UNICA

UNIVERSITÀ
DEGLI STUDI
DI CAGLIARI

Ph.D. DEGREE IN
ELECTRONIC AND COMPUTER ENGINEERING

Cycle XXXVI

TITLE OF THE Ph.D. THESIS

Advancements in Wi-Fi-Based Passenger Counting and Crowd Monitoring: Techniques and Applications

Scientific Disciplinary Sector(s)

ING-INF/03

Ph.D. Student: Lucia Pintor

Supervisor Luigi Atzori

Final exam. Academic Year 2022/2023
Thesis defence: February 2024 Session

Abstract

The widespread use of personal mobile devices, including tablets and smartphones, created new opportunities for collecting comprehensive data on individual movements within cities while preserving their anonymity. Extensive research focused on turning personal mobile devices into tools for measuring human presence. To protect privacy, the data collected must be anonymous or pseudo-anonymous, leading to the preference for management data. A common approach involves analysing probe requests, which are Wi-Fi protocol messages transmitted by mobile devices while searching for access points. These messages contain media access control (MAC) addresses, which used to be unique identifiers. To safeguard the privacy of smartphone users, the major manufacturers (Google, Apple, and Microsoft) have implemented algorithms that generate random MAC addresses, which change often and unpredictably. This thesis focuses on the problem of fingerprinting Wi-Fi devices based on analysing management messages to overcome previous methods that relied on the MAC address and became obsolete. Detecting messages from the same source allows counting the devices in an area, calculating their permanence, and approximating these metrics with the ones of the humans carrying them. An open dataset of probe requests with labelled data has been designed, built, and used to validate the experiments. The dataset is also provided with guidelines for collecting new data and extending it. Since the dataset contains records of individual devices, the first step of this study was simulating the presence of multiple devices by aggregating multiple records in sets. Many experiments have been conducted to enhance the accuracy of the clustering. The proposed techniques exploit features extracted from individual management messages and from groups of messages called bursts. Moreover, other experiments show what happens when one or more features are split into their components or when the logarithm of their value is used. Before running the algorithm, a feature selection was performed and exploited to improve the accuracy. The clustering methods considered are DBSCAN and OPTICS.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Knowing how people move in urban areas is critical for effectively deploying and managing many city services, such as transport mobility services, security procedures during crowded public events, and designing new public spaces. Several technologies have been exploited to collect relevant data to get valuable insights on the number of people that gather in different points of interest, the amount of time the people spend there, and how frequently people return. Controlling the customers' location helps to calibrate the services delivered to them. For example, a big group of people near a bus stop might trigger demand for more buses operating on that pathway. Sensing no people in an area for a long time might disable the public lights and save energy. Discerning the normal flows of people from anomalies helps promptly recognise emergencies or accidents and formulate timely responses to prevent disorders. Many of these methods may be invasive to people's privacy (e.g., recognition of the person by video or photo) or require people to pass through gates (e.g., radar placed at entrances or exits), or be tied to objects that must be carried by people specifically for tracking purposes (e.g., badges) (Singh et al., 2020). Devices sensing the crowding of an area should preserve privacy and be economically and energetically sustainable. These issues can be overcome in most scenarios by exploiting techniques that rely on analysing the messages broadcasted by devices people carry daily, such as smartphones. Collecting signalling data allows the processing of messages that devices send anyway, so there is no need to solicit them to send extra data that might degrade the performance of the network. Moreover, these messages have been designed to reach unknown receivers and do not contain personal data. In the rest of the introduction, we depict the background of this work, describe the main objectives of the thesis and outline the goals. The last section of this chapter shows a list of publications related to this thesis.

1.1 Motivation and Background

Crowd monitoring has become a key discipline in modern urban management. In an increasingly interconnected world, effectively monitoring and managing flows of people is crucial for ensuring safety and optimising resource allocation. Crowd monitoring involves tracking individuals' movement within a specific area to mitigate potential risks associated with overcrowding (e.g., stampedes and emergencies) while ensuring a smooth and organised flow of people (Shu et al., 2017). It can be applied to various contexts, like urban centres, public transportation hubs, events or touristic attractions (Yang et al., 2023). Many technological solutions have been developed in this field (Pintor et al., 2024).

Cameras are one of the most widely used technologies to monitor the flow of people due to their versatility. Appropriately placing cameras makes it possible to monitor very large areas compared to other types of sensors. On the other hand, it is necessary to use appropriate shape-recognition algorithms to identify people and their activities. The high accuracy of these methods made them the most popular technologies for monitoring crowd flows (Redmon et al., 2016) (Ilyas et al., 2019) (Sindagi and Patel, 2018). However, the privacy regulations established by many countries limit their usage if people are recognisable.

Thus, other technologies have become more popular for some specific applications. Some examples are radars (*radio detection and ranging*) and lidars (*light detection and ranging*), which are sensors to detect the presence of objects, including people, and measure their distance (ranging) (Skolnik, 2008). These sensors emit electromagnetic radiations that are reflected or absorbed by the surfaces of objects on their way. When objects reflect this energy, it usually generates an echo returning to the sensor. The time needed to reach the target object and return to the sensor is proportional to the distance of the object. Nowadays, many commercial radars and lidars implement people-counting functions. Radar applications for detecting and identifying human targets are currently topics of great interest in the scientific community because of the variety of use cases they embrace (e.g., autonomous driving, search and rescue operations, intelligent environments, etc.) (Jalalvand et al., 2019) (Zhao et al., 2019) (Günter et al., 2020) (Shackleton et al., 2010). While radars and lidars have strengths, they also have certain disadvantages in tracking people compared to other methods. Radars and lidars are often more expensive to install and maintain and typically require larger, fixed equipment. Moreover, raw data from radars and lidars can be complex and require sophisticated algorithms for interpretation, making them less straightforward for certain applications.

Therefore, given the widespread usage of *personal mobile devices*, including tablets, smartphones, and smartwatches, novel opportunities for gathering ex-

tensive data about people have appeared. As a result, many researchers utilised personal mobile devices to create indicators to measure human presence. Mobile devices produce considerable overhead traffic to connect with other devices and access networks. Considering the Wi-Fi and Bluetooth protocols, this traffic is usually management data emitted to transmit the requirements necessary to start communications. Since it does not contain personal information about the device owner, it can be used to characterise the source and track it. This approach assumes that each person carries one personal device which emits electromagnetic signals while its wireless interfaces (i.e., Wi-Fi interface and cellular antenna) are active. These electromagnetic signals can be collected and processed to extract features characterising the single device. This technique does not require users to take any specific action, such as installing a particular application on their device.

Of course, the data collected must be anonymous or pseudo-anonymous to protect users' privacy, so it is usually preferred to collect management data. Devices to collect Wi-Fi messages are called Wi-Fi sniffers and are usually very cheap and easy to install, but their range is limited to a few dozen metres indoors and just over 100 metres outdoors. In addition, if a tracked device leaves the monitored zone for a long time, it may happen that the acquisitions of its packets before and after leaving the monitored zone cannot be related. Thus, to have effective monitoring, it is necessary to have several sensors with slightly overlapping coverage to track the smartphones continuously and more precisely. This thesis focuses on using Wi-Fi management messages, detailed in the next chapter, to extract characteristics, or features, of the emitting device, create its fingerprint and track it. Many studies demonstrated the benefits of this approach; hence, the novelty of this thesis is the analysis of features which affect the accuracy the most.

Other technologies related to data collected by smartphones are available, such as *call data records* (CDRs) (Berlingerio et al., 2013) (Janecek et al., 2015) or data voluntarily shared by mobile device users (*crowdsourcing*). Yet, in the first case, buying this data from the mobile network operators (MNOs) who own CDRs is necessary. In contrast, in the second case, it is necessary to persuade people to install specific apps on their smartphones and accept sharing their data continuously. These two approaches are often not economically sustainable.

1.2 Scope of the Thesis

The focus of this work is the design of a methodology to evaluate the importance of all the major features that could be used to track devices through Wi-Fi frame sniffing and to compare alternative clustering algorithms. There are many challenges. First, these messages have no identifiers, so a dataset with real ground truth is needed to validate the experiments properly. Other crucial steps are extracting

information from the raw messages, selecting the best-performant features, choosing the clustering algorithms, and calibrating them.

The work of this thesis can be summarised in the five phases depicted in Table 1.1: i) Documentation, ii) Dataset realisation, iii) Scenario synthesis, iv) Feature extraction and analysis, and v) Comparative analysis. The first phase was concentrated in the initial six months but always accompanied the subsequent phases. The in-depth study of the standard was crucial in identifying the research topic, which is why Chapter 2 is dedicated to it. The literature search was also the basis for identifying a gap in the state of the art, described in Chapter 3, which directed our research towards creating a dataset of management messages with ground truth to provide a tool for the validation of results and experimenting with methods to increase the accuracy.

Our decision to publish the dataset in an open format and a framework to replicate the process is fuelled by the need for further methods to test people counting and tracking systems based on analysing Wi-Fi packets. Indeed, most of the previously published works considered the number of people as the objective for the counting system (Determe et al., 2022) (Nitti et al., 2020) (Uras et al., 2020b). However, this is insufficient because unknown devices can be in the area (e.g., access points), or even though the final count of devices is correct, the groups of messages generated by the clustering algorithms might not reflect the real classification according to the originating device. A study in which the obtained results can be compared to the exact labelling helps to develop efficient clustering algorithms and to identify paths for improvements. Thus, the innovation of our dataset lies in capturing data from devices placed in isolated environments so that one is certain of the source of each message, as described in detail in Chapter 4.

The next step is the simulation of real scenarios by fusing labelled data originating from different devices. Timestamps of management messages are modified by adding an offset to make them contemporary. A separate file lists the MAC addresses of each device to keep track of labels. The scenarios thus realised are useful to study which features of these messages are most useful to group them correctly according to their sources. A detailed analysis of the methods for extracting valuable information to characterise the sources of the messages in the dataset is presented in Chapter 5. Various experiments, illustrated in Chapter 6, were conducted with two different clustering algorithms to determine the optimal set of features and input parameters. The algorithms considered are *DBSCAN* (density-based spatial clustering of applications with noise) and *OPTICS* (ordering points to identify the clustering structure). Their performance is compared with different sets of features and input parameters. Some features are obtained by separating the message fields into their components and calculating their values' logarithms. Results are shown as tables and discussed, highlighting the strengths and flaws of each approach.

Phase	Description
Documentation	The first step in this work was an in-depth study of the Wi-Fi standard and methods for counting and tracking people via management messages.
Dataset realisation	The lack of datasets with a real ground truth led us to define criteria for collecting this data and a framework to replicate the process and extend our open source dataset.
Scenario synthesis	Since each file in the dataset contains data collected from a single device, they were merged to simulate the presence of multiple devices contemporaneously.
Feature extraction and analysis	Merged captures are converted into data frames containing only useful information. The feature importance is calculated to discard the less influential data.
Comparative analysis	Clustering methods are compared with different features and parameters.

Table 1.1: Summary of the phases of the doctoral career

Chapter 7 discusses the experimentation conducted with the University of Rennes 1 (France), with whom we collaborated to create a system that fingerprints Wi-Fi signals at the physical level to characterise transmitting devices. As this is a different topic from the rest of the thesis, an introduction and conclusions specific to the chapter are presented. The concluding chapter, Chapter 8, summarises the main contributions and presents some of the possible directions that could be explored in the future.

1.3 Outline and Main Results of the Thesis

This thesis aims to define methods to evaluate the accuracy in identifying the source of Wi-Fi devices by analysing the content of management messages. Management messages are characterised by a time reference and the receiving power level related to the receiver distance. These pieces of information are helpful in studying when these devices are in a monitored area and how far they are from the sensing device, so grouping messages with the same source allows for studying when a device entered and exited a specific area or moved from a monitored area to another. This makes it possible also to count the devices in an area, calculate their permanence, and approximate these metrics with the ones of the humans carrying them. Previous methods became obsolete because of the continuous evolution of the Wi-Fi standard, shown in Figure 1.1, which implemented new features to preserve privacy.

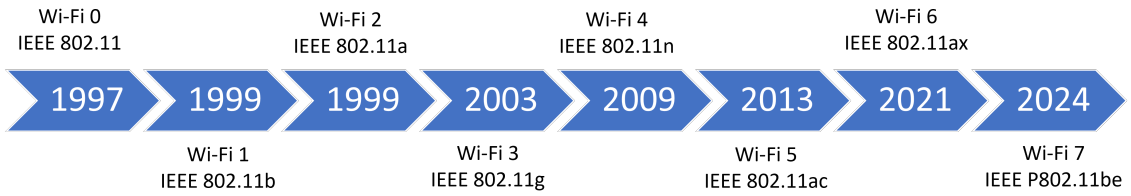


Figure 1.1: Evolution of the Wi-Fi standard

The contributions of this thesis include the design and realisation of a database of management messages from various mobile devices and the analysis of key elements enabling the classification and counting of users. In order to make our work available to the community, the collected data is publicly available as an open-source dataset, and the code to replicate the experiments can be found on the GitHub platform. This effort addresses the frequently encountered obstacle of the need for open and accessible databases. We used the dataset to validate the experiments because it contains records of individual devices, so labels referring to the originating device could be created.

The next step of this study is simulating the presence of multiple devices by aggregating multiple records of individual devices in sets. After that, capture files are converted into Python structures to analyse them with machine-learning tools. So, each sample is transformed into an array of features containing the values of the main fields of the management messages. Features are then ranked by importance using a Random Forest algorithm to discard the less influential data in analysing single messages and *bursts*, which are groups of adjacent messages where the same randomised MAC address is kept. Details on these features are provided in Chapter 5. Many experiments have been conducted to enhance the accuracy of the clustering. Some of the proposed techniques show what happens when one or more features are split into their components or when the logarithm of their value is used. Finally, chapter 7 is an initial proof of concept for radio frequency identification based on raw samples without decoding.

1.4 List of Pubblications

This thesis is based on the following publications:

- L. Pintor and L. Atzori. *A dataset of labelled device wi-fi probe requests for*

mac address de-randomization - 2021, 2021. Open Source Dataset available on Mendeley.

- L. Pintor and L. Atzori. *A dataset of labelled device wi-fi probe requests for mac address de-randomization.* Computer Networks, 2022.
- L. Pintor and L. Atzori. *Analysis of wi-fi probe requests towards information element fingerprinting.* GLOBECOM 2022 - 2022 IEEE Global Communications Conference, 2022.
- L. Pintor, M. Uras, G. Colistra, and L. Atzori. *Monitoring People's Mobility in the Cities: A Review of Advanced Technologies.* Springer Nature Switzerland, 2023.

The thesis is also based on experiments described in a paper still in submission, whose title, authors and journal are documented below:

- L. Pintor and L. Atzori. *Crowd-Monitoring through Wi-Fi Frames Fingerprinting: Importance-based Feature Selection and Extensive Performance Analysis.* IEEE Transactions on Cognitive Communications and Networking.

Chapter 2

IEEE 802.11 Wi-Fi

Wi-Fi, short for "Wireless Fidelity," is a widely used technology that allows mobile devices that embed this technology to connect to the Internet and communicate with each other wirelessly. It is a family of standards that have been defined by the *institute of electrical and electronics engineers* (IEEE) to implement *wireless local area networks* (WLANs). IEEE 802.11 uses various frequencies, including 2.4 GHz and 5 GHz frequency bands. The 2.4-GHz band has between 11 and 14 channels, depending on the geographical area. Channels are spaced 22 MHz apart but overlap, resulting in only three non-overlapping channels: 1, 6, and 11. The 5-GHz band presents more channels and their bandwidth is variable. In the context of this thesis, only the 2.4-GHz band is considered. Management messages are crucial for this communication protocol because they are used for various purposes related to network control and connectivity. Management messages contain characterising information about the transmitting device and its connectivity, allowing for fingerprinting and tracking as long as it remains in the monitored area. The activity of passively intercepting data travelling through a network is called sniffing and can be performed through Wi-Fi sniffing systems (or sniffers), which capture and inspect data packets exchanged over Wi-Fi channels. Collecting management messages might provide insights into the interaction between the humans carrying the mobile devices and the environment (Li et al., 2020) (Tan and Gary Chan, 2021) (Vattapparamban et al., 2016). The rest of the chapter highlights the interaction of Wi-Fi standard with the layers of the open systems interconnection model, discusses different types of management messages, describes how research has evolved after the spread of *media access control* (MAC) address randomisation algorithms, and delves into some of the fields of probe requests and some aspects regarding how these messages are transmitted. The last section presents how these technologies are considered from the point of view of privacy regulations.

2.1 The Open Systems Interconnection model

The *open systems interconnection* (OSI) model (ISO/IEC JTC 1, 1994) is a conceptual model from the *international organization for standardization* (ISO) that standardises the functions of a communication system into seven abstraction layers that have the purpose of facilitating the interoperability and communication between systems and devices. Each layer in the OSI model performs specific functions and interacts with adjacent layers, from the physical implementation of transmitting bits across a communications medium to the highest level of applications. The seven layers of the OSI model can be summarised as:

- The *physical* layer (or layer 1) is concerned with the transmission and reception of raw unstructured bits over a physical medium.
- The *data link* layer (or layer 2) is responsible for the reliable transmission of frames between devices over a physical medium.
- The *network* layer (or layer 3) manages logical addressing, routing, and data forwarding between devices on different networks.
- The *transport* layer (or layer 4) ensures reliable end-to-end communication by providing error detection, flow control, and data segmentation and reassembly.
- The *session* layer (or layer 5) establishes, maintains, and terminates sessions (or connections) between applications on different devices.
- The *presentation* layer (or layer 6) translates data between the application layer and the lower layers.
- The *application* layer (or layer 7) provides network services directly to end-users or applications.

Although the OSI model is not strictly adhered to in practice, it remains a valuable reference for understanding network protocols and their interactions. This model allows communications between two parties through *protocol data units* (PDUs), which assume different denominations depending on the layer, as shown in Figure 2.1. The original data from the application layer is encapsulated with some additional header in each layer until it reaches the physical layer. After that, it is transmitted through physical media to another device, which can be an intermediate or the destination node, and layer by layer, the headers are decrypted to access the information and process it.

