ackerman. "WiFi-based IoT Devices Profiling Attack based on Eavesdropping of Encrypted WiFi Traffic". In: *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*. 2022, pp. 385–392.

[15] R.C. Amorim. "Constrained clustering with minkowski weighted k-means". In: *2012 IEEE 13th International Symposium on Computational Intelligence and Informatics (CINTI)*. 2012, pp. 13–17.

[16] Sandhya Aneja, Nagender Aneja, and Md Shohidul Islam. "IoT device fingerprint using deep learning". In: *2018 IEEE international conference on internet of things and intelligence system (IOTAIS)*. 2018, pp. 174–179.

[17] Noah Apthorpe, Dillon Reisman, Srikanth Sundaresan, Arvind Narayanan, and Nick Feamster. "Spying on the smart home: Privacy attacks and defenses on encrypted IoT traffic". In: *arXiv preprint arXiv:1708.05044* (2017).

[18] M.R. Arbabshirani. "Advanced machine learning in action: identification of intracranial hemorrhage on computed tomography scans of the head with clinical workflow integration". In: *NPJ Digit Med* (2018), p. 1.

[19] S.B. Atitallah. “Leveraging deep learning and IoT big data analytics to support the smart cities development: review and future directions”. In: *Comput Sci Rev* (2020), p. 38.

[20] J. S. Atkinson. “Your wifi is leaking: inferring private user information despite encryption”. PhD thesis. UCL (University College London), 2015.

[21] A.T. Azar. “Drone deep reinforcement learning: a review”. In: *Electronics* 10 (2021).

[22] Ryan S Baker. “Educational data mining: An advance for intelligent systems in education”. In: *IEEE Intelligent systems* 29.3 (2014), pp. 78–82.

[23] F. Balducci, D. Impedovo, and G. Pirlo. “Machine learning applications on agricultural datasets for smart farm enhancement”. In: *Machines* 6 (2018).

[24] A. L. Buczak and E. Guven. “A survey of data mining and machine learning methods for cyber security intrusion detection”. In: *IEEE Commun. Surveys Tuts.* 18.2 (2016), pp. 1153–1176.

[25] A. Bundy. *Preparing for the future of artificial intelligence*. Berlin: Springer, 2017.

[26] Wladimir De la Cadena, Asya Mitseva, Jens Hiller, Jan Pennekamp, Sebastian Reuter, Julian Filter, Thomas Engel, Klaus Wehrle, and Andriy Panchenko. “TrafficSliver: Fighting Website Fingerprinting Attacks with Traffic Splitting”. In: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 2020, pp. 1971–1985.

[27] X. Cai, R. Nithyanand, and R. Johnson. “CS-BuFLO: A congestion sensitive website fingerprinting defense”. In: *Proceedings of the 13th Workshop on Privacy in the Electronic Society*. 2014, pp. 121–130.

[28] Tianqi Chen and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 785–794. DOI: 10.1145/2939672.2939785. URL: https://doi.org/10.1145/2939672.2939785.

[29] H. Cheng and R. Avnur. “Traffic Analysis of SSL-Encrypted Web Browsing”. In: (). URL: http://www.cs.berkeley.edu/~daw/teaching/cs261-f98/projects/final-reports/ronathan-heyning.ps.

[30] S. Chilamkurthy. “Deep learning algorithms for detection of critical findings in head CT scans: a retrospective study”. In: *Lancet* (2018), p. 392.

[31] Soohyeon Choi, Manar Mohaisen, Daehun Nyang, and David Mohaisen. “Revisiting the Deep Learning-Based Eavesdropping Attacks via Facial Dynamics from VR Motion Sensors”. In: *ICICS*. 2023, pp. 399–417.

[32] J.S. Chou and N.T. Ngo. “Time series analytics using sliding window metaheuristic optimization-based machine learning system for identifying building energy consumption patterns”. In: *Appl Energy* (2016), p. 177.

[33] M. Conti, Q. Q. Li, A. Maragno, and R. Spolaor. “The dark side (-channel) of mobile devices: A survey on network traffic analysis”. In: *IEEE Commun. Surveys Tuts.* 20.4 (2018), pp. 2658–2713.

[34] A. Das, W.-K. Ng, and Y.-K. Woon. “Rapid association rule mining”. In: *Proceedings of the tenth international conference on Information and knowledge management*. 2001, pp. 474–481.

[35] C.T.U.-University Dataset. *The Stratosphere I.P.S.Project*. 2016. URL: https://stratosphereips.org/category/dataset.html.

[36] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton. “Peek-a-boo, I still see you: Why efficient traffic analysis countermeasures fail”. In: *2012 IEEE Symposium on Security and Privacy*. 2012, pp. 332–346.

[37] A. Essien, I. Petrounias, P. Sampaio, and S. Sampaio. “A deep-learning model for urban traffic flow prediction with traffic events mined from twitter”. In: *World Wide Web* (2020), pp. 1–24.

[38] A. Essien, I. Petrounias, P. Sampaio, and S. Sampaio. “Improving urban traffic speed prediction using data source fusion and deep learning”. In: *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*. 2019, pp. 1–8.

[39] S. Feghhi and D. J. Leith. “An efficient web traffic defence against timing-analysis attacks”. In: *IEEE Transactions on Information Forensics and Security* 14.2 (2018), pp. 525–540.

[40] G. Fernandes, J. J. P. C. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proença. “A comprehensive survey on network anomaly detection”. In: *Telecommun. Syst.* 70.3 (2019), pp. 447–489.

[41] Y. Freund and R.E. Schapire. “Experiments with a new boosting algorithm”. In: *Icml*. Vol. 96. 1996, pp. 148–156.

[42] H. Fujiyoshi, T. Hirakawa, and T. Yamashita. “Deep learning-based image recognition for autonomous driving”. In: *IATSS Res* 43 (2019).

[43] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. “Generative adversarial nets”. In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[44] J. Guerrero-Ibáñez, S. Zeadally, and J. Contreras-Castillo. “Sensor technologies for intelligent transportation systems”. In: *Sensors* 18 (2018).

[45] S. Hakak, M. Alazab, S. Khan, T.R. Gadekallu, P.K.R. Maddikunta, and W.Z. Khan. “An ensemble machine learning approach through effective feature extraction to classify fake news”. In: *Future Generation Computer Systems* 117 (2021), pp. 47–58.

[46] J. Han, J. Pei, and Y. Yin. “Mining frequent patterns without candidate generation”. In: *ACM Sigmod Record* 29 (2000), pp. 1–12.

[47] S.A. Harmon. “Artificial intelligence for the detection of covid-19 pneumonia on chest ct using multinational datasets”. In: *Nat Commun* 11 (2020).

[48] J. He. “The practical implementation of artificial intelligence technologies in medicine”. In: *Nat Med* (2019), p. 25.

[49] K. He, X. Zhang, S. Ren, and J. Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[50] Guido R Hiertz, Dee Denteneer, Lothar Stibor, Yunpeng Zang, Xavier Pérez Costa, and Bernhard Walke. "The IEEE 802.11 universe". In: *IEEE Communications Magazine* 48.1 (2010), pp. 62–70.

[51] G.E. Hinton. "A practical guide to training restricted boltzmann machines". In: *Neural networks: Tricks of the trade*. Springer, 2012, pp. 599–619.

[52] A. Hintz. "Fingerprinting Websites Using Traffic Analysis". In: *Privacy Enhancing Technologies*. 2003, pp. 171–178.

[53] M. Houtsma and A. Swami. "Set-oriented mining for association rules in relational databases". In: *Proceedings of the Eleventh International Conference on Data Engineering*. 1995, pp. 25–33.

[54] J. Huang, Y.-F. Li, and M. Xie. "An empirical analysis of data preprocessing for machine learning-based software cost estimation". In: *Information and Software Technology* 67 (2015), pp. 108–127.

[55] G.F. Huseien and K.W. Shah. "A review on 5G technology for smart energy management and smart buildings in Singapore". In: *Energy AI* (2022), p. 7.

[56] Rhongho Jang, Jeonil Kang, Aziz Mohaisen, and DaeHun Nyang. "Catch Me If You Can: Rogue Access Point Detection Using Intentional Channel Interference". In: *IEEE Trans. Mob. Comput.* 19.5 (2020), pp. 1056–1071.