In this context, the physical and data-link layers are of particular interest for the work in this thesis because the Wi-Fi standard defines one of their protocols. Wi-Fi operates wirelessly, depicting the physical characteristics of the communication medium, so during the last decades, it has changed many times

	Layer	Protocol data unit
7	Application	Application data
6	Presentation	Presentation data
5	Session	Session data
4	Transport	Segment
3	Network	Packet
2	Data link	Frame
1	Physical	Bits

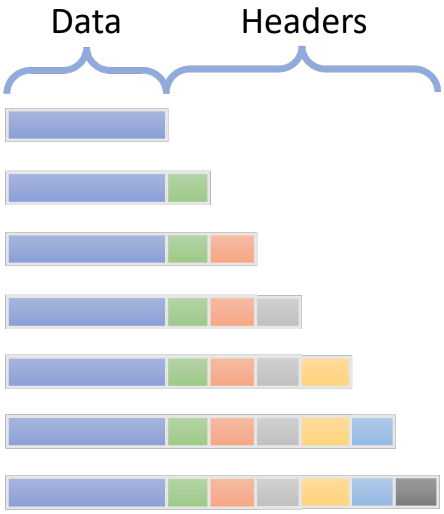


Figure 2.1: Layers of the OSI model

to adapt to different frequency bands, modulation techniques, and data rates for wireless communication. The data-link layer is also involved in the evolution of the Wi-Fi standard since its sublayers, the *logical link control* (LLC) and the *media access control* (MAC), manage the flow control and the access to the media. The many versions of the Wi-Fi standard added new procedures to share the medium, encrypt data and authenticate users, so these sublayers have also been affected. Moreover, new security measures have been deployed to handle issues that have emerged in the previous versions of the standard.

2.2 Management Messages

Section 9.3.3 of the last Wi-Fi standard (IEEE, 2021) defines a set of MAC frames as sequences of components in a specific order. The messages a mobile device constructs and decodes are determined by the functions supported by that particular device. As shown in Figure 2.2, a MAC frame always has a frame check sequence (FCS), a MAC header, and a variable-length frame body. While the MAC header has a fixed structure, the frame body is variable and contains information specific to the frame type and subtype. The key to decoding the frame body is reading the FCS field: the first octet of this field indeed uses two bits for the protocol, two bits for the type of frame, and four bits for the subtype of the frame.

The three frame types are control, data, and management. Each of the frame types has several defined subtypes. The frames related to the connection of a mobile device to an access point are all management messages, as shown in Table 2.1. The type is coded in the third and second most significant bits of

the FCS field, whereas the subtype is coded in the following four bits. *Beacon frames* are continuously broadcasted by access points (APs) at regular intervals to announce the presence of a Wi-Fi network. *Probe requests* (PRs) are transmitted on every channel by mobile devices to request information from APs in their proximity (Fenske et al., 2021). *Probe responses* are sent by APs in response to probe requests. *Authentication requests* are sent by the device that wants to join a scanned network. On reception of the authentication frame, APs send an acknowledgement and then an *authentication response*. Authentication frames aim to validate the device type, in other words, verify that the requesting device has the necessary capabilities to join the network. *Association requests* are sent by the mobile device in the association phase after the authentication succeeds. *Association responses* are sent by APs in response to association requests.

According to Section 11.1.4 of the last Wi-Fi standard (IEEE, 2021), the connection process can be started either by a mobile device (active scanning) or an access point (passive scanning), as shown in Figure 2.3. In *passive scanning*, the mobile device listens to each channel scanned for a while and returns information on all beacon frames received. After scanning one channel, the device initiates scanning in another until it scans all indicated channels. If among the received beacon frames, there are some emitted by known access points, the mobile device sends a probe response to that specific AP to connect to it. The AP answers with an acknowledgement message.

On the contrary, in *active scanning*, the mobile device broadcasts probe requests on every channel to discover available APs. Access points within range respond with a probe response frame, advertising the wireless network name, supported data rates, encryption types if required, and other 802.11 capabilities of the AP. If multiple APs answer, the mobile device selects the most suitable one. Whatever the type of scan, the connection procedure continues with the authentication and association phases. Thus, the mobile device sends an authentication request to the selected AP, the access point responds with an authentication reply, the mobile device sends an association request frame to the access point, and finally, the access point replies with an association response.

Among all these messages, probe requests are usually collected because they are transmitted without encryption by mobile Wi-Fi devices, even in the absence of APs, and contain useful information to identify their sources (i.e., MAC address, supported data rate, and vendor-specific fields). Sniffing these frames requires a Wi-Fi antenna that supports monitor mode and specific software to capture packets. One of the main critical issues is that mobile devices usually send probe requests in rotation in different channels to increase the probability of finding an AP. A sniffer collects packets transmitted in the channel into which it is tuned and partially collects those of neighbouring channels. For this reason, a good practice

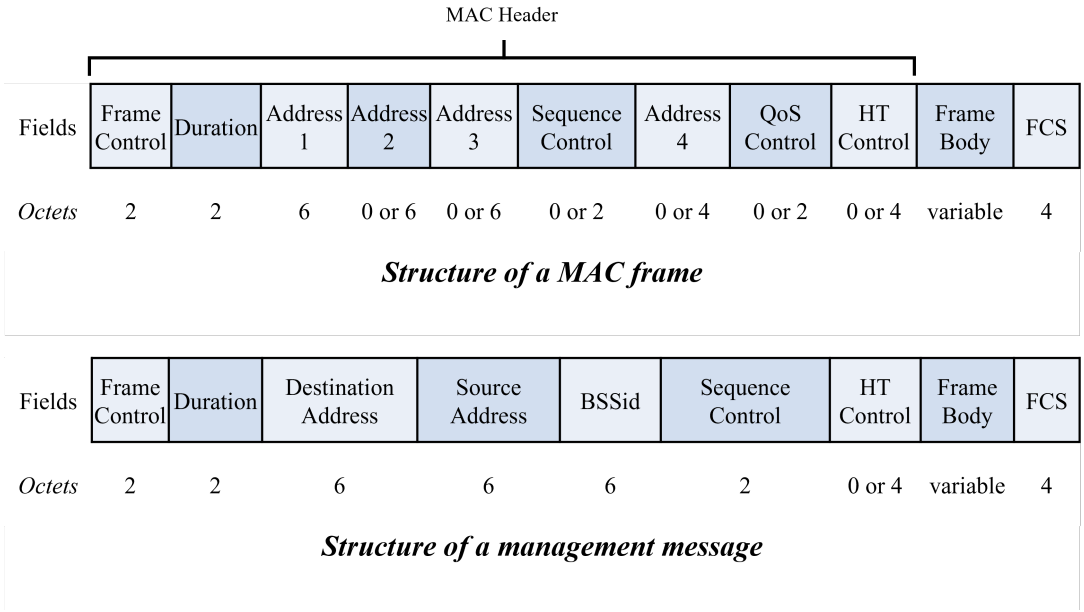


Figure 2.2: Comparison between a general frame and a management message

Type B3 B2 of FCS	Type description	Subtype B7 B6 B5 B4 of FCS	Subtype description
00	Management message	1000	Beacon
00	Management message	0100	Probe request
00	Management message	0101	Probe response
00	Management message	1011	Authentication
00	Management message	0000	Association request
00	Management message	0001	Association response

Table 2.1: Main frames involved in connecting a mobile device and an access point

is to use a sniffer with three antennas tuned into three non-overlapping channels.

2.3 MAC Address Randomisation

In traditional Wi-Fi communication, devices typically use a fixed source MAC address, their *factory* address (Dagelić et al., 2019). This made tracking and identifying them relatively easy in the past (Cunche, 2014). This procedure was used to detect the presence of personal devices by observing the unique MAC addresses in the captured traces and consequently estimate the number of people in a given area (Di Luzio et al., 2016). However, modern devices implement algorithms to *randomise* and change it frequently and unpredictably depending on the connection status (connected to an AP or not), vendor, model, and

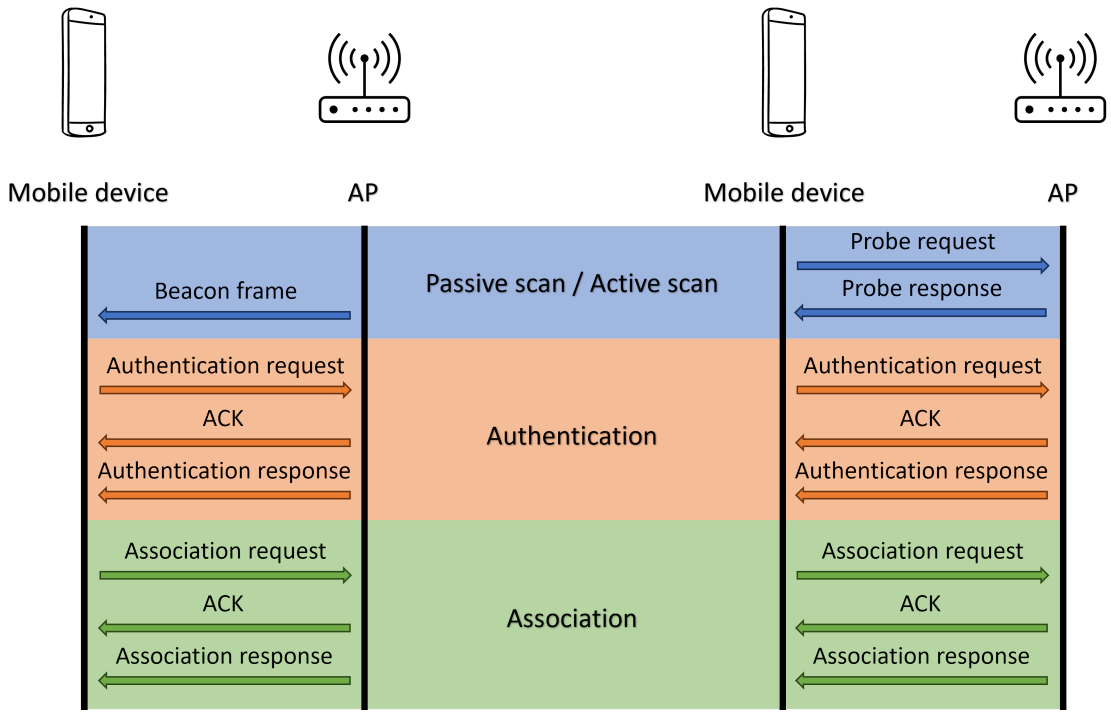


Figure 2.3: Procedure to connect a mobile device to an access point.

operative system (OS) (Vanhoef et al., 2016) (Li et al., 2019). Hiding factory MACs has become necessary because, even though this address does not contain any personal information, it might be linked to personal information through data cross-checking. Most modern mobile OSs contrast these privacy breaches by avoiding transmitting unnecessary information (i.e. SSID fields are often empty), sending probe requests less frequently, and using random MAC addresses. MAC address randomisation algorithms might differ depending on the OS installed in the device: a new MAC address might be assigned every time the screen is turned off, at regular intervals, or when the user does not interact with it for a while.

Therefore, a good practice of modern algorithms for grouping probe requests originating from the same source device is separating factory and random addresses and processing them separately (Uras et al., 2020b). According to the IEEE 802.11 standard, the discrimination between factory and random addresses is based on the analysis of the seventh bit of each MAC address (IEEE, 2021): i) if this bit is set to 0, then the address is *globally unique* and is used as a source address by devices that do not perform randomisation algorithms; ii) whereas, if this bit is set to 1, then the address is *locally administered*, meaning that the MAC address is configured via software. Figure 2.4 shows the structure of a MAC address, which is composed of two parts, the *organisational unique identifier* (OUI) and the *network interface controller* (NIC). A MAC address comprises 48 bits, or 6 octets. If the

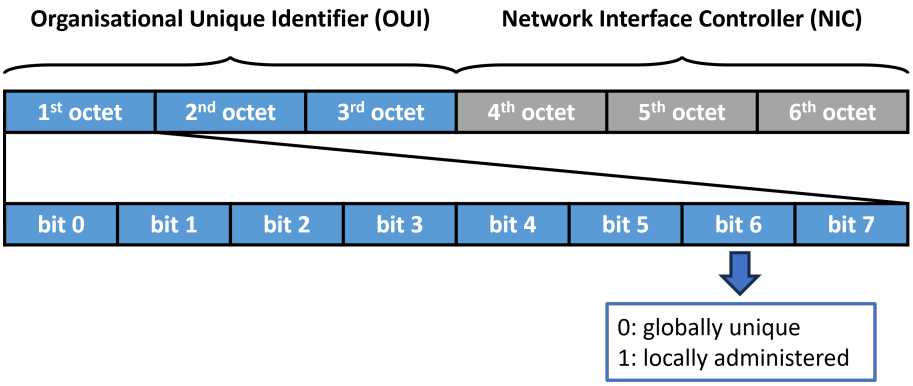


Figure 2.4: Structure of a MAC address.

MAC address is globally unique, the first three octets are the OUI, a code the IEEE assigned to identify the interface manufacturer. The last three octets are the NIC, an identifier the manufacturer assigns to its product. If a device uses its factory MAC as its source address, then the MAC address can be considered an identifier, as it is unique and constant. On the contrary, this is not true for random addresses because grouping all the MAC addresses generated by a single device is not trivial.

MAC address randomisation challenges the counting algorithms that use MAC addresses as device identifiers. This functionality led to an evolution of these kinds of algorithms that now perform additional steps to group the messages that might have been produced from the same source by analysing valuable characteristics or features of the sniffed messages (Oliveira et al., 2019) (Uras et al., 2020a) (Vega-Barbas et al., 2021). This clustering does not compromise users' privacy because the factory MAC cannot be reconstructed. Moreover, once the tracked device moves away from the sniffer, it might not be linked again to probe request streams collected previously. Most algorithms were validated by comparing the number of probe request clusters counted by each algorithm and the ground truth of the number of people in the observed area. This comparison has some flaws because the number of people might differ from the number of devices. Moreover, although this comparison demonstrates good results, individual probe requests might not be clustered correctly.

2.4 Information Elements

Due to the widespread randomisation of MAC addresses, searching for other information that could characterise the messages transmitted by the same device is necessary. In this thesis, we considered the components of the *frame body*, which is a variable-length field of the management messages, as shown in the lower part of Figure 2.2. This field includes the *information elements* (IEs), also known as

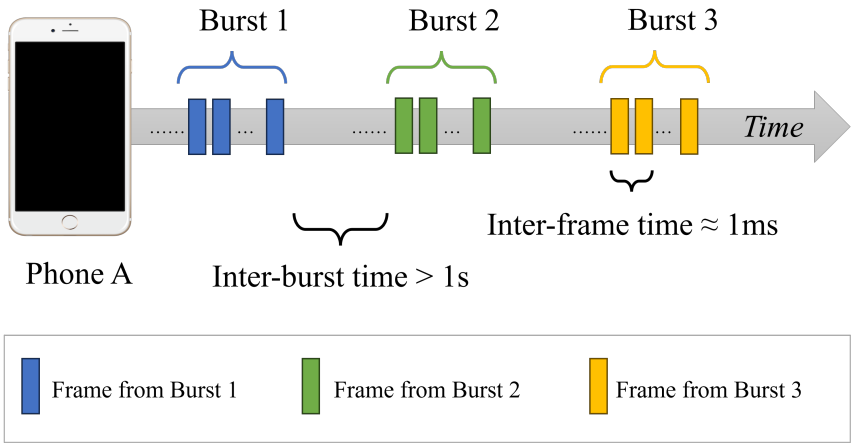


Figure 2.5: Example of a device sending bursts of probe requests

tagged parameters or just elements, which are fundamental components of Wi-Fi frames. Each IE is identified by a unique *information-element identifier* (IE ID) and contains specific data elements. The content of an IE varies depending on its purpose. Some examples are lists of supported rates, the transmission channel, and transmission capabilities (e.g., the channel width, supported modulation and coding scheme). IEs are detailed in Section 9.4.2 of the last version of the Wi-Fi standard (IEEE, 2021). In Chapter 6 of this thesis, we demonstrate that a carefully selected set of IEs makes it possible to create a fingerprint and thus catalogue messages according to the device that emitted them.

2.5 Bursts of Frames

As mentioned above, several PRs are usually transmitted in close succession; these messages can be aggregated in groups called *bursts*. Frames within the same burst have an inter-time shorter than $1ms$, while the distance between two bursts originating from the same device is over 1 second. The time necessary to send a complete burst is shorter than $10ms$, whereas the inter-burst time is longer than $1s$ (Matte et al., 2016), as shown in Figure 2.5. Furthermore, all the frames of the same burst have the same MAC address, while frames of different bursts usually have a different MAC if the transmitting device performs the randomisation. The fact that the MAC address does not change within the burst makes it easy to aggregate these messages and calculate further metrics. In particular, the inter-timing between PRs of the same burst can be studied, or errors in specific fields can be corrected using the fashion or median of the values.

2.6 Privacy Regulations

One of the reasons why interest in counting and tracking people via management messages has grown is due to privacy constraints that limit other technologies. Notably, in 2018, the most restrictive privacy regulation was adopted in Europe, the *general data protection regulation* (GDPR) (Eur, 2018). GDPR gives individuals greater control over their personal data and imposes strict rules on organisations and businesses that collect, process or store it. Personal data refers to any information that allows for direct or indirect identification of a person (e.g., a name, an identification number, or an online identifier). GDPR also has a global impact, as many non-EU businesses that handle EU citizens' data must comply with its regulations. The law strongly emphasises transparency and responsible data handling.