[57] RhongHo Jang, Jeonil Kang, Aziz Mohaisen, and DaeHun Nyang. "Rogue Access Point Detector Using Characteristics of Channel Overlapping in 802.11n". In: *ICDCS*. 2017, pp. 2515–2520.

[58] Zhiping Jiang, Kun Zhao, Rui Li, Jizhong Zhao, and Junzhao Du. "PHYAlert: identity spoofing attack detection and prevention for a wireless edge network". In: *Journal of Cloud Computing* 9.1 (2020), pp. 1–13.

[59] X. Jing, Z. Yan, and W. Pedrycz. "Security data collection and data analytics in the Internet: A survey". In: *IEEE Commun. Surveys Tuts.* 21.1 (2018), pp. 586–618.

[60] G.H. John and P. Langley. "Estimating continuous distributions in bayesian classifiers". In: *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1995, pp. 338–345.

[61] Changhun Jung, Sian Kim, Rhongho Jang, David Mohaisen, and DaeHun Nyang. "A Scalable and Dynamic ACL System for In-Network Defense". In: *CCS*. 2022, pp. 1679–1693.

[62] S.S. Kamble, A. Gunasekaran, and S.A. Gawankar. "Sustainable industry 4.0 framework: A systematic literature review identifying the current trends and future perspectives". In: *Process Saf Environ Protect* 117 (2018), pp. 408–425.

[63] Sina Fathi Kazerooni, Yagiz Kaymak, and Roberto Rojas-Cessa. "Identification of User Application by an External Eavesdropper using Machine Learning Analysis on Network Traffic". In: *17th IEEE International Conference on Communications Workshops, ICC Workshops 2019, Shanghai, China, May 20-24, 2019*. IEEE, 2019, pp. 1–6. ISBN: 978-1-7281-2373-8. DOI: 10.1109/ICCW.2019.8756709. URL: https://doi.org/10.1109/ICCW.2019.8756709.

[64] V. Khadse, P.N. Mahalle, and S.V. Biraris. "An empirical comparison of supervised machine learning algorithms for internet of things data". In: *2018 Fourth*

International Conference on Computing Communication Control and Automation (ICCUBEA). 2018, pp. 1–6.

[65] S. Khalid, T. Khalil, and S. Nasreen. “A survey of feature selection and feature extraction techniques in machine learning”. In: *Science and Information Conference*. 2014, pp. 372–378.

[66] S. Khan. “Towards interoperable blockchains: a survey on the role of smart contracts in blockchain interoperability”. In: *IEEE Access* (2021), p. 9.

[67] Lior Khermosh, Zachy Haramaty, and Jeff Mandin. “Implementing IEEE 802.1 AE and 802.1 AF Security in EPON (1GEPON and 10GEPON) Networks”. US Patent 8,397,064. 2013.

[68] Byung-Hak Kim, Ethan Vizitei, and Varun Ganapathi. “GritNet: Student performance prediction with deep learning”. In: (2018).

[69] A. Krizhevsky, I. Sutskever, and G.E. Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[70] S. Kumar, P. Tiwari, and M. Zymbler. “Internet of things is a revolutionary approach for future technology enhancement: a review”. In: *J Big Data* (2019), p. 6.

[71] S. Kushwaha, S. Bahl, A.K. Bagha, K.S. Parmar, M. Javaid, A. Haleem, and R.P. Singh. “Significant applications of machine learning for covid-19 pandemic”. In: *J Ind Integr Manag* 5.4 (2020).

[72] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim. “A survey of deep learning-based network anomaly detection”. In: *Cluster Comput.* 22.1 (2019), pp. 949–961.

[73] S. Lalmuanawma, J. Hussain, and L. Chhakchhuak. “Applications of machine learning and artificial intelligence for covid-19 (sars-cov-2) pandemic: A review”. In: *Chaos Sol Fract* (2020), p. 110059.

[74] C.D. Lehman. “Mammographic breast density assessment using deep learning: clinical implementation”. In: *Radiology* (2019), p. 290.

[75] J. Li, Z. Li, G. Tyson, and G. Xie. “Your privilege gives your privacy away: An analysis of a home security camera service”. In: *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE. 2020, pp. 387–396.