However, if the identification of a person requires additional information, it is considered as *pseudonymisation*. Although a factory MAC address is a unique identifier, it is not personal data because it is tied to a device. Without knowing the relationship between the device and its owner, it is nearly impossible to identify a certain individual with the MAC address only. If randomisation of the MAC is applied, the device more or less frequently changes the address it transmits in probe requests, making it almost impossible to identify the device owner. To completely remove any breach of privacy that might occur with the issuance of new regulations, MAC addresses can be anonymised by using non-reversible encryption within the sniffer itself so that this information is neither transmitted nor stored. The experiments presented in this thesis use a public dataset in which the names of device owners do not appear, so it was not necessary to anonymise the data.

Chapter 3

State of Art

Several works exploit the analysis of Wi-Fi traffic to count or track people. Yet, the randomisation of the MAC address still limits these methods. One of the papers that pioneered in this research field is Vanhoef et al. (2016), which demonstrated that PRs contain enough information to perform tracking, even though the MAC address changes periodically. This work models a tracking algorithm that fingerprints mobile devices depending on the IEs. The same authors proposed additional metrics based on the arrival time to group frames into bursts with an incremental learning algorithm (Matte et al., 2016). However, this work still has flaws, such as the used dataset was collected in 2013, before MAC address randomisation was first implemented. The dataset was indeed anonymised before the publication, but the authors of the dataset were not the same as the paper, so the ground truth might not be reliable.

Among the fields of PRs, the *service set identifier* (SSID) used to be a useful source of information because, in the past, mobile devices were filling this field with the names of their *preferred network list* (PNL). This vulnerability has been exploited to fetch the provenance of the crowds and their relationships (Di Luzio et al., 2016) (Cunche et al., 2012), to create fake access points and perform attacks (Vanhoef et al., 2016) (Rusca et al., 2023). Nowadays, the SSID is usually blank because it allows malicious attackers to create networks with the names of known SSIDs and exploit this vulnerability to hijack or manipulate the traffic of unaware users.

The last trend is based on passively collecting management messages and clustering them with unsupervised Machine-Learning algorithms. Table 3.1 compares some methods from the literature, which mostly rely on the information elements and other frame features stable for frames coming from the same source. Some methods have a time window for the acquisition shorter than 1 minute (Hao et al., 2023), (Determe et al., 2022), (Simončič et al., 2023a), while others have longer ones that last 5 minutes (Trasberg et al., 2021) or more than 10 minutes

Reference	Method	Validation comparison	Average Error
Hao et al. (2023)	Neural Networks	Manual count	13.44%
Determe et al. (2022)	Statistical estimator	Count of a video-processing algorithm	12%
Trasberg et al. (2021)	Dynamic regression model	Data about sales in shops and restaurants	18.4% and 6.2%
Furuya et al. (2021)	Sequential algorithm	Manual count	1.58%
Tan and Gary Chan (2021)	Cost function	Label of the Probe Request	20%
Simončič et al. (2023a)	Clustering with OPTICS	Manual count	4%
Uras et al. (2022)	Clustering with DBSCAN, OPTICS, and HDBSCAN	Manual count	10%
Our method	Clustering with DBSCAN and HDBSCAN	Label of the Probe Request	7.5%

Table 3.1: Comparison with related works

(Uras et al., 2022). However, methods with shorter time windows also exploit information about the received signal strength to aggregate messages originating from the same device. These approaches are valid in situations where the people are waiting or staying in the same location but might not be optimal while considering people in transit. Moreover, only some approaches (Trasberg et al., 2021) (Simončič et al., 2023a) (Uras et al., 2022) use public open datasets (Simončič et al., 2023b) (Mohorčič et al., 2023) (Pintor and Atzori, 2021) (London, 2019), allowing for the reproducibility and comparison of results. Datasets with labeled samples also enhance the validation that is no longer based on the manual count or an alternative counting system that might be affected by errors. Summarising, two limitations characterize these works: i) the performance is provided by applying the algorithm on traffic traces that are not publicly available and that are difficult, if not impossible, to reproduce; ii) an extensive analysis of the importance of the feature to drive their selection to fingerprint has not been performed.

3.1 Counting

The simplest application for the management data is the count people, which involves detecting and characterising the changes in the environment or the signals emitted by different devices. Sensors should cover the entire region to be monitored. The *channel state information* (CSI) analysis can be used to detect changes in Wi-Fi signal features caused by user presence. Machine learning models such as *neural*

networks (NNs) can be trained to correlate signal patterns with user counts (Zhang et al., 2021) (Cheng and Chang, 2017). Other approaches detect the Wi-Fi devices of people moving through the area and elaborate the messages they broadcast to create a transmitter signature and distinguish individual devices from one another (Oliveira et al., 2019) (Yang et al., 2023). The counting algorithm must consider factors such as device mobility, signal strength, and potential fluctuations in the number of devices connected to Wi-Fi (e.g., due to people entering or leaving the area). Finally, the accuracy can be evaluated manually or by comparing the results with other reliable counting methods. The technique presented in this thesis allows labels to be assigned to all Probe Requests, so it is possible to count people by counting the labels assigned in a given time range (Pintor and Atzori, 2022b).

3.2 Localisation

Localisation techniques are usually based on multilateration and ranging methods. Distances are calculated through the *received signal strength indicator* (RSSI), a metric to quantify the power level of the received signal from a transmitting device which has a mathematical relationship with the distance between transmitter and receiver (Jianwu and Lu, 2009) (Luo and Hsiao, 2019). Some approaches are based on mapping the monitored area: they generally create a database of reference RSSI values during the training (survey phase) by manually collecting signal data at different known locations, then, when a device is to be localised (query phase), the RSSIs measured by that device are compared with the reference values in the database. The system finally estimates the device location based on the closest match or through machine-learning techniques. However, radio maps are highly susceptible to environmental modifications and their performance drops when the number of devices to localise increases (Qureshi et al., 2019) (Chabbar and Chami, 2017) (He and Chan, 2016). Moreover, experimental results shown in Tonggoed and Panjan (2022) indicated that the accuracy of the proposed system is about 2.4 m, so the authors suggest combining this approach with the data from other sensors. Similar applications localise devices with other technologies like frequency modulation (FM) (Mukhopadhyay et al., 2017), Bluetooth low energy (BLE) (Thaljaoui et al., 2015), and Zig-Bee (Arif et al., 2018). The technique presented in this thesis does not consider the distance of the mobile devices from the sensors because all dataset samples were collected by keeping constant the distance between the sensor and the sensed devices.

3.3 Tracking

Generally speaking, tracking relies on detecting, estimating and recording a succession of locations. The route of a person is stored as a series of geographic locations

ordered over time, allowing for user profiling. Identifying the habitual paths allows to understand common origins or destinations and points of convergence joined to many trajectories. The main issue is privacy protection, which leads to a preference for anonymous tracking systems such as those based on management-message analysis. Following this approach, it is necessary to distribute the sensors appropriately because if the devices leave the monitored area for long periods, then it is no longer possible to reconstruct the entire path. The size of each area monitored by a single sensor depends on the sensitivity of the antenna and the presence of obstacles in the line of sight (Determe et al., 2022). An *origin-destination* (OD) matrix can be calculated if multiple sensors are associated with *points of interests* (POIs) (Uras et al., 2020b) (Nitti et al., 2020) (Traunmueller et al., 2018): this matrix is a data representation used in transportation analysis to illustrate the flow of people or goods between different origins and destinations within a given network or region. It presents a comprehensive snapshot of movement patterns, showing the volume of trips or transfers between specific starting points (origins) and ending points (destinations). The technique presented in this thesis can be applied to data collected by a sensor network. By augmenting the data with information about the sensor that collected it, it is possible to cluster all the messages generated by a mobile device, assign it the same label and determine its path by combining the positions of the sensors, which become the anchors of the system.

3.4 Device Fingerprinting

Fingerprinting a device involves measuring its characteristics or something related to it, such as the signals it emits. Various pieces of information can be extracted from management messages, but not all are equally useful to distinguish messages from different sources (Potortì et al., 2016) (Togashi et al., 2016). In the past, a widely used technique to identify devices was to use their factory MAC addresses, which are globally unique identifiers. However, with the spread of randomisation algorithms, identifying all devices this way is no longer possible (Martin et al., 2017). Thus, it is necessary to consider other information contained in the Wi-Fi messages, such as IEs (Vega-Barbas et al., 2021) (Vanhoef et al., 2016) (Pintor and Atzori, 2022b), the Wi-Fi protected setup (WPS) (Fenske et al., 2021) (Martin et al., 2017), or the sequence number and statistics related to the burst (Uras et al., 2020b). The IE analysis approach is probably the most popular because it is entirely passive and realises a packet signature by considering the content or size of information elements in management messages. On the other hand, the WPS-based methods involve realising fake access points so that when the device tries to connect to them, additional information is collected to retrieve the factory MAC address, performing the *derandomisation*. These methods obviously cannot be used legally in contexts other than academic experiments, but they have raised interest in the possible privacy violations inherent in the Wi-Fi protocol. Experiments that rely on burst statistics

instead produce new metrics related, for example, to the packet transmission frequency and the number of packets sent in the same burst. It is possible to measure these quantities because the packets in the same burst keep the same MAC address and can be easily grouped. Other approaches are based on analysing Wi-Fi signals at the physical level and will be explored in more detail in Chapter 7.

Chapter 4

A dataset of Labelled Device Probe Requests

Our dataset was designed to provide detailed information about the device which had emitted the probe requests to allow the analysis of the behaviour of each model separately. The dataset contains captures taken according to a standard procedure, not specifically designed for using specific algorithms. However, in the next chapter, we present one of the possible applications, namely the selection of particular parts of each sample to improve the accuracy of a clustering system based on unsupervised machine-learning algorithms. The need for this kind of dataset depends on the fact that even though many probe request datasets are accessible and open-source (Ferrara et al., 2020) (V. Barbera et al., 2013) (Robyns et al., 2017), none of them considers devices separately. Distinguishing a device from another in an unknown environment might be unfeasible because there are no unique identifiers: IP addresses are not defined in probe requests, and MAC addresses might be randomised. Additionally, not knowing the environment and the position of each device restricts the usage of power thresholds for discriminating devices. These aspects also highlight the need for datasets with labelled Wi-Fi probe requests that were missing in the literature. The collection of probe requests from multiple devices that implement the randomisation of MAC addresses is complex because we cannot discriminate against more than one device at a time, even using power threshold filters. The power level fluctuates depending on the distances between the source-emitting channel frequency and the sniffer-detecting channel frequency. Moreover, unpredictable noise and the structure of the environment might affect this measure. Due to this fact, considering different thresholds for more than one device might lead to errors.

In order to get labelled probe requests, a simple solution is to collect them separately for each device and then simulate the presence of multiple devices simultaneously by modifying the timestamp of each probe request. This method generates synthetic captures of labelled probe requests with multiple devices.

Having data similar to the one we can collect in real time can be helpful for testing and training algorithms that directly count the number of people in an area. However, our dataset is not appropriate for algorithms based on power threshold (i.e., counting the number of devices or calculating the relative locations of each source) because the original captures are collected in similar environmental conditions (same distance from the sniffer) and particular power thresholds have been already used to filter data. Our dataset might be used to achieve different purposes related to probe request analysis. Some use cases of this dataset might be: i) assessment of the performance of clustering algorithms; ii) training of new ML-based algorithms to improve clustering performance; iii) analysis of other elements such as the information elements, sequence numbers, or burst structures. In the rest of the chapter, we present the devices used for data collection, their configuration, the collection and filtering process, and how we used the dataset to simulate realistic scenarios.

4.1 Sensing Devices and Configuration

The dataset was acquired with a Raspberry Pi 3 (Model B+) with three additional Wi-Fi interfaces. The Wi-Fi interface embedded in the Raspberry does not support the monitor mode, which is necessary to collect probe requests, so it was necessary to use extra antennas that can be powered through the embedded USB ports of the Raspberry. We selected three low-consumption antennas that can be directly plugged into the Raspberry Pi to inspect three channels contemporaneously. These antennas are the same model, Realtek RTL8188CU, and support the 2.4 GHz bandwidth and various modes, including the monitor mode. The minimum signal power that can be detected is -110 dBm. The Raspberry Pi was configured with a Raspbian Buster OS where the Wireshark library was installed. The sniffing script is written in Python language and performs the following operations: it configures the monitor mode in all interfaces, sets them to specific channels (namely 1, 6, and 11 at the 2.4-GHz band), and starts the data acquisition. The script is available in the public GitHub repository "WiFi-Sniffer". The data collected from each interface is saved in a different file. The sniffing script is composed of two parts: i) interface configuration (configure_interfaces.py) and ii) a sniffing sub-process launcher (start_sniffing.py). Further details are provided in the code documentation (Pintor and Atzori, 2022a). The files in the dataset are in packet-capture (PCAP) format and contain only probe requests from a device at a time. All packets originating from the sniffer were removed.

4.2 Data Collection

The dataset comprises 315 capture files and some tables to match each capture with its source device. Since each file takes a few KB, the whole dataset weighs 1.44 MB. A total of 22 devices were analysed: 8 were observed in an anechoic chamber and 14 in an unshielded environment. The main feature of the dataset is its subdivision by device, which allows a more accurate analysis of the behaviour of individual devices in different modes. Moreover, the data can be labelled to train machine learning algorithms or verify the correct functioning of algorithms that aim to count devices through probe requests in the presence of random MAC addresses. A complete list of the analysed devices is provided in 4.1 (extract of devices.csv file in Pintor and Atzori, 2021). Most of these devices use Android OS (17) and the others iOS (5).

Part of the data was collected in the anechoic chamber of the Department of Engineering of the University of Cagliari by placing only the sniffer and a smartphone inside for each capture. Before the data collection, we verified that the anechoic chamber shielded any Wi-Fi communications to ensure each file contained probe requests from a single device. We performed a background capture with the door closed, placing only the sniffer inside. The only captured packets were emitted by the embedded interface of the sniffer, easily identifiable because it was using its factory MAC address. We repeated the same experiment, keeping the door of the anechoic chamber open. We collected probe requests from unknown devices and management packets of the Wi-Fi Access Points of the University. After this calibration, we sniffed a series of devices alone on three channels for 20 minutes for each mode. The length of the captures was selected after some experimental tests in which we observed some devices for an hour with the Wi-Fi interface switched off. In this case, the sending of probe requests is rarer, so a group of frames is transmitted at most every 20 minutes instead of every 3.

Modes are device settings that we classified as A, S, PA, PS, WA, and WS. These modes can be subdivided into active-screen modes (A, PA, and WA) and inactive-screen modes (S, PS, and WS). The device kept the screen switched on during the whole capture in the active-screen modes by playing a video. On the contrary, the device kept the screen on standby in inactive-screen modes. Furthermore, power-saving modes (PA and PS) refer to captures in which the device kept the power-saving setting active (all other captures have this setting disabled). Finally, captures in WA and WS modes were made with the device keeping the Wi-Fi interface switched off, whereas, in all other modes, devices kept the Wi-Fi interface active without connecting it to any access point. Table 4.2 summarises the settings of each mode with the three aspects considered. For example, the mode WS implies that the examined device has its screen on standby, its Wi-Fi interface is switched off, and the power saving setting is disabled.

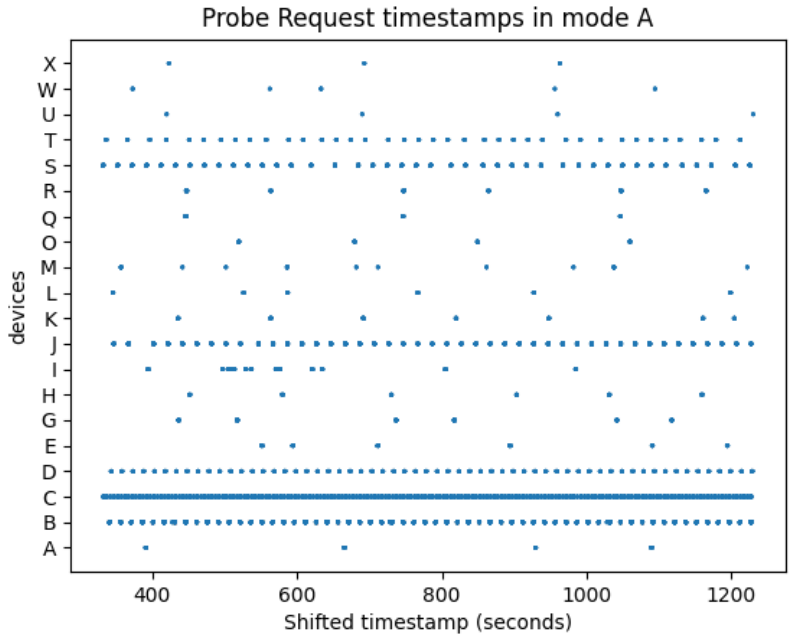


Figure 4.1: Sending time of the probe requests

After analysing the data collected in the anechoic chamber, we made additional captures in other environments. The analysis of this data allowed us to design a filtering algorithm based on power threshold for environments with a specific setting. Any undesired Wi-Fi interface within two meters from the sniffer must be removed, and the smartphone to analyse must be placed near the sniffer (within 20 centimetres) to use our filtering procedure. The radius of the free space around the sniffer was defined through experiments in which various captures had been made with sources (smartphones and other Wi-Fi devices) at different distances. At distances greater than 2 meters, no power peak of signal equal to or over -60 dBm was detected, which is distant enough from the -40 dBm threshold used for filtering. The name of each PCAP file in the dataset contains information about the ID of the device under consideration, the timestamp of the capture, the selected channel, and the device setting. For example, in the file "A-ts-2021-May-21-h11-m57-s24-modeS-ch-1-th-40.pcap", we consider the device with ID A, the capture occurred on 21st May 2021 at 11:57, considering channel 1, S mode (i.e., screen off, Wi-Fi on, and power-saving mode disabled), and a power threshold of- 40dBm.

Device ID	Device OS	Device OS version	Device vendor	Device model	Anechoic room	Random MAC
A	Android	11	Samsung	Galaxy M31	YES	YES
B	Android	6.00.01	Xiaomi	Redmi 4	YES	YES
C	Android	4.02.02	Samsung	Galaxy S4	YES	NO
D	Android	6.0	Huawei	ALE-L21	YES	YES
E	Android	10	Xiaomi	Mi A2 Lite	YES	YES
G	Android	10	Huawei	P20	YES	NO
H	Android	7.0	Samsung	Galaxy S6	NO	NO
I	Android	8.00.00	Samsung	Galaxy S7	NO	YES
J	Android	8.01.00	Xiaomi	Redmi 5 Plus	NO	YES
K	Android	10	Samsung	Galaxy J6	NO	YES
L	Android	11	Google	Pixel 3A	NO	YES
M	iOS	14.05.01	Apple	XS max	YES	YES
N	iOS	12.05.02	Apple	iPhone 6	YES	YES
O	Android	Oxygen 11	One Plus	Nord	NO	YES
Q	Android	9	Huawei	P10	NO	NO
R	Android	9	Huawei	Honor 9	NO	YES
S	Android	10	Xiaomi	Redmi Note 7	NO	YES
T	Android	11	Xiaomi	Redmi Note 9S	NO	YES
U	iOS	14.6	Apple	iPhone XR	NO	YES
V	Android	11	Google	Pixel 3A	NO	YES
W	iOS	14.05.01	Apple	iPhone 12	NO	YES
X	iOS	14.6	Apple	iPhone 7	NO	YES

Table 4.1: Smartphones used to produce the dataset

Mode	Active screen on	Wi-Fi on	Power saving on
A	X	X	
S		X	
PA	X	X	X
PS		X	X
WA	X		
WS			

Table 4.2: Device modes ("X" means that the relevant setting is enabled)

4.3 Filtering

Our filtering algorithm performed additional steps in case of capture in a non-isolated environment to simulate anechoic chamber capture conditions. The first step of the filtering is the removal of packets originating from a list of known interfaces (e.g., those of the Raspberry). Access-point MAC addresses are added to this list next. MAC addresses from APs are easily identifiable because these devices also send other management messages (probe response and beacon messages) with the same MAC they use for probe requests. Later, this list is used to discard all the packets that use one of these addresses as its source. The second filtering step uses a particular power threshold that takes advantage of the sending pattern of probe requests. Figure 4.1 is a plot showing the behaviour of devices with screen active and power-saving mode off. A similar sending pattern is observed when the power-saving mode is active and/or the screen is on standby. The letters in the vertical axis refer to the devices in the database. Moreover, we have verified that iOS devices transmit packets with almost the same power level in all channels in the experimental tests in the anechoic chamber. In contrast, Android devices have more variable power values. Android devices send a series of packets in short intervals (bursts) followed by pauses of a few minutes in which nothing is transmitted. During the burst, all packets maintain the same MAC address. Also, in the case of some Android devices, if the screen remains active, the MAC does not change even in different bursts. Our filtering algorithm, "SnifferFiltering", is composed of four parts: i) file-name grouping (merges all files of the same capture), ii) data conversion in Python structures, iii) power-threshold filtering, and iv) statistics and chart generation.

4.4 Simulation of Realistic Scenarios

Our dataset is freely available in the Mendeley repository (Pintor and Atzori, 2021) and allows for studying the individual behaviour of different smartphones with various settings based on display status, Wi-Fi connection, and power saving. Since the database captures contain only data emitted from a single smartphone, we combine multiple files to mimic realistic scenarios. Labels are preserved through external files that record the list of MAC addresses associated with each device. No scenario has different devices with the same MAC address. At the end of this phase, twelve merge files are generated to contain the synthesised scenarios (identified with numbers from 0 to 11): the merged files from 0 to 5 contain probes of devices set in the same modes, and other files contain probes of devices set in various modes. The composition of the scenarios is shown in Table 4.3. Each row refers to a device mode, and each column refers to a merged file (or scenario). Each file from the original dataset has a device ID and a mode. The devices of the first half of scenarios have the same setting, whereas they are shuffled in the second half. When the merged file

Scen.	0	1	2	3	4	5	6	7	8	9	10	11
mode A	all						A, G, M, S	B, H, T	C, I, O, U	D, J, V	E, K, Q, W	L, R, X
mode S		all					B, H, N, T	C, I, O, U	D, J, V	E, K, Q, W	L, R, X	A, G, M, S
mode PA			all				C, I, O, U	J, V	E, K, Q, W	L, R, X	A, G, M, S	B, H, N, T
mode PS				all			J, V	E, K, Q, W	L, R, X	A, G, M, S	B, N, T	C, I, O, U
mode WA					all		E	L, R	G, S	B, T	I, U	D, J
mode WS						all	L, R	A, G	B, H, T	I, U	J	E, K

Table 4.3: Composition of the synthesised scenarios

contains only data from devices in the same mode, "all" is reported. Otherwise, the IDs of the devices in the corresponding mode are listed. We are aware that combining artificial captions collected from individual devices may not perfectly replicate the behaviour of a group of devices located in the same area under the coverage of a sniffer. Yet, as far as we know, the literature does not propose other methods to label PRs according to the source device.

Chapter 5

Extraction and Analysis of Features for Device Fingerprinting

A feature refers to an individual, measurable property or characteristic of data. Features can be numerical, textual, categorical, or related to images. Feature engineering is the process of extracting and selecting them to improve the performance of a machine-learning model. This involves converting into numeric formats, scaling, normalising, and creating new features from existing ones. This way, features become arrays of numbers typically between 0 and 1. Normalisation is a good practice to ensure that high-value features do not overshadow others with lower values. A set of samples is usually organised in a matrix (an array of arrays) and associated with an array containing the labels corresponding to the classes if these are known. Supervised algorithms use the vector of labels to train themselves to recognise known classes, characterising them according to the feature values of their samples. These algorithms can then be tested with data other than the training set to verify their accuracy. On the other hand, no training is required for unsupervised machine-learning algorithms, so it is not necessary to know the class of the samples and, in some cases, not even the number of classes. The testing of these algorithms is more complex and requires calibration to predict reliable results. In summary, features are fundamental for machine learning, as they capture the characteristics of data that the model uses to learn patterns and make predictions. In this chapter, we describe how the conversion from packet to array of samples takes place, how features are selected and the calculation of their importance in the model prediction.

5.1 From Packets to Arrays

The scenario files are in PCAP format, as the captures in the original dataset. PCAP files contain network packet data and are typically used to analyse the network traffic for diagnostics. Features are extracted from the PCAP files with a Python algorithm that uses the scapy library and converted into Python data

frames. A Python data frame is a 2D data structure of the Pandas library that arranges data in rows and columns, providing a convenient way to store and manipulate structured data. These data frames contain the scenarios described in Section 4.4: columns represent features and rows represent samples. From each probe request, the mac address, the sequence number, the receiving power, the timestamp of acquisition, and a list of information elements are extracted.

Particularly, the IEs extracted and analysed are:

- IE 0 - The *SSID* (service set identifier) usually contains the network name of an AP memorised in the list of known APs of the mobile device. Nowadays, this field usually assumes an empty value with a length of 0 to prevent attacks that mimic known APs.

- IE 1 - The *supported rates* specify if the device supports the eight basic supported transmission rates, also called the basic service set (BSS).

- IE 3 - The *DSSS parameter set* contains the information about the direct sequence spread spectrum (DSSS). This field has a fixed length because it contains an integer number that encodes the identifier of the Wi-Fi channel used for the transmission.

- IE 45 - The *HT capabilities* is used for advertising the values of the following parameters used by the source device: optional high transmission (HT) capabilities, aggregate MAC service data unit (A-MSDU), supported modulation coding scheme (MCS), HT extended capabilities, transmit beamforming capabilities, and antenna selection (ASEL) capabilities. Figure 5.1 details the structure of this IE.

- IE 50 - The *extended supported rates* specifies if the device supports the extended service set (ESS) with supported transmission rates above those defined in IE 1.

- IE 107 - The *Interworking* contains information about the access network options and the SSID of hotspot networks. Most of the devices in the considered dataset do not support this functionality.

- IE 127 - The *extended capabilities* adds information about the capabilities of a mobile station that are not contained in IE 45. The length of this field is variable.

- IE 191 - The *VHT capabilities* states if a device supports very high throughput (VHT) capabilities, e.g., higher length of the MAC protocol data unit (MPDU) or the support for space-time block coding (STBC).

- IE 221 - The *vendor specific* is composed of a field for the organisation-unique identifier (OUI) and a variable field for additional information customised by the producer of the device, as shown in Figure 5.2. Multiple IEs 221 can be found in a single probe request.

- IE 255 - The *element ID extension* identifies a series of new IEs with the primary ID 255 and a secondary ID contained in the field element ID extension. The combination of these identifiers defines the content of this IE. Multiple IEs 255 can be found in a single probe request.

All these IEs are converted into numeric values as described in the Algorithm 1 and normalised with the scikit-learn (Pedregosa et al., 2011) MinMaxScaler so that these can be processed correctly in the following statistical analysis. The percentage of packets in the dataset affected by each IE is illustrated in Table 5.1.

Notably, it should be considered that a single PR frame can contain multiple 221 and 255 IEs with different values (Tan and Gary Chan, 2021). We managed this aspect by summing the values of all the replicated IEs. Moreover, IE 221 has been divided into the OUI and the fields features, and IE 45 into the bitmask and the flag features. This decision derives from the fact that the two fields that compose the IEs mentioned above sometimes have different scales, as shown in Figures 5.2 and 5.1, so the importance of fields with low values (221_fields and 45_flags) is nullified through the sum with the other fields of the corresponding IEs (221_oui and 45_bitmask). Another transformation considered in the performed experiments is the conversion in base-10 logarithms, which might be needed because the range of values is very wide and spread.

Moreover, other experiments are conducted using bursts as samples with additional features related to: i) the number of probes in a burst; ii) the difference in sequence number between the first and last PR; iii) and the difference in arrival time between the last and first probes of a burst. The new features are calculated after grouping PRs by MAC address and sorting them by time. After that, the inter-time between consecutive probes is calculated to aggregate PRs into bursts. Once PRs are grouped into bursts, a single array of values is extracted for each burst. At the burst level, the IEs are also considered features; in this case, the statistical mode of the values of the same IE of the PRs of that burst is computed. Calculating the statistical mode of the IEs should remove outlier values, which sometimes occur. Furthermore, these new features can improve the clustering.

5.2 Feature Selection

Feature selection involves choosing a subset of the most relevant features (or input variables) from the dataset while discarding less important or redundant ones. Re-

IE ID	Name	Type	Affected packets	%
0	SSID	string	69701	100.0
1	Supported Rates	array	69701	100.0
3	DSSS Parameter Set	number	68211	97.9
45	HT Capabilities	array	67487	96.8
50	Extended Supported Rates	array	69700	99.9
107	Interworking	array	2327	3.3
127	Extended Capabilities	array	58236	83.6
191	VHT Capabilities	array	7770	11.1
221	Vendor Specific	array	63465	91.1
255	Element ID Extension	array	10007	14.4

Table 5.1: IEs in the dataset with the percentage of frames where they are included

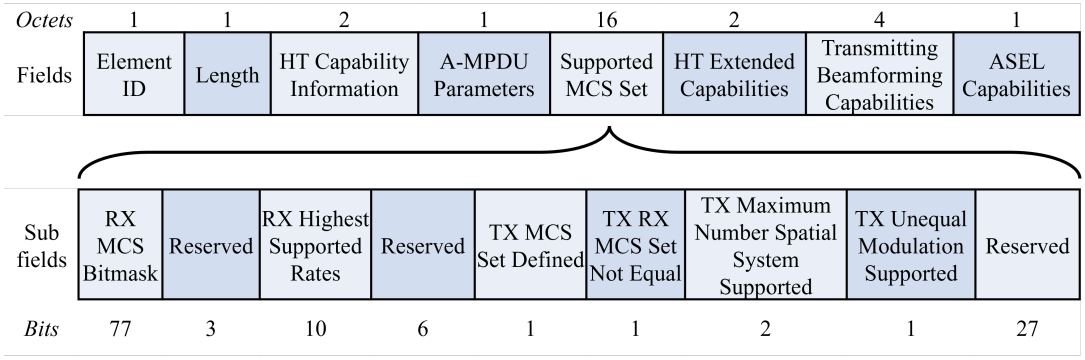


Figure 5.1: IE 45 comprises fields and subfields

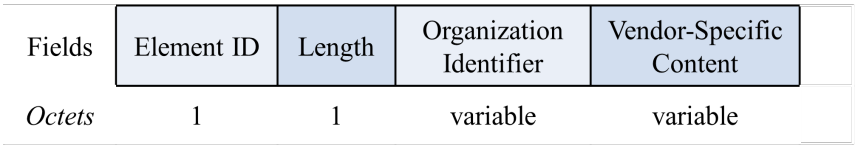


Figure 5.2: IE 221 fields: the organisation unique identifier (OUI) and the vendor-specific content

Algorithm 1 Conversion of the IEs content

```
if ie content is not present then
    value ← −1
else if ie is number then
    value ← number
else if ie is string then
    value ← sum of chars converted in decimals
else if ie is array then
    value ← sum of elements converted in decimals
end if
if ie has multiple instances then
    value ← sum of instance values
end if
```

moving irrelevant or noisy features enhances the predictive accuracy and reduces the risk that the model is perfectly adapted only to specific training sets (overfitting). Moreover, if fewer features are used, the dataset complexity decreases, and model training becomes easier. Among the methods for feature selection, there are some that i) include statistical tests (*filter methods*), ii) use the machine learning model itself to evaluate feature subsets (*wrapper methods*), and iii) select features as part of the model training process (*embedded methods*) (Abubakar et al., 2022). In order to study the impact of the information contained in the IE fields, we used wrapper methods based on the *random forest* (RF) classifiers from the scikit-learn framework, which contains ensemble algorithms. Ensemble methods are machine-learning algorithms that use multiple decision trees to improve classification performance. Each tree predicts a class for the input, and the most predicted class becomes the output of RF (Breiman, 2001) (Chen et al., 2020). The internal classification trees have different sub-sets and orders of features, leading to the reduction of over-fitting. We performed this analysis to identify and eliminate the features that might be irrelevant or even add noise to the classification. The following subsections describe how Random Forest can be exploited for the feature selection (Menze et al., 2009) (Behnamian et al., 2017) and the obtained results.

5.3 Feature Importance

Decision trees are suitable for finding non-linear prediction rules. Additionally, random-forest classifiers offer measures for feature ranking such as the *Gini importance* (Altmann et al., 2010) and the *Permutation importance* (Altmann et al., 2010). Gini importance is calculated by summing the impurity (or level of disorder) reductions achieved by changing how the tree uses the features to split its branches. In other words, it calculates how much a particular feature reduces the misclassi-

fication. Thus, features with higher Gini importance scores are more significant in decision-making within the model, while features with lower scores have less impact. Permutation importance is another metric to evaluate the significance of individual features in a predictive model. It measures how much a specific feature contributes to the performance by examining the impact of randomly permuting the values of that feature. This procedure breaks the relationship between the feature and the sample. Thus, the drop in the model score indicates how much the model depends on the feature. Usually, the value of the Permutation importance is an average of multiple algorithm iterations. However, this process can be computationally intensive, requiring repeatedly re-evaluating the performance for each permutation. Importance scores range from 0 to 1, where 0 implies no importance, and 1 indicates that the feature perfectly separates the data into distinct categories. Machine-learning libraries like scikit-learn provide tools to calculate Gini and Permutation importance for decision trees and ensemble models. It is essential to understand which features contribute most to performance and simplify it by focusing on the most important attributes.

5.4 Results

Before performing any statistical analysis on the features to evaluate their significance, their physical meaning has been considered to understand their role in the clustering procedures. The first information element analysed is IE 3, which defines the emitting channel and can assume multiple values, as shown in Figure 5.3, since the transmitter broadcasts messages in all the channels during the active scanning. No device has a single characteristic value. Most values are shared among multiple devices. These graphs refer to scenario 0, but similar results are obtained for the other scenarios. When the IE 3 is missing, a red cross has been drawn. The IE 3 cannot be considered a good feature because many devices have the same values, and also, a single device assumes as many values as the number of channels allowed in the carrier. However, this field is useful to discard corrupted packets because when its value is missing, other fields are affected and are different from other PRs emitted by the same device. These behaviours have been observed in the entire dataset. Removing these samples has been necessary to avoid affecting the classification procedure (Tan and Gary Chan, 2021). Another field that is discarded is IE 0 because it is usually empty in modern devices for security reasons. Moreover, when it is present, the transmitter fills it with one of the network names memorised into the preferred network list to accelerate the connection procedure. Accordingly, IEs 0 and 3 are discarded because unsupervised algorithms do not perform well when features vary within the same class (intra-class variance) or have similar values for different classes (inter-class similarity) (Venkataramanan et al., 2021). Intra-class variance might create more clusters for samples of the same class, while inter-class similarity might cluster together samples of different classes.

Features	Gini
ie127	0.334
ie45	0.223
ie221	0.220
ie255	0.077
ie191	0.060
ie1	0.047
ie107	0.025
ie50	0.015

(a) Gini importance

Features	Permutation
ie127	0.161
ie221	0.060
ie45	0.059
ie107	0.009
ie191	0.005
ie1	0.005
ie255	0.004
ie50	0.001

(b) Permutation importance

Table 5.2: Ranking of the features averaged in all the scenarios, considering each Probe Request as a sample

The analysis of features continued with the search for the optimal set of features with the support of decision trees (Wang et al., 2021) of the random-forest family. This supervised algorithm is trained with labelled data, and its structure is exploited to measure the Gini and Permutation importance. The RF algorithm of scikit-learn (Pedregosa et al., 2011) library has been used, setting the max depth of the tree equal to 4 and the random state parameter to 0. The max depth is set to limit memory consumption. Instead, the random state controls the randomness of the sampling and is set to allow repeatability. About the IE content, since the selected version of Random Forest accepts only numeric inputs, we converted and normalised them as described in Section 5.1. We did not separate training and testing sets because this analysis focused on the study of the feature importance rather than the classification of the RF algorithm. While calculating the Permutation importance, the number of its iterations is set to 30. Gini and Permutation importance ranked similarly when the samples are individual probe requests, as shown in Table 5.2. In this case, selecting the features was simple because the IEs 45, 127, and 221 cover 77.7% of the Gini importance and 92.1% of the Permutation importance. All features below 0.1 in Gini importance and below 0.01 in Permutation importance have been discarded. The results obtained using these and other derived features are shown in Chapter 6. The ranking differs when bursts are used as samples, as shown in Table 5.3. According to Gini importance, the best features are IEs 1, 45, 127, 255, 221, *delta_time*, and *probes*. These features account for 86.0% of the Gini importance and are all scored above 0.1. According to the Permutation importance, the best features are IEs 1, 45, 107, 127, 221, probes, and *delta_time*. Their importance constitutes 94.7% of the whole Permutation importance. These sets of features and some of their subsets are used in the experiments described in Subsection 6.2.2 The scikit-learn library (Pedregosa et al., 2011) is used to calculate both kinds of importance.