[76] S.H. Ling, P.P. San, and H.T. Nguyen. “Non-invasive hypoglycemia monitoring system using extreme learning machine for type 1 diabetes”. In: *ISA Trans* (2016), p. 64.

[77] B. Liu, W. Hsu, and Y. Ma. “Integrating classification and association rule mining”. In: *Proceedings of the fourth international conference on knowledge discovery and data mining*. 1998.

[78] U.K. Lopes and J.F. Valiati. “Pre-trained convolutional neural networks as feature extractors for tuberculosis detection”. In: *Comput Biol Med* (2017), p. 89.

[79] V. López, A. Fernandez, S. Garcia, V. Palade, and F. Herrera. “An insight into classification with imbalanced data: empirical results and current trends on using

data intrinsic characteristics". In: *Inf. Sci* 250 (2013), pp. 113–141. URL: https://doi.org/10.1016/j.ins.2013.07.007.

[80] C. Ma. "Smart city and cyber-security; technologies used, leading challenges and future recommendations". In: *Energy Rep* (2021), p. 7.

[81] Yu Mi, David Mohaisen, and An Wang. "AutoDefense: Reinforcement Learning Based Autoreactive Defense Against Network Attacks". In: *CNS*. 2022, pp. 163–171.

[82] L. Molina, A. Blanc, N. Montavont, and L. Simić. "Identifying channel saturation in Wi-Fi networks via passive monitoring of IEEE 802.11 beacon jitter". In: *Proceedings of the 15th ACM International Symposium on Mobility Management and Wireless Access*. 2017, pp. 63–70.

[83] A. Moore, D. Zuev, and M. Crogan. *Discriminators for Use in Flow-Based Classification*. Tech. rep. Queen Mary, University of London, 2005.

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

[85] D.W. Otter, J.R. Medina, and J.K. Kalita. "A survey of the usages of deep learning for natural language processing". In: *IEEE Trans Neural Netw Learn Syst* (2020).

[86] Fannia Pacheco, Ernesto Exposito, Mathieu Gineste, Cedric Baudoin, and Jose Aguilar. "Towards the Deployment of Machine Learning Solutions in Network Traffic Classification: A Systematic Survey". In: *IEEE Communications Surveys Tutorials* 21.2 (2019), pp. 1988–2014. DOI: 10.1109/COMST.2018.2883147.

[87] A. Paleyes, R.G. Urma, and N.D. Lawrence. “Challenges in Deploying Machine Learning: A Survey of Case Studies”. In: *ACM Computing Surveys* 55.6 (2022), pp. 1–29.

[88] A. J. Pinheiro, J. d. M. Bezerra, C. A. Burgardt, and D. R. Campelo. “Identifying IoT devices and events based on packet length from encrypted traffic”. In: *Computer Communications* 144 (2019), pp. 8–17.

[89] N. Prates, A. Vergütz, R. T. Macedo, A. Santos, and M. Nogueira. “A defense mechanism for timing-based side-channel attacks on IoT traffic”. In: *2017 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2017, pp. 1–6.

[90] J.R. Quinlan. “C4.5: programs for machine learning”. In: *Mach Learn* (1993).

[91] S. Raschka. “Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning”. In: *arXiv preprint arXiv:1811.12808* (2018).

[92] S. Rezaei and X. Liu. “Deep learning for encrypted traffic classification: An overview”. In: *IEEE Commun. Mag.* 57.5 (2019), pp. 76–81.

[93] L. Rokach. “A survey of clustering algorithms”. In: *Data mining and knowledge discovery handbook*. 2005, pp. 269–298.

[94] Eyal Ronen and Adi Shamir. “Extended functionality attacks on IoT devices: The case of smart lights”. In: *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. 2016, pp. 3–12.

[95] H. Salehi and R. Burgueño. “Emerging artificial intelligence methods in structural engineering”. In: *Eng. Struct* 171 (2018), pp. 170–189. URL: https://doi.org/10.1016/j.engstruct.2018.05.084.

[96] P. Santi, D. Ram, C. Rob, and E. Nathan. “Behavior-based adaptive call predictor”. In: *ACM Trans Auton Adapt Syst* 6.3 (2011), 21:1–21:28.