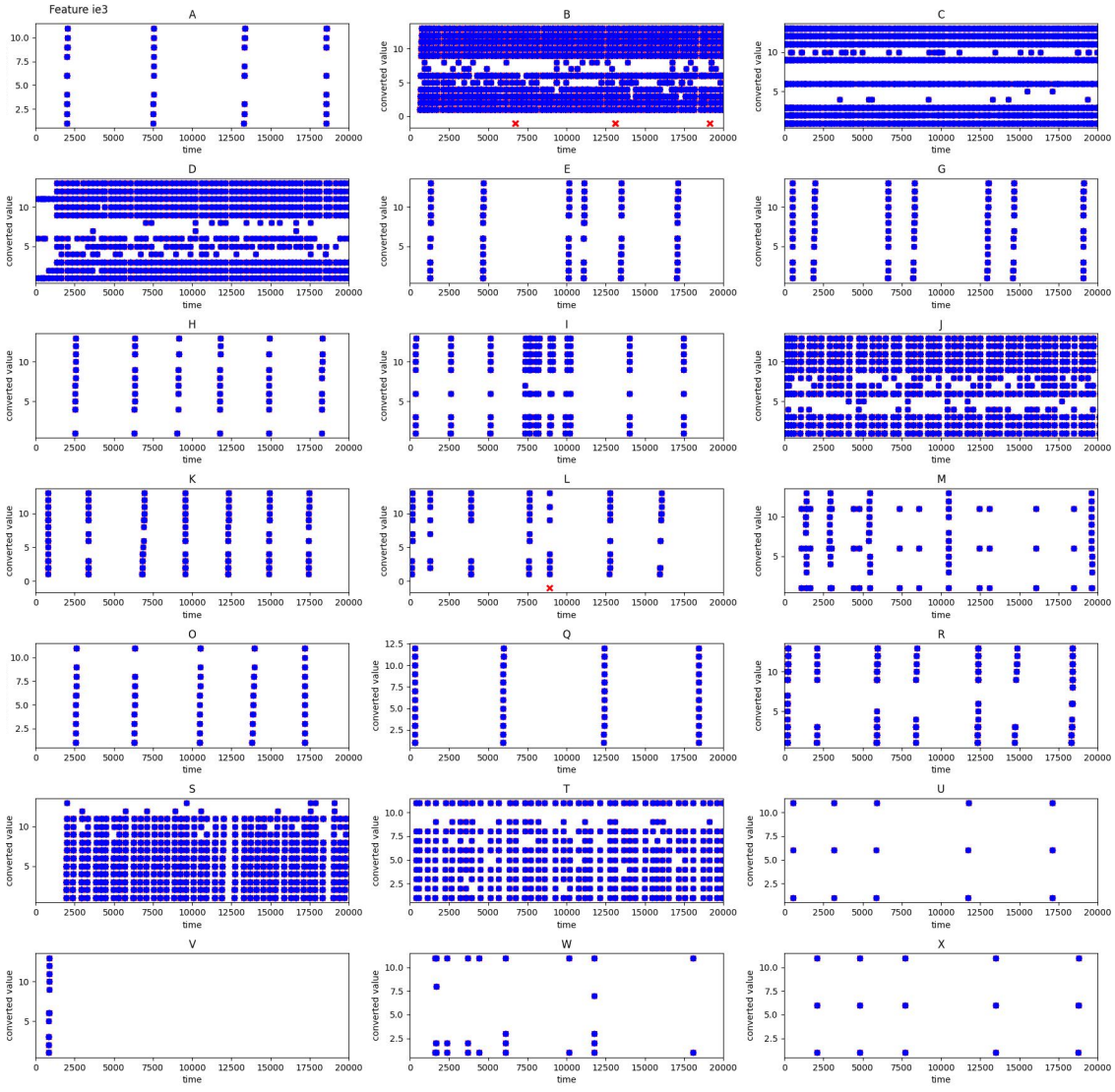


Figure 5.3: Subplot of the IE-3 values

Features	Gini
delta_time	0.151
probes	0.132
ie45	0.132
ie1	0.121
ie127	0.110
ie255	0.109
ie221	0.105
delta_seq	0.054
ie107	0.053
ie191	0.025
ie50	0.008

(a) Gini importance

Features	Permutation
ie45	0.035
delta_time	0.031
ie127	0.030
ie221	0.024
ie1	0.022
probes	0.022
ie107	0.017
ie255	0.016
delta_seq	0.008
ie191	0.002
ie50	0.001

(b) Permutation importance

Table 5.3: Ranking of the features averaged in all the scenarios, considering each burst as a sample

Chapter 6

Clustering Methods for Wi-Fi Fingerprinting Towards Information Element Analysis

We selected *unsupervised* clustering algorithms to group messages originating from the same device because, in a real environment, neither the number of Wi-Fi devices nor the sources of probe requests are known. Clustering can be defined as the process of discovering structure in data to group related objects together in clusters. A cluster is a group of objects that are more similar to each other than to objects in other groups. Each sample is defined by a set of features (or characteristics) that are used by a metric called a distance or similarity function to create clusters of similar items. The similarity of objects can be calculated as a geometric distance between arrays that measure the features of samples. Samples with similar values are more likely to be grouped in the same cluster, while samples far apart are more likely to be assigned to different clusters. In this scenario, clustering is applied to discover structure in the sequence of probe request frames and group them into clusters of frames predicted to be generated by the same source. Devices in an area transmit many management messages, which contain the requirements of the network they want to connect to. These requirements should not change often because they relate to the hardware and operative system installed in the mobile device. So, since the similarity of samples can be calculated as a geometric distance between arrays of features, messages from the same device should have similar features and be very close to each other.

Previous similar studies (Tan and Gary Chan, 2021) (Delzanno et al., 2023) (Covaci, 2022) used the DBSCAN (Ester et al., 1996) algorithm to cluster the input samples. DBSCAN groups data points based on their closeness to dense regions, not requiring prior information about setting the number of clusters. The inputs of the DBSCAN algorithm are the maximum distance between neighbour samples of the same cluster (*eps*), the minimum number of points to consider the

group a cluster (*min_samples*), and a set of features for each sample. DBSCAN starts by choosing an arbitrary point p, thus selecting all the points in the radius *eps*. If at least *min_samples* points, including p, are within distance *eps* of it, then p is considered a core point and assigned to a new cluster. If an additional core point is found in this cluster, the neighbourhood is expanded to include all its neighbouring points. The process is repeated until no more points can be assigned to the cluster. The algorithm terminates once all points have been processed. However, DBSCAN cannot detect clusters of varying density because it uses a constant distance value (*eps*) to determine whether a point is in a dense neighbourhood. DBSCAN algorithm assumes that the densities of different clusters are equal, even though many real-world datasets manifest different densities.

OPTICS algorithm (Ankerst et al., 1999) extends DBSCAN by generating a hierarchical clustering result for a variable neighbourhood radius. OPTICS generalises the DBSCAN approach by relaxing the *eps* parameter from a single value to a value range and automatically selects the one that fits better with the density variance. OPTICS builds a reachability graph with the vector distances of the input samples, differently from DBSCAN. The reachability graph shows the ordering of the points processed by OPTICS on the x-axis and the reachability distance on the y-axis. Points belonging to a cluster have a low reachability distance to their nearest neighbour, so clusters look like valleys in the reachability plot. The deeper the valley, the denser the cluster. In the rest of this thesis, when DBSCAN and OPTICS are mentioned, reference is made to the respective algorithms implemented in the scikit-learn library (Pedregosa et al., 2011). The default distance functions of the scikit-learn library were used for both algorithms: the Euclidean distance for DBSCAN and the Minkowski distance for OPTICS.

The rest of the chapter describes the metrics used to verify the accuracy of both clustering algorithms and the results obtained. The final section summarises the strengths and weaknesses of this approach.

6.1 Metrics to Verify the Accuracy

Since the dataset used for the experiments contains the true label of the samples, it is possible to use metrics that exploit the ground truth information to evaluate the clustering quality. The main objective of this work is to find a clustering methodology whereby the number of clusters is equal to or very close to the number of real classes. For this reason, the first metric considered is the difference between the number of classes (which correspond to the devices) and the number of clusters (or groups calculated by the algorithms), hereafter referred to as *error*. We called this quantity *delta* in Pintor and Atzori (2022b). This value is simple to calculate but does not consider how well the algorithm aggregates samples of the same class

in the same cluster and separates samples of different classes.

Moreover, this metric is useful when considering a single scenario, so we calculated its average to evaluate the performance in all scenarios. Therefore, we considered the following metrics:

- The *average error* (AE) is the average of the absolute of the delta of all the scenarios listed in Table 4.3. Since scenarios might have positive and negative deltas, taking the average of absolute values provides a measure of central tendency that is not affected by the sign of the numbers, focusing on the magnitude of variations rather than the direction.
- The *confidence interval* (CI) is an estimated range of values within which the value of a parameter is likely to fall. It provides a measure of the uncertainty or variability associated with estimating population characteristics based on a sample of data. A common way to express a confidence interval is by "*estimate* $\pm$ *margin of error*."
- The *v-measure* (VM) is a harmonic average of completeness and homogeneity (Rosenberg and Hirschberg, 2007). The difference between these two concepts is that while homogeneity measures how the clusters are composed of samples from a single class, completeness measures how samples of the same class are assigned to the same cluster.
- The *noisy points* (NPs) are the outlier values automatically discarded by the clustering algorithm. Yet, cutting the noise lowers the size of the dataset.
- The *silhouette coefficient* (SC) gives information about the similarity between the elements of the cluster and the separability between the clusters (Rousseeuw, 1987): this coefficient is optimal when it is close to 1, signals overlapping clusters when it is near 0, and indicates that samples of the different classes are assigned to the same cluster when it is negative.
- The *rand index* (RI) measures the similarity between the clusters and classes. It considers couples of samples and counts the ones that match the predicted and true labels (Valles Coral et al., 2022). The adjusted version of the RI extends the range of values from -1 to 1 to account for random cluster assignments and has an expectation equal to zero.

Table 6.1 summarizes these metrics and their acronyms that are used in the following tables. Moreover, calibrating the algorithm parameters implies that the difference between the number of devices and clusters is close to zero and that the v-measure value is sufficiently close to one. Incrementing *eps* usually reduces the number of clusters because the distance between them is reduced, and some neighbouring clusters can be merged. On the contrary, incrementing *min_samples* means that small clusters are discarded.

Acronym	Description	Range of values
ms	min_sample parameter	greater than 2
AE	average error	greater than 0
CI	confidence interval	greater than 0
VM	v-measure	between 0 and 1
NP	noise points	greater than 0
SC	silhouette coefficient	between -1 and 1
RI	random index	between 0 and 1

Table 6.1: Acronyms used in the tables of results

6.2 Results

Since some values of the IEs depend on the hardware and others on the user's settings, we aimed to demonstrate that a subset of the IEs might help discriminate messages from different devices, even though the source MAC address is randomised. For this reason, we selected features that are constant for the same device but vary considering different devices. Features like these are well-performing for unsupervised clustering algorithms that can be used in real-world scenarios. Based on the IE importance described earlier in Chapter 5, we now confirm the hypothesis that a set of IEs represents a fingerprint for probe requests, allowing for clustering them according to the same source. In this section, four types of experiments are presented: the first one shows the results obtained with the best-performing features using individual PRs as samples, the second uses the features extracted at the level of the burst, the third separates some of the IEs into their component fields, which are then considered as features, and the fourth performs the clustering by using the logarithm of the previously considered features. These are better detailed at the end of the following subsections with tables showing the following metrics: absolute error (AE), confidence interval (CI), v-measure (VM), noise points (NPs), silhouette coefficient (SC), and random index (RI). In the tables of the following subsections, the column "clusters" indicates how many clusters were counted on average, considering an average number of devices equal to 16.67. The *eps* value is not displayed in OPTICS tables because it varies to adapt to the density of samples.

6.2.1 Individual Probe Requests

The first analysis used the clustering algorithms with many sets of features by modifying the *min_samples* and *eps* parameters, as shown in Table 6.2. Each subtable is related to the two clustering algorithms considered, DBSCAN and OPTICS. The column "clusters" indicates how many clusters were counted on average, considering an average number of devices equal to 16.67. The column AE (absolute error) is the average absolute difference between the number of real devices and clusters the algorithm counts in all sets. The column CI (confidence

eps	ms	clusters	AE	CI	VM	NP	SC	RI	features
0.001	**10**	**16.08**	**1.25**	**0.58 +/- 0.89**	**0.96**	**3.17**	**0.87**	**0.98**	**45, 127, 221**
0.01	10	16.50	1.67	0.17 +/- 1.05	0.96	5.42	0.88	0.98	1, 45, 50, 107, 127, 191, 221, 255
0.001	**10**	**16.08**	**1.25**	**0.58 +/- 0.89**	**0.96**	**3.17**	**0.87**	**0.98**	**45, 50, 107, 127, 221**
0.01	10	16.50	1.67	0.17 +/- 1.05	0.96	5.42	0.87	0.98	45, 127, 221, 255
0.01	6	16.08	1.42	0.58 +/- 0.92	0.97	0.17	0.88	0.98	1, 45, 127, 221
0.01	6	15.58	1.25	1.08 +/- 0.85	0.96	0.17	0.87	0.98	45, 127, 191, 221
0.001	10	13.58	3.08	3.08 +/- 0.89	0.94	0.00	0.82	0.97	45, 127
0.001	2	15.67	1.33	1.00 +/- 0.93	0.93	0.17	0.57	0.94	45, 221
0.001	10	13.58	3.08	3.08 +/- 0.89	0.94	0.00	0.82	0.97	127, 221

(a) Individual probes with DBSCAN

ms	clusters	AE	CI	VM	NP	SC	RI	features
60	18.25	5.42	-1.58 +/- 3.60	0.84	252.50	0.87	0.90	45, 127, 221
80	18.33	6.17	-1.67 +/- 4.52	0.80	473.83	0.88	0.87	1, 45, 50, 107, 127, 191, 221, 255
60	18.25	5.42	-1.58 +/- 3.60	0.84	252.50	0.87	0.90	45, 50, 107, 127, 221
80	18.25	6.08	-1.58 +/- 4.45	0.80	467.92	0.87	0.87	45, 127, 221, 255
80	16.25	5.75	0.42 +/- 3.78	0.83	356.00	0.88	0.90	1, 45, 127, 221
60	18.25	5.42	-1.58 +/- 3.60	0.84	252.50	0.87	0.90	45, 127, 191, 221
10	**15.17**	**1.83**	**1.50 +/- 0.98**	**0.96**	**0.17**	**0.77**	**0.98**	**45, 127**
60	17.58	4.75	-0.92 +/- 3.12	0.81	176.08	0.57	0.89	45, 221
60	18.00	5.33	-1.33 +/- 3.52	0.83	231.50	0.82	0.89	127, 221

(b) Individual probes with OPTICS

Table 6.2: Best results using individual probe requests as samples

interval) shows the average difference between the number of real devices and clusters the algorithm counts and the margin of error (MOE). The column VM (v-measure) explains how well the clusters reflect the real association between probe requests and transmitting devices. Finally, the last column, features, indicates which information elements were considered.

Analysing the table, we notice that the best results are obtained using DBSCAN with the IEs 45, 127, and 221. The best result obtained with OPTICS is the one using only features 45 and 127. In this case, the silhouette coefficient (SC) is lower and, thus, greater similarity between samples of different clusters compared to other results with different sets of features. However, the number of discarded points (NP) is the lowest for OPTICS algorithms. In most cases, the silhouette value is close to 1, meaning that samples are well-matched to their clusters and poorly matched to others. The random index (RI) is higher for DBSCAN, indicating that it better reflects the real classes than OPTICS. Moreover, according to the normalised mutual information score, classes are sufficiently independent of one another. OPTICS automatically selected a value for *eps* smaller than DBSCAN because samples are probably very sparse. Moreover, we demonstrated that some features do not affect the clustering since, in both Tables, the best set of features (IEs 45,127 and 221) and the set of features containing IEs 50, 107, and 191 additionally gave the same results. IE 50 contains the extended supported rates, which are nowadays supported by all devices, so this field always has the same content. On the contrary, IEs 107 and 191 are rare, as illustrated in Table 5.1 and due to the feature-extraction algorithm, their value is usually -1, indicating a missing value. Using IEs 1 and 255 worsens the result of both clustering algorithms. Furthermore, the performance might degrade when one of the most important features (IEs 45, 127, and 221) is not considered. This has been proved for DBSCAN and partially for OPTICS.

In conclusion, each row of Table 6.2 contains average values of the 12 experimental scenarios. Table 6.3 includes the results obtained using DBSCAN with *eps* equal to 0.001 and *min_samples* to 10. A relationship between the number of samples and the v-measure score can be noted by analysing Table 6.3: usually, the more samples are available, the better the classification is. This experiment also demonstrated similar performance when devices have different settings, like a real environment. Unfortunately, many times, more devices are aggregated in the same cluster. The causes of this error might depend on the fact that we are analysing high-level information that can be modified via software and depends on the OS of the device. Devices with the same operative system might send probe requests with the same content in the IEs, making the fingerprints of different devices indistinguishable. Consequently, the algorithm counts fewer devices. This issue especially affects iOS devices constantly updated to the latest version (Martin et al., 2017), making them share similar IE signatures.

scenario	devices	clusters	delta	VM	samples	NP	SC	RI	features
0	21	18	3	0.98	24666	0	0.72	1.00	45, 127, 221
1	20	21	-1	0.97	8809	0	0.91	0.99	45, 127, 221
2	21	19	2	0.97	24647	0	0.71	1.00	45, 127, 221
3	18	18	0	0.97	8020	19	0.91	1.00	45, 127, 221
4	10	9	1	0.88	1480	0	0.84	0.88	45, 127, 221
5	10	9	1	0.91	589	0	0.76	0.90	45, 127, 221
6	16	16	0	0.97	8912	0	0.91	0.99	45, 127, 221
7	16	15	1	1.00	19860	0	0.99	1.00	45, 127, 221
8	18	15	3	0.98	6075	0	0.93	1.00	45, 127, 221
9	16	18	-2	0.96	9682	1	0.85	0.99	45, 127, 221
10	16	17	-1	0.97	5244	0	0.91	1.00	45, 127, 221
11	18	18	0	0.99	18438	18	0.98	1.00	45, 127, 221

Table 6.3: Individual probe clustering using eps equal to 0.001 and min_samples to 10 in DBSCAN

6.2.2 Bursts of Frames

A similar analysis was performed by grouping probe requests through bursts, which are regular intervals shorter than a second, in which the device keeps the same MAC address and sends many messages. The interval between a burst and the following one is often around a few minutes for new devices and some seconds for old devices. We compacted groups of probe requests into bursts to calculate other features such as the number of packets sent in a burst (which we call *probes* in Table 6.4), the difference between the first intra-burst sequence number and the last one (which we call *delta_seq*), and the difference between the first intra-burst arrival time and the last one (which we call *delta_time*). These features related to bursts are described in Section 5.1. Nevertheless, the performance of DBSCAN decreased compared to the results obtained considering probe requests individually. This result stems from the fact that selecting only one sample for each burst drastically reduces the amount of input data for the classifier (only 4789 bursts were counted out of 69700 probe requests). Surprisingly, the performance of OPTICS with the features IE 45, 127, and 221 is broadly higher, considering bursts as samples. The absolute error while considering probes individually with these features is 5.42, whereas it drops to 2.92 while aggregating frames into bursts. Also, the v-measure improves from 0.84 to 0.95 when switching from individual frames to bursts. Moreover, the features related to bursts (*delta_time*, *delta_seq*, and *probes*) seem to degrade the performance in all cases.

6.2.3 Split Features

Since we already demonstrated that, except for the three best-ranked IEs, the others did not add valuable improvements, the following experiments extract multiple features from these IEs. Breaking down a composite feature into its constituent

eps	ms	clusters	AE	CI	VM	NP	SC	RI	features
0.01	6	10.08	6.58	6.58 +/- 1.75	0.53	161.42	0.65	0.70	45, delta_time, probes
0.01	6	11.25	5.42	5.42 +/- 1.24	0.73	62.33	0.76	0.84	45, 127, delta_time
0.01	2	18.50	2.67	-1.83 +/- 1.50	0.86	11.67	0.50	0.92	45, 127, 221, delta_seq
0.01	6	11.42	5.25	5.25 +/- 1.37	0.73	63.67	0.77	0.84	45, 127, 221, delta_time
0.01	6	10.00	6.83	6.67 +/- 1.89	0.53	164.75	0.79	0.70	45, 127, 221, 1, 107, probes, delta_time
0.01	6	10.08	6.75	6.58 +/- 1.89	0.52	166.25	0.78	0.70	45, 127, 221, 1, 255, probes, delta_time
0.01	6	10.00	6.83	6.67 +/- 1.89	0.53	164.75	0.78	0.70	45, 127, 221, 1, probes, delta_time
0.001	6	12.75	4.58	3.92 +/- 2.20	0.48	168.75	0.75	0.67	45, 127, 221, probes
0.001	**2**	**14.50**	**2.17**	**2.17 +/- 1.13**	**0.96**	**1.92**	**0.84**	**0.98**	**45, 127, 221**

(a) Burst features with DBSCAN

ms	clusters	AE	CI	VM	NP	SC	RI	features
10	14.42	8.42	2.25 +/- 5.70	0.29	333.50	0.65	0.49	45, delta_time, probes
10	18.83	8.50	-2.17 +/- 6.38	0.38	280.92	0.76	0.57	45, 127, delta_time
6	17.75	5.75	-1.08 +/- 4.26	0.69	57.17	0.50	0.81	45, 127, 221, delta_seq
10	18.50	8.33	-1.83 +/- 6.36	0.37	285.08	0.77	0.56	45, 127, 221, delta_time
10	14.42	8.42	2.25 +/- 5.70	0.29	333.92	0.79	0.49	45, 127, 221, 1, 107, probes, delta_time
10	14.42	8.42	2.25 +/- 5.70	0.28	335.75	0.78	0.48	45, 127, 221, 1, 255, probes, delta_time
6	15.42	5.08	1.25 +/- 3.54	0.51	128.25	0.75	0.69	45, 127, 221, probes
2	**17.08**	**2.92**	**-0.42 +/- 1.89**	**0.95**	**2.08**	**0.84**	**0.98**	**45, 127, 221**

(b) Burst features with OPTICS

Table 6.4: Best results using bursts of frames as samples

parts might allow the extraction of meaningful information from individual fields that might be lost when considering the feature as a whole. Moreover, splitting can help capture non-linear relationships between features and the target variable or provide better insight into which components are more critical for making predictions. Specifically, as shown in Figure 5.2, IE 221 comprises two parts: the OUI of the vendor and the payload field. Since the vendor defines the payload format, the OUI part, which identifies the vendor, allows for the decoding of the IE. Regarding IE 45, it must be considered that it comprises multiple fields. Observing its structure, illustrated in Figure 5.1, it is possible to notice that the field bitmask allocates more bits than the other fields. The bitmask field allocates almost a third of the whole IE. This led to separating the bitmask from the remaining fields in these experiments. Moreover, while the bitmask field presents only three values (-1, 255, 510), the flags field is usually between 50 and 150 (see Figure 6.1). Device L seems to be an exception for the bitmask field because it presents variable values of the bitmask component. As shown in Table 6.5, DBSCAN does not improve its performance while splitting the most important features. We can notice the same results as the individual probe clustering (subsection 6.2.1) while using the features IE 45_flags, 127, and 221, meaning that the component IE 45_bitmask does not affect the clustering. All other cases show a degradation of performance. Regarding OPTICS, a little improvement is given while using the features IE 45, 127, and 221_oui. Component 221_oui is more influential with this clustering method because points of the same cluster have similar distributions, but points of different clusters do not. The different density of points is a characteristic considered in OPTICS but not in DBSCAN.

6.2.4 Logaritmic Conversion

Similarly to the previous experiment, only IEs 45, 127, and 221 are considered. These features occasionally exhibit significantly large values. For instance, as shown in Figure 6.2, for the feature IE 221, one device uses a value above $1e7$, another 255, and a third 160. The values of the last two devices are closely positioned relative to the first one. The logarithmic transformation has been employed to mitigate this disparity and achieve a more even distribution of values. Generally speaking, logarithmic transformations can help normalise a feature distribution and avoid features with significant variances dominating the learning process. Applying a logarithmic transformation might balance the impact of extreme values by making feature variance contribute more evenly to the model learning process. It is important to appropriately handle zero and negative values when calculating logarithms, as their logarithms are not defined. The considered data does not contain negative or decimal values before the process, so the only negative value to manage is -1, which remains -1 since there is no value equal to 0.1. Zero values are managed by converting them arbitrarily into -0.5, which is lower than the logarithm of the minimum positive value reported for these features. However, in this case, logarithm

eps	ms	clusters	AE	CI	VM	NP	SC	RI	features
0.001	10	15.58	1.25	1.08 +/- 0.85	0.96	0.17	0.85	0.98	45_bitmask, 127, 221
0.001	**10**	**16.08**	**1.25**	**0.58 +/- 0.89**	**0.96**	**3.17**	**0.87**	**0.98**	**45_flags, 127, 221**
0.01	20	15.92	1.58	0.75 +/- 1.11	0.92	72.33	0.83	0.95	45_flags, 45_bitmask, 127, 221_oui, 221_fields
0.01	**10**	**16.08**	**1.25**	**0.58 +/- 0.89**	**0.96**	**3.17**	**0.87**	**0.98**	**45_flags, 45_bitmask, 127, 221**
0.01	20	15.25	1.58	1.42 +/- 0.92	0.92	68.42	0.83	0.95	45, 127, 221_fields
0.01	20	15.50	1.50	1.17 +/- 0.99	0.92	68.42	0.83	0.95	45, 127, 221_oui, 221_fields
0.01	10	15.58	1.25	1.08 +/- 0.85	0.96	0.17	0.79	0.98	45, 127, 221_oui

(a) Split features with DBSCAN

ms	clusters	AE	CI	VM	NP	SC	RI	features
60	18.25	5.42	-1.58 +/- 3.60	0.84	240.75	0.85	0.90	45_bitmask, 127, 221
60	18.25	5.42	-1.58 +/- 3.60	0.84	252.50	0.87	0.90	45_flags, 127, 221
60	17.92	5.58	-1.25 +/- 3.73	0.82	307.50	0.83	0.90	45_flags, 45_bitmask, 127, 221_oui, 221_fields
60	18.25	5.42	-1.58 +/- 3.60	0.84	266.67	0.87	0.90	45_flags, 45_bitmask, 127, 221
60	17.92	5.58	-1.25 +/- 3.73	0.82	290.42	0.83	0.90	45, 127, 221_fields
60	17.92	5.58	-1.25 +/- 3.73	0.82	293.33	0.83	0.90	45, 127, 221_oui, 221_fields
10	**16.08**	**1.25**	**0.58 +/- 0.89**	**0.96**	**0.17**	**0.79**	**0.98**	**45, 127, 221_oui**

(b) Split features with OPTICS

Table 6.5: Best results using portions of IEs as features

eps	ms	clusters	AE	CI	VM	NP	SC	RI	clustering
0.001	10	16.08	1.25	0.58 +/- 0.89	0.96	3.17	0.87	0.98	DBSCAN
-	60	18.25	5.42	-1.58 +/- 3.60	0.84	252.67	0.87	0.90	OPTICS

Table 6.6: Best clustering results with logarithmic features.

conversion did not change the results, which are the same as the first experiment considering the same features, as shown in Table 6.6.

6.3 Strenghts and Flaws of this Approach

This approach makes it possible to collect information on crowding and flows of people without threatening privacy. Sniffers are usually inexpensive, but creating a dense network of sensors is necessary to monitor an area. The presented results achieve high accuracy, with absolute errors of 1.25 on an average number of devices of 16.6. The experiments showed that the best results are obtained with the DBSCAN algorithm when the clustering is performed at frame level when considering the three features that the Gini importance analysis has shown to be the most relevant (IE 45, IE 127, and IE 221). The optimal DBSCAN configuration parameters are $eps = 0.001$ and $min\_sample = 10$. In this case, the average error in counting the number of devices is 1.25 over an average of 16.67 devices, which brings an average error of around 7.5%.

Burst clustering is limited by reducing the number of samples because bursts contain 18 Probe Requests on average. These results might improve by having longer captures, which should have more samples. The experiments where the main features are split showed that the 45_bitmask does not influence the clustering. At the same time, IE 221_oui is more significant with OPTICS because samples of the same cluster have similar distributions and samples of different clusters do not. The different density of points is perceived by OPTICS but not by DBSCAN. Moreover, IE 221_oui drastically reduces the number of discarded samples. Finally, the expected benefits of using logarithms are not evident.

However, additional experiments can be performed to test the robustness and generality of the approach. For example, these methods might work differently with fewer devices or other datasets collected in other environments. The obtained results with an error of 7.5% are encouraging for many crowd-monitoring applications, such as counting the number of people in a bus, counting the number of attendants in a public event, and detecting whether the number of people in a room exceeds a given threshold. It is important to point out that the number of devices does not correspond to the number of people for several reasons, such as some individuals can bring more than one device, and some may have forgotten

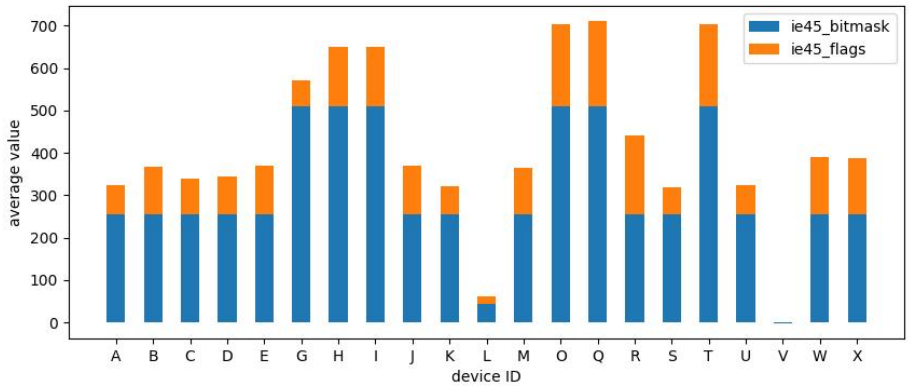
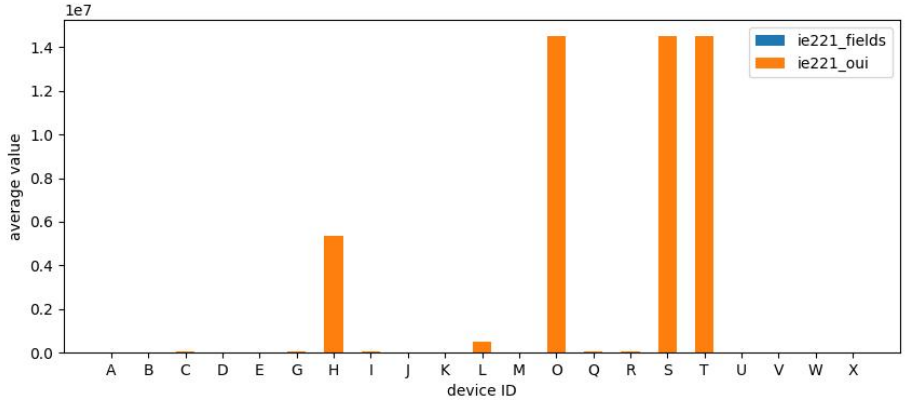
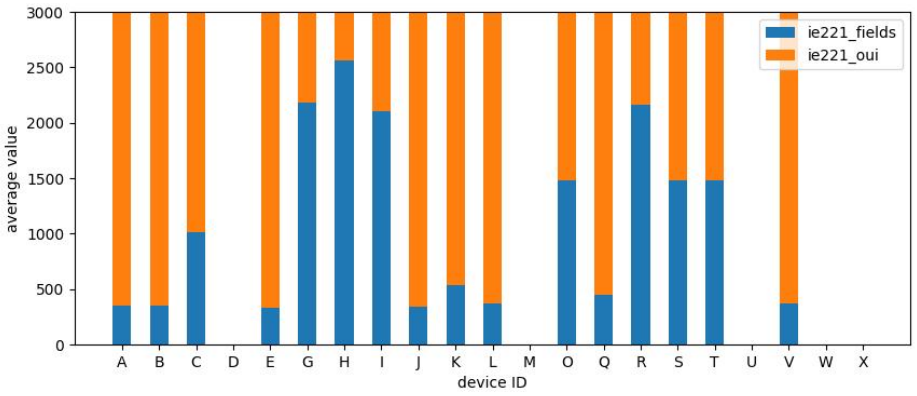


Figure 6.1: Average value of the IE 45 given by the sum of the components bitmask and flags

it. It is also important to highlight that the analysis of the features needs to be updated, as the operating systems of the major vendors may introduce different behaviours in the Wi-Fi management procedures, which can make the tracking more challenging.



(a) Values assumed by all devices



(b) Values assumed by all devices limited to 3000 to highlight that the OUI component is preponderant and almost nullifies the other one

Figure 6.2: Average value of the IE 221 given by the sum of its components

Chapter 7

Insight on the Physical-Level Fingerprinting based on the In-phase and Quadrature Imbalance

Even though the accuracy of the clustering presented in the previous chapter reached good performance, the easiness of replicating the same messages (and consequently features) with different devices motivated us to investigate other aspects. Other devices can easily assemble and re-transmit fields inside the probe requests. In Ketkhaw and Thipchaksurar (2017), a fake AP can replicate the behaviour of a real one, and mobile devices can also be mimicked via software. Furthermore, this chapter demonstrates that the probe requests of the dataset we considered previously can be replicated with a Raspberry Pi. Consequently, we investigated the analogical electromagnetic waves, specifically on the in-phase and quadrature (IQ) signals (Sankhe et al., 2020). Analogical transmitted signals depend on the hardware of the antennas (e.g., manufacturing errors and ageing), which makes them unique and difficult to replicate, even though the antenna respects the standards for its purpose. Our study is a proof of concept to explore a new application for IQ samples. We performed experiments in a controlled environment where a single device was transmitting each time. Making the same experiments in uncontrolled environments will be more complicated because we cannot control when multiple devices transmit simultaneously and how the noise affects the channel. The more devices are present in an area, the more collisions happen. As a consequence, the accuracy of the estimate diminishes while the monitored area becomes more crowded. However, if an approximation of the number of people is sufficient, this method allows for monitoring without privacy violations. The anomaly we consider in this study, the IQ imbalance, stems mostly from the non-orthogonality of the oscillator, which might alter the phase and amplitude of the signal. In other words, different devices will have different imperfections in the transmitted signals (Mohammadian and Tellam-

bura, 2021). In the rest of the chapter, we introduce some insights about the Wi-Fi standard regarding the physical layer, the information we tried to extract from the signal (IQ imbalance) and the method we used. The last two sections summarise the results obtained, limitations and contributions.