[97] V.V. Semenov, I.S. Lebedev, and M.E. Sukhoparov. “Approach to classification of the information security state of elements for cyberphysical systems by applying side electromagnetic radiation”. In: *Sci. Tech. J. Inf. Technol. Mech. Opt* 18.1 (2018), pp. 98–105. URL: https://doi.org/10.17586/2226-1494-2018-18-1-98-105.

[98] V.V. Semenov, I.S. Lebedev, M.E. Sukhoparov, and K.I. Salakhutdinova. “Application of an autonomous object behavior model to classify the cybersecurity state”. In: *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*. 2019, pp. 104–112. URL: https://doi.org/10.1007/978-3-030-30859-9_9.

[99] M. R. Shahid, G. Blanc, Z. Zhang, and H. Debar. “IoT devices recognition through network traffic analysis”. In: *2018 IEEE International Conference on Big Data (Big Data)*. 2018, pp. 5187–5192.

[100] M. Shen, K. Ye, X. Liu, L. Zhu, J. Kang, S. Yu, Q. Li, and K. Xu. “Machine learning-powered encrypted network traffic analysis: A comprehensive survey”. In: *IEEE Commun. Surveys Tuts.* 25.1 (2023), pp. 791–824.

[101] Nathan Shone, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi. “A deep learning approach to network intrusion detection”. In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 2.1 (2018), pp. 41–50. DOI: 10.1109/TETCI.2017.2772792.

[102] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman. “Classifying IoT devices in smart environments using network traffic characteristics”. In: *IEEE Transactions on Mobile Computing* 18.8 (2018), pp. 1745–1759.

[103] Iman IM Abu Sulayman, Rongji He, Marlin Manka, Andrew Ning, and Abdelkader Ouda. “LiFi/WiFi Authentication and Handover Protocols: Survey, Evaluation, and Recommendation”. In: *2021 International Symposium on Networks, Computers and Communications (ISNCC)* (2021), pp. 1–6.

[104] Q. Sun, D. R. Simon, Y.-M. Wang, W. Russell, V. N. Padmanabhan, and L. Qiu. “Statistical Identification of Encrypted Web Browsing Traffic”. In: *Proceedings of the 2002 IEEE Symposium on Security and Privacy*. 2002, pp. 19–30.

[105] P. Velan, M. Čermák, P. Čeleda, and M. Drašar. “A survey of methods for encrypted traffic classification and analysis”. In: *Int. J. Netw. Manag.* 25.5 (2015), pp. 355–374.

[106] K Vengatesan, Abhishek Kumar, M Parthibhan, Achintya Singhal, and R Rajesh. “Analysis of Mirai botnet malware issues and its prediction methods in internet of things”. In: *International conference on Computer Networks, Big data and IoT*. 2018, pp. 120–126.

[107] Y. Wan, K. Xu, G. Xue, and F. Wang. “IoTargos: A multi-layer security monitoring system for internet-of-things in smart homes”. In: *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE. 2020, pp. 874–883.

[108] T. Wang and I. Goldberg. “Walkie-talkie: An efficient defense against passive website fingerprinting attacks”. In: *26th USENIX Security Symposium (USENIX Security 17)*. 2017, pp. 1375–1390.

[109] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng. “Malware traffic classification using convolutional neural network for representation learning”. In: *2017 International Conference on Information Networking (ICOIN)*. 2017, pp. 712–717.

[110] S. Xiong, A. D. Sarwate, and N. B. Mandayam. “Network traffic shaping for enhancing privacy in IoT systems”. In: *IEEE/ACM Transactions on Networking* (2022), pp. 1–16.

[111] Qiang Xu, Rong Zheng, Walid Saad, and Zhu Han. “Device fingerprinting in wireless networks: Challenges and opportunities”. In: *IEEE Communications Surveys & Tutorials* 18.1 (2015), pp. 94–104.

[112] Q. Zhu, C. Yang, Y. Zheng, J. Ma, H. Li, J. Zhang, and J. Shao. “Smart home: Keeping privacy based on air-padding”. In: *IET Information Security* 15.2 (2021), pp. 156–168.