7.1 Wi-Fi Physical Layer

Probe requests use the same format as any other Wi-Fi message for the *physical-layer protocol data unit* (PPDU). Figure 7.2 shows that a PPDU contains a preamble, a header, a physical layer service data unit (PSDU), tail bits, and pad bits. The preamble and the header are patterns the receiver uses to interpret the received signal, learn the channel state information (CSI), and demodulate the data transmitted through *orthogonal frequency-division multiplexing* (OFDM). The physical-layer preamble (SYNC) is typically fixed: it is composed of 10 repetitions of the *short training sequence* (STS), a *guard interval* (GI), and two repetitions of the *long training sequence* (LTS). STS has a duration of $0.8\mu s$, LTS $3.2\mu s$, and GI $1.6\mu s$. Summarising, the total duration of the preamble is $16\mu s$. Chapter 17.3 of IEEE (2021) provides more details about the Wi-Fi physical level.

7.2 IQ Imbalance

Transceivers for general-purpose devices are often affected by radio-frequency impairments, which introduce interference and slightly degrade performance. The impairments considered in this chapter are due to the *in-phase and quadrature* (IQ) imbalance. Ideally, IQ modulators and demodulators should provide two orthogonal channels for the signal: one branch for the I component and one for the Q component. The imbalance stems from the non-orthogonality of these two components. IQ imbalance is often due to the manufacturing or ageing of the device, so it is difficult to evaluate and correct it. This mismatch might depend on the frequency (frequency-selective) or be constant over the signal bandwidth (frequency-flat) (Mohammadian and Tellambura, 2021). The exploitation of IQ signals is mainly focused on security applications related to *radio frequency fingerprinting* (RFF). According to this approach, Wi-Fi devices can be characterised by minute imperfections in their hardware and the consequent anomalies in the signals they modulate and transmit. These defects are unique and slowly vary with the usage of the device. Therefore, it is difficult to reproduce the same behaviour of a device, allowing for the extraction of features for identification. These imperfections produce several errors: quantisation errors, carrier frequency offsets, non-linear distortions, phase noise and IQ imbalance (Tang et al., 2021). The approach described by Suski II et al. (2008) calculates features extracted from the preamble waveform (phase and amplitude) of IQ signals generated by Wi-Fi devices. They reached an identification accuracy

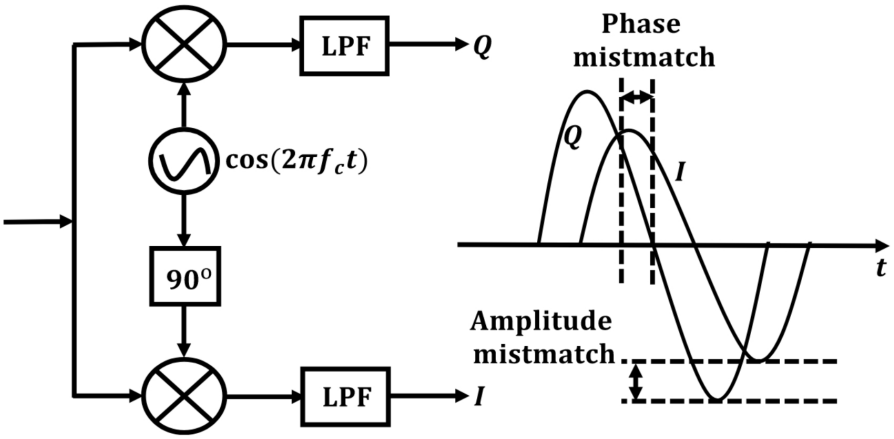


Figure 7.1: Demodulator with amplitude and phase mismatches in the IQ signal Adapted from (Mohammadian and Tellambura, 2021)

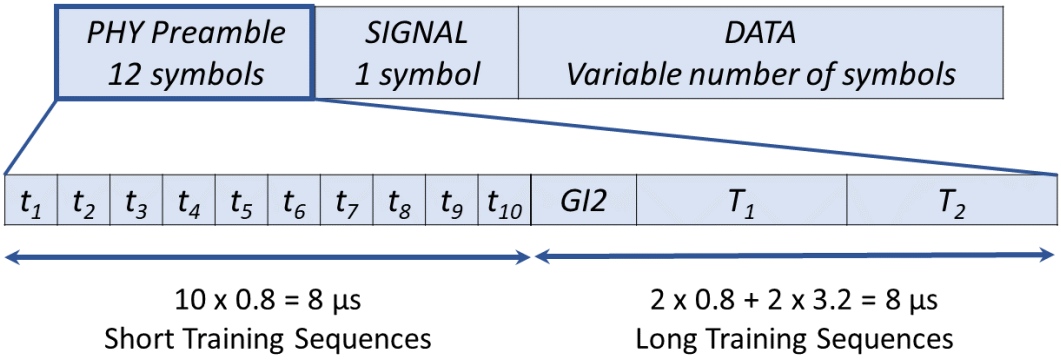


Figure 7.2: PPDU format and preamble structure.

of 80% with three devices. Other approaches exploit *convolutional neural networks* (CNNs), which are supervised ML methods with high accuracy for image processing. RFF features are converted into images (e.g., differential constellation trace figures) and used as input for these algorithms. These methods granted an accuracy of over 99% in controlled environments with ZigBee (Peng et al., 2020) and Wi-Fi (Sankhe et al., 2019) protocols. Zhou et al. (2022) proposed an *incremental learning* (IL) method to update the identification model, discard old data from the training set, and reduce the training time while reaching an accuracy of over 96%.

7.3 Radio Frequency Fingerprinting for Monitoring Crowds

We aim to demonstrate that it is possible to cluster IQ signals produced by the same source because each transmitter has its imbalance, which constitutes its fingerprint. To support this claim, we performed reproducible experiments in insulated environments with a transmitter that sends probe requests according to the experiment schedule. Moreover, controlling the transmitter allows us to explore diversity in different domains: i) *space domain* (the transmitter and the receiver are placed in different positions), ii) *time domain* (experiments are automated and conducted sequentially, so we can replicate some of them to study the temporal diversity too), and iii) *frequency domain* (transmissions are performed in all the Wi-Fi channels available in Europe). Studying the diversity of the same device is important to understand its behaviour in different conditions and properly address the clustering of signals from different sources. The main difficulty with this approach lies in applying a method based on a supervised system trained to recognise well-defined classes to a use case where neither the number of classes is known nor a training set is available. Another fundamental problem is the reception bandwidth of the *software define radio* (SDR), which, on the one hand, must comply with the Shannon-Hartley theorem and be at least twice the frequency of the signal, and on the other hand, is limited by the available memory and the sampling rate supported by the SDR. Since each Wi-Fi channel at 2.4GHz has a bandwidth of $22MHz$, we needed a high-performant SDR that supported up to $44MHz$ as sampling frequency. Yet, the captures could last only a few seconds because the high sampling rate made the capture files grow fast.

7.4 Experimental Setting

This section describes the hardware we used for the experiments, the architecture, and how we set and performed the experiments inside the anechoic chamber.

7.4.1 Hardware

The Hardware we used for the experiments is:

- *A software-defined radio* (model USRP X310) designed by Ettus Research and National Instruments collected the IQ samples of the messages transmitted during the experiments.

- *Tree Wi-Fi dongle interfaces*, respectively produced by Lifetron, Edimax, and Shenzhen, have been used for transmitting and collecting probe requests. Two different antennas were used in turn to transmit and receive probe requests.

The transmitting antenna was connected to the laptop controlling the experiment and the receiving to the Raspberry Pi.

- *A Raspberry Pi 3* (model B+) recorded the probe requests with Wireshark to verify that they were replicas of the dataset probe requests. One of the Wi-Fi dongles was connected to this device and used as a receiving antenna because the embedded one does not support the monitor mode.

- *A laptop* with Ubuntu OS was used to control all the experiments. It was connected to most devices through a switch and RJ45 cables. The adapter of the small form-factor pluggable (SFP) cable to manage the SDR was not supported, so an additional laptop with this interface was connected to the network to forward the GnuRadio commands to the SDR. The two computers are called PC1 and PC2 in the rest of the chapter.

7.4.2 Architecture

In order to identify the source of each message correctly, we insulated the transmitter (Wi-Fi dongle connected to PC1) and the receiving antennas (SDR and Wi-Fi dongle connected to the Raspberry Pi) by placing them inside an anechoic chamber. Figure 7.3 shows the wiring of the devices and Table 7.1 lists their roles and IP addresses. The legend shows that solid lines represent RJ45 Ethernet cables, dashed lines illustrate small form-factor pluggable (SFP) cables, and dotted lines represent USB extension cables. PC1 is directly connected to the Wi-Fi interface inside the anechoic chamber, which performs the transmission. Probe requests generated by the Operative System are blocked by setting the Wi-Fi interface in monitor mode. The probe request selected for the experiment is built and sent via Scapy framework for Python language. Additionally, this Wi-Fi interface has no IP address because it is not connected to any network. Furthermore, the switch has the role of connecting multiple devices to PC1 so they can receive *secure shell* (SSH) commands and perform their tasks as soon as the PC1 demands: the Raspberry Pi runs a script to start the Wi-Fi sniffing, and PC2 runs another script to start the SDR data collection. The data collected are Wireshark captures (collected and stored in the Raspberry Pi), IQ samples (collected by the SDR and stored in PC2), and reports about the experiments (stored in PC1). All files generated during a single experiment have similar names to simplify the analysis and comparison.

7.4.3 Collection of IQ data

We collected IQ signals through SDRs tuned on the carrier frequency of the Wi-Fi channel where our transmitter was set. Moreover, we automatised the experiments by creating files to set the Wi-Fi channel used, the gain of the SDR receiver, the waiting time between two consecutive probe requests, the number and the content of

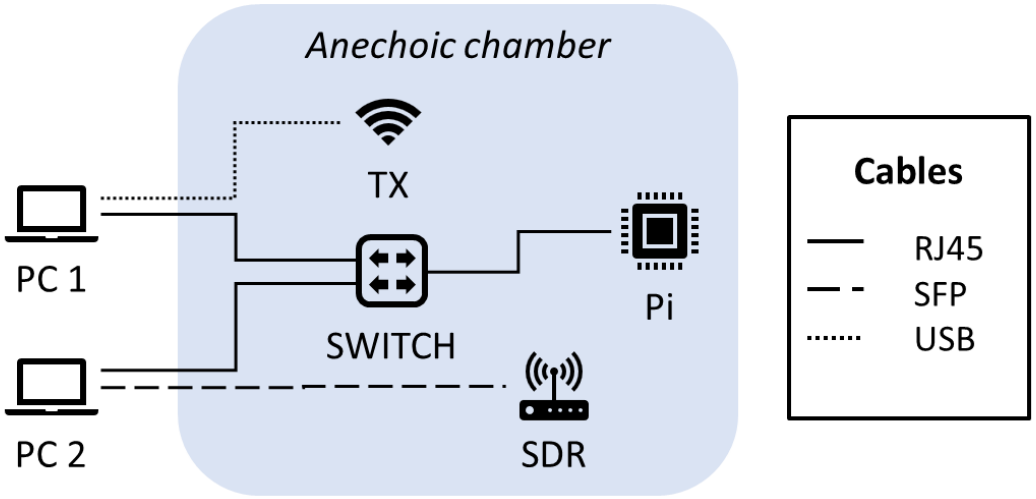


Figure 7.3: Architecture of the experimental setting

Device	Role	IP address
PC1	Controls the experiments	192.168.10.3
PC2	Gets commands from PC1 Forwards commands to SDR	192.168.10.14 192.168.40.1
Pi	Collects probe requests	192.168.10.15
SDR	Collects IQ signals	192.168.40.2

Table 7.1: Roles of the devices involved in the experiments

the messages to transmit, and the sampling rate. PC1 runs the main Python script with the input parameters for the experiments. This script starts three parallel subprocesses through the fork command:

- The first subprocess builds a probe request and sends it through the connected Wi-Fi interface inside the anechoic chamber multiple times. We set 200 repetitions in most of the experiments because it was challenging to synchronise all the devices.
- The second subprocess connects PC1 to the Raspberry Pi via SSH and starts the collection of probe requests (sniffing) on that device with Wireshark. Having the Wireshark captures lets us verify if the probe requests were effectively sent and other unwanted Wi-Fi messages were transmitted.
- The third subprocess allows PC1 to start the script to control the SDR from PC2 via SSH. The SDR then collects IQ samples and transmits them through the SPF cable to store them in PC2.

The transmitting window is larger than the receiving one. In other words, the transmitting antenna sends probe requests before the two receivers (Raspberry and

SDR) are active and ends after they finish. In this way, we granted the collection of at least a complete probe request. This artifice was necessary to simplify the architecture because i) the SDR starts to collect data after some checks, which take seconds, and ii) the SDR collects thousands of complex samples every second, leading to buffer filling and storage consumption. To use SDRs to count people outside the controlled environment, it is necessary to apply filters directly to the SDR to discard the samples collected when no signal is transmitted.

7.5 Equivariant Adaptive Separation via Independence

The RFF methods described in the literature are designed to identify known devices securely with supervised algorithms. However, since the features of these algorithms grant the identification of the source, we want to demonstrate that they can also be valuable for developing new techniques for device counting and tracking. Our study collected some features by exploiting the *equivariant adaptive separation via independence* (EASI). This approach uses a step-sized algorithm to optimise the orthogonality between sources. Thus, applying an EASI algorithm to IQ signals should overcome the non-orthogonality of these two axes and reconstruct the ideal signal. Briefly, the adaptive estimation of the source signal is given by:

$$\mathbf{y}(n) = \mathbf{W}(n)\mathbf{x}(n) \tag{7.1}$$

Where $\mathbf{x}$ is a 1D array of length 2 containing the IQ samples and $\mathbf{W}$ is a 2x2 separating matrix initialised as an identity matrix. The discrete-time notation n is used in all the equations. Moreover, bold lowercase letters indicate arrays, whereas bold capital letters indicate matrixes.

$$\mathbf{W}(n+1) = (\mathbf{I} - \mu(n)\mathbf{H}(\mathbf{y}(n)))\mathbf{W}(n) \tag{7.2}$$

The separating matrix $\mathbf{W}$ is updated in every iteration until it reaches convergence. In equation (7.2), μ is the step size, $\mathbf{I}$ is a 2x2 identity matrix, and $\mathbf{H}$ is a simplified matrix-valued adaptation function. The full definition can be found in Valkama et al. (2001) and Hesse et al. (2008).

$$\mathbf{H}(\mathbf{y}) = \mathbf{y}\mathbf{y}^H - \mathbf{I} \tag{7.3}$$

The separation expressed in the equations above is performed by eliminating the nonlinear cross-correlation between the I and Q signals. Indeed, the estimated signal $\mathbf{y}$ evaluates the ideal signal before the transmission. On the other hand, the $\mathbf{W}$ matrix is related to the IQ imbalance and contains information about it. In this perspective, our first set of features comprises the elements in the diagonal of the $\mathbf{W}$ matrix after it reaches convergence.

7.6 Results

After collecting data from some experiments, we tried to calculate the module of the complex numbers collected as IQ samples. Our approach firstly uses a threshold slightly higher than the noise (0.0025) to group samples of the same probe request and discard the ones collected when no signal was transmitted for over 1000 consecutive samples. The threshold and the number of guard samples between probes depend on the context and must be experimentally calculated. Secondly, a new threshold equal to a percentage (10%) of the max value of the module of the IQ samples of each probe request is used to find the first sample over it. Figures 7.4 and 7.5 show the module of the IQ samples collected in one of the experiments in the anechoic chamber. The red line is a threshold to discard the noise. Given the structure of the Wi-Fi preamble, described in section 7.1, once the first sample is identified, it is simple to extract the preamble because its duration is fixed ($16\mu s$). We then identified the ten STS symbols and the two LTS symbols, as shown in Figures 7.6 and 7.7. The dots in the graph correspond to the samples and are interpolated with a blue line. The localisation of the preamble starting time is called transient detection and is essential to separate signal samples from the background noise (Suski II et al., 2008). The transient detection can be evaluated using features based on the phase and the amplitude of the collected signal.

After identifying the preamble, we applied the EASI method, described in Section 7.5, to orthogonalise the two IQ components. Our hypothesis was to use the values of the compensation matrix obtained after the algorithm convergence as features to discriminate antennas. The process began by calculating the first iteration with equation 7.1 by initialising the matrix $\mathbf{W}$ as a 2x2 identity matrix and using the first sample of the preamble. The value obtained is the first approximation of the compensation array $\mathbf{y}$ that can be multiplied by the original IQ samples to obtain a corrected sample. The process continues by updating the matrix $\mathbf{W}$ with equations 7.2 e 7.3 and repeating the procedure until convergence of $\mathbf{W}$ is achieved. However, in most experiments, the values of this matrix did not stabilise, as shown in Figures 7.8. The same antenna (Edimax) displayed various behaviours while compensating for the IQ imbalance for the same probe request. The results obtained with the antenna Lifetron were similar. Furthermore, we reported different behaviour for probe requests emitted by the same antenna and similar behaviour for different antennas.

7.7 Limitation and Contributions of this Work

Although this trial did not report any publication-worthy results, it was useful in identifying probable causes and other issues related to using IQ signals for tracking devices. Since we found different behaviours for the same interface sending the

same probe request, the problem could lie in the code or the data collected. The algorithm for calculating the compensation matrix, initially written in Python, was rewritten in Julia language to compensate for any issues in the code and libraries. Yet, changing the programming language did not change the results. Another possibility inherent in the computerised calculation might be due to null samples, so an offset was added to all the samples to avoid zero crossing. The offset stabilised the values of $\mathbf{W}$, meaning they are no longer infinite. However, they still do not converge to a constant value.

Another assumption is that the transmitting-antenna hardware used is of low quality and produces random noise that varies the behaviour of the interface, making it impossible to be constant. Moreover, another reason for this perturbation is surely associated with the variation in the reception power of the radio. Since the received signals do not all have the same power, it might disturb the algorithm. Moreover, the algorithm that we used is blind, and if it converges does not guarantee the uniqueness of the solution. Other approaches consider other features based on graphical representations of the IQ signals (Jian et al., 2020). Probably, it would have been necessary to build a pre-processing stage which guarantees a correction of the power bias. Since we are dealing with well-defined signals (preambles), it would surely have been more appropriate to build a method that knows these pilots to deduce an estimate. These issues might also be due to the bandwidth of the SDR because the device might reduce the sampling frequency to avoid crashing. A last but not least problem is the high memory requirement to store the IQ samples. Essentially, a solution can be the application of filters directly in the SDR to discard under-threshold values at the source.

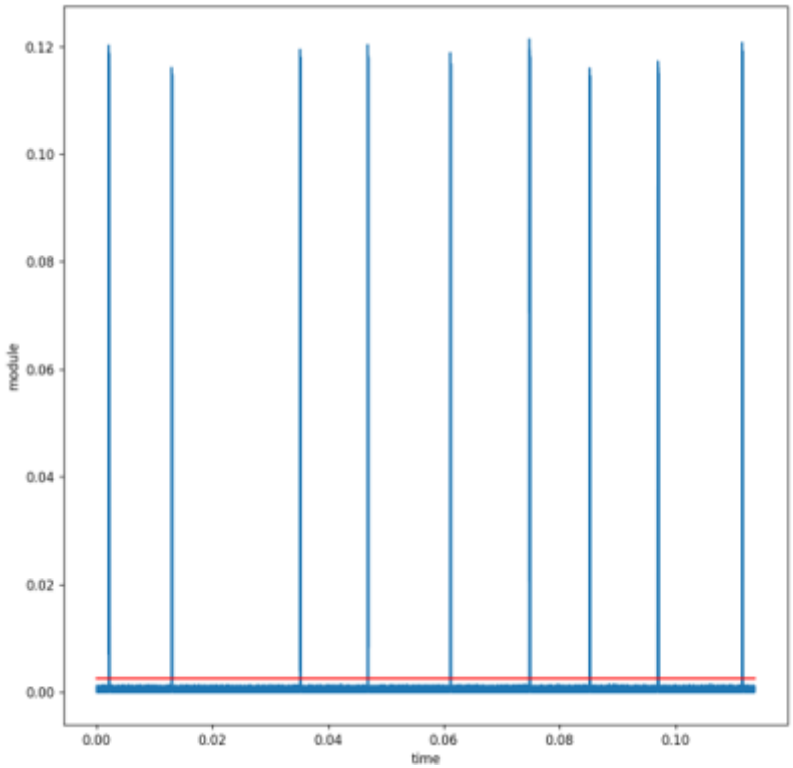


Figure 7.4: Module of the IQ samples collected in one of the experiments

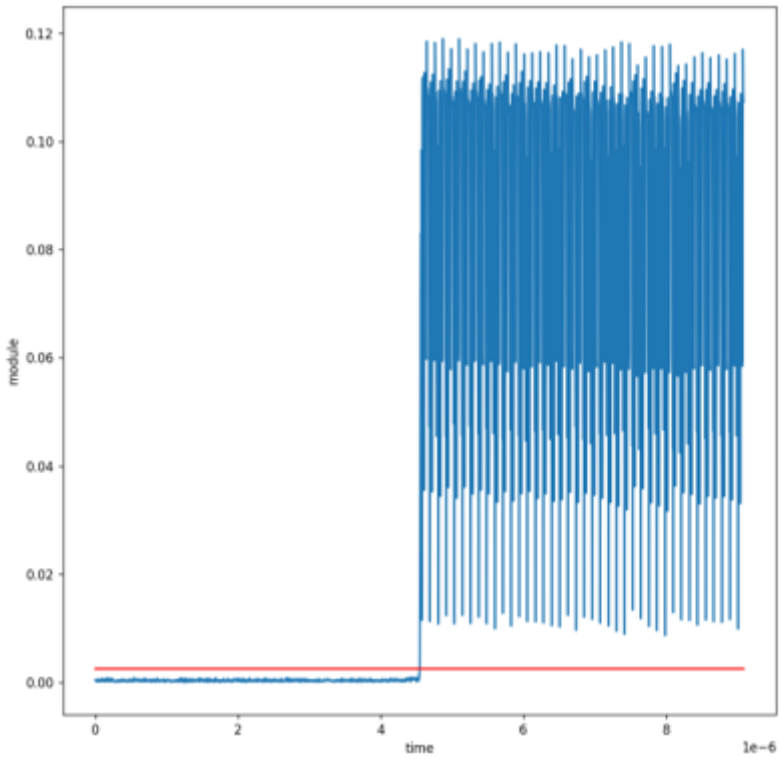


Figure 7.5: Zoom of the module of the IQ samples in a single probe request

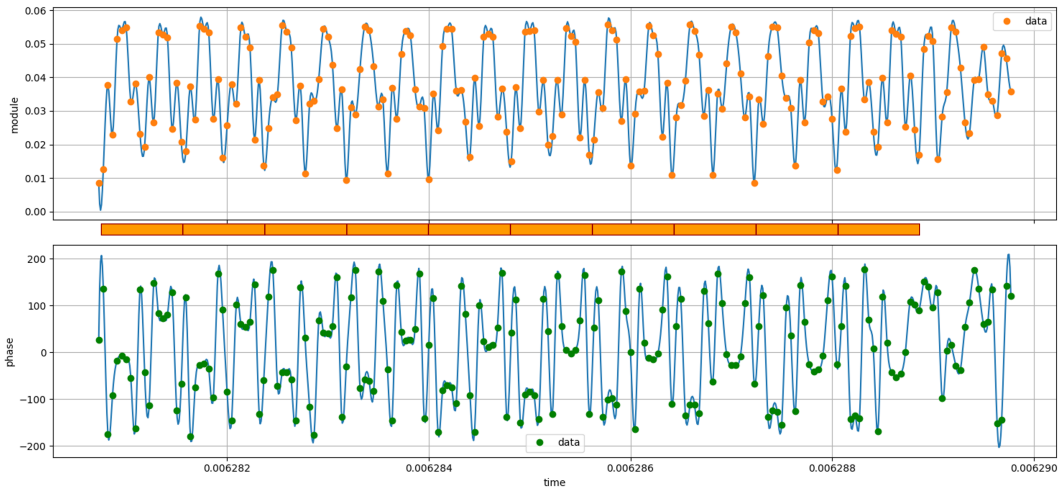


Figure 7.6: Module and phase of the ten symbols of the STS (0.8μs each)

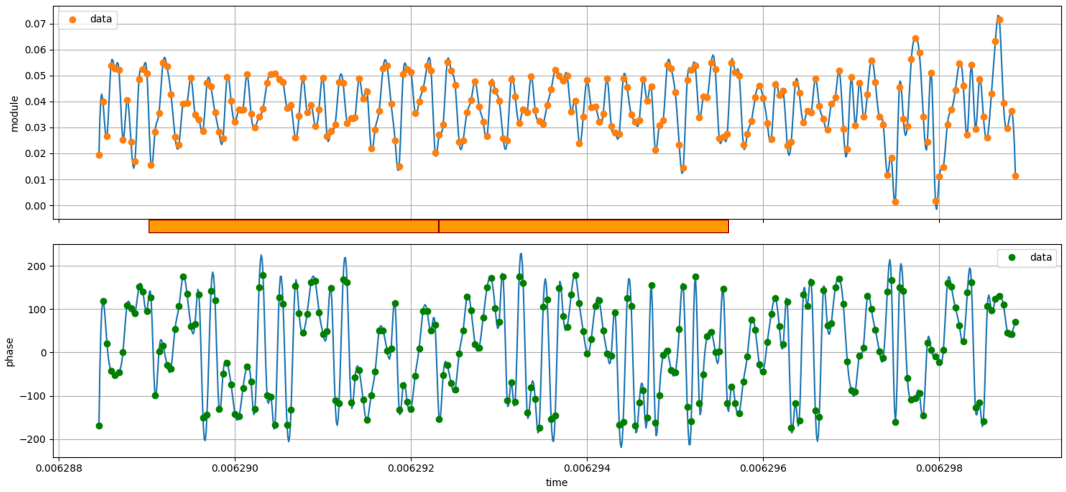
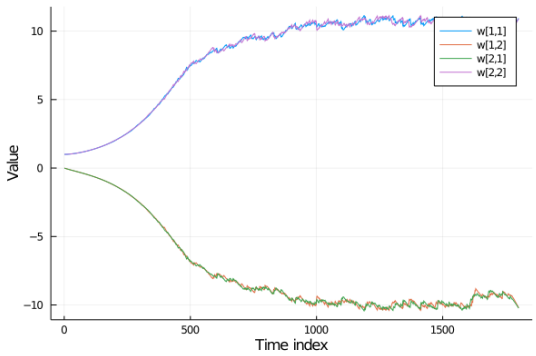
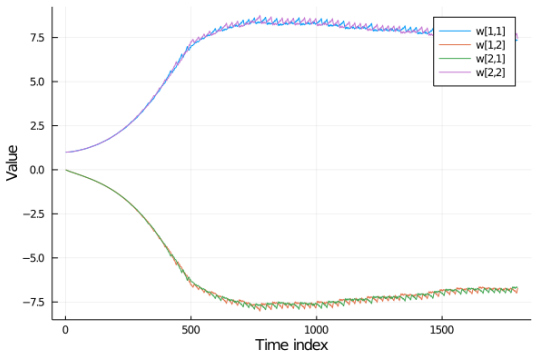


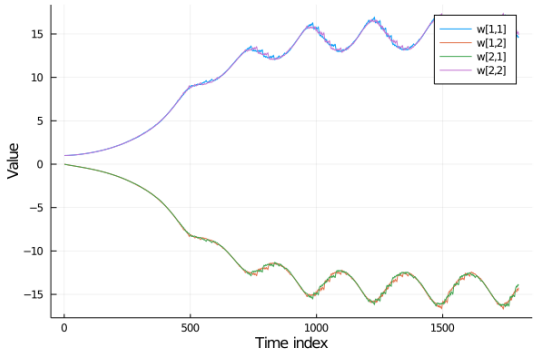
Figure 7.7: Module and phase of the two symbols of the LTS (3.2μs each)



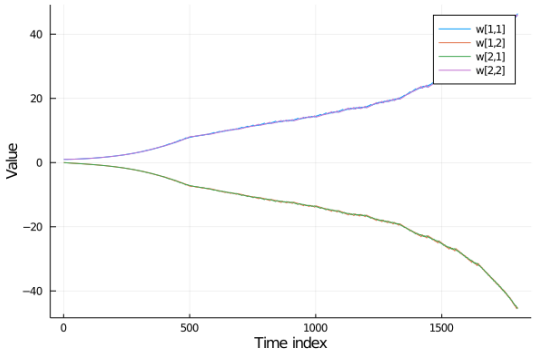
(a) W is converging with a low noise level



(b) W components are going to zero



(c) W is converging with a high noise level



(d) W components are not converging

Figure 7.8: Subgraphs showing the four components of the W matrix T

Chapter 8

Discussion and Conclusions

Monitoring the flow of people in a smart city can provide valuable data for urban planning, public safety, transportation optimisation, and various other purposes. The main strengths of the methods to count the number of people and track crowds based on the analysis of Wi-Fi management messages are the cost-effectiveness of sensors and the protection of privacy. Knowing how people move within a city allows the optimisation of resources like public transportation to meet actual demand. Moreover, real-time data on traffic flow can be used to manage and alleviate congestion, reducing commuting times and emissions. However, data collected with the proposed approach may not represent the entire population, leading to biased decisions if not properly accounted for. For example, people who do not have a smartphone would be invisible to the system, or people with several Wi-Fi devices would be considered as a group of people. Another difficulty is related to the testing of these systems, as it is difficult to verify the source of each message without using artifices such as merging packet captures, which were known to originate from the same source as we have proposed in Chapter 4. Building a dataset of individual tracks takes time and should be updated at least as frequently as the average smartphone lifespan, i.e., every few years. In particular, since the Wi-Fi standard is continuously evolving, modern devices support additional IEs compared to older ones to identify new features. This chapter summarises the contributions, limitations, future directions, and concluding remarks.

8.1 Summary of Contributions

This work focuses on enhancing the accuracy of clustering systems to discriminate probe requests originating from the same source device to count the number of mobile devices (and then estimate the number of people) under the coverage of a sniffing device. Our contributions can be summarised as follows:

- We created and published a dataset with real ground truth to test methods to count and track people based on analysing Wi-Fi probe requests;

- We presented new methods to extract, represent, and analyse the major available features of the PRs that can be used to fingerprint the Wi-Fi devices when both single frames and bursts of frames are considered;
- We performed an extensive investigation about the importance of the features to identify those that are the more significant when applying a clustering algorithm to aggregate the frames generated by the same device;
- We compared the performance of two clustering algorithms by considering the following indexes: the content of the IEs, the average delta time between a probe and the following one of the same burst, the difference between the first and last sequence number of the burst, and the number of probes per burst.

8.2 Limitations and Future Directions

Despite the good results achieved, the analysis shows that some IEs always assume the same values and are no longer discriminating. Some examples are IEs 1 and 50, which contain the rates available according to the standard and are now supported by almost all devices. Over time, other network requirements may also be supported by all devices and thus become non-discriminatory features. For this reason, it is essential to have up-to-date databases to adapt these methods to the future evolution of the standard. Furthermore, some of the newer devices considered in the dataset use IEs not present in the others. In particular, IE 255 encodes additional secondary elements not yet defined in the current standard. Other algorithms for grouping frames according to the source can also be experimented with in future works. A possible approach is to use *dynamic programming* that could rely on a cost function that evaluates the goodness of each different combination of sequences of frames generated by different sources in the captured trace. Complexity may be an issue in this case, so the solution may not scale very well with the number of captured frames. *Genetic algorithms* can be applied to reduce complexity to find sub-optimal solutions for the sake of computational time.

8.3 Concluding Remarks

This thesis provides a methodology for modelling a dataset with labelled management messages and procedures for identifying new feature sets to obtain more accurate results. Yet, new directions can be explored, especially for developing new techniques to facilitate testing with real ground truth, because creating a dataset still requires time and must be frequently updated. We believe that this work provides useful tools and guidelines with the hope that this will improve, in the future, the state of research in this field and the accuracy of methods for people counting and crowd monitoring based on the analysis of Wi-Fi management messages.

Acknowledgements

This work has been partially funded by

- the Italian Ministry for the Economic Development (MISE), under the framework "Asse II del programma di supporto tecnologie emergenti (FSC 2014-2020)", project Monifive and
- the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, "Sustainable Mobility Center" (Centro Nazionale per la Mobilità Sostenibile), CNMS, CN 00000023.

Appendix A

Tables of abbreviations

Abbreviation	Meaning
AP	access point
BLE	bluetooth low energy
CDR	call data records
CNN	convolutional neural network
CSI	channel state information
EASI	equivariant adaptive separation via independence
FM	frequency modulation
GDPR	general data protection regulation
ID	identifier
IE	information element
IEEE	institute of electrical and electronics engineers
IL	incremental learning
IQ	in-phase and quadrature
lidar	light detection and ranging
MAC	media access control
ML	machine learning
MNOs	mobile network operators
NIC	network interface controller
NN	neural network
OD	origin-destination
OFDM	orthogonal frequency-division multiplexing
OS	operative system
OUI	organisational unique identifier
PC	personal computer
PCAP	packet capture
PNL	preferred network list
POI	points of interest
PR	probe request
radar	radio detection and ranging
RFF	radio frequency fingerprinting
SDR	software define radio
SFP	small form-factor pluggable
SSH	secure shell
WLAN	wireless local area network

Table A.1: General abbreviations used in this thesis

Abbreviation	Meaning
AE	absolute error
AMI	adjusted mutual information
CI	confidence interval
MI	mutual information
MOE	margin of error
NMI	normalised mutual information
NP	noisy points
RI	rand index
SC	silhouette coefficient
VM	v-measure

Table A.2: Abbreviations of the metrics used in this thesis

Abbreviation	Meaning
DBSCAN	density-based spatial clustering of applications with noise
OPTICS	ordering points to identify the clustering structure
RF	random forest

Table A.3: Machine Learning algorithms used in this thesis

A-MSDU	aggregate MAC service data unit
ASEL	antenna selection
BSS	basic service set
DSSS	direct sequence spread spectrum
ESS	extended service set
GI	guard interval
HT	high transmission
LTS	long training sequence
MCS	modulation coding scheme
MPDU	MAC protocol data unit
PPDU	physical-layer protocol data unit
PSDU	physical layer service data unit
RSSI	received signal strength indicator
SSID	service set identifier
STBC	space-time block coding
STS	short training sequence
VHT	very high throughput
WPS	Wi-Fi protected setup

Table A.4: Abbreviations of Wi-Fi components

Bibliography

S. M. Abubakar, Z. Sufyanu, A. Miyim, V. Arputharaj, and S. Kumar. Comparisons of filter, wrapper and embedded-based feature selection techniques for consistency of software metrics analysis. *SLU Journal of Science and Technology*, Vol4:188–204, 07 2022. doi: 10.56471/slujst.v4i.238.

A. Altmann, L. Toloşi, O. Sander, and T. Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 04 2010. ISSN 1367-4803. doi: 10.1093/bioinformatics/btq134.

M. Ankerst, M. Breunig, P. Kröger, and J. Sander. Optics: Ordering points to identify the clustering structure, 06 1999.

S. A. Arif, M. H. Niaz, N. Shabbir, M. H. Zafar, S. R. Hassan, and A. ur Rehman. Rssi based trilatertion for outdoor localization in zigbee based wireless sensor networks (wsns). In *2018 10th International Conference on Computational Intelligence and Communication Networks (CICN)*, pages 1–5, 2018. doi: 10.1109/CICN.2018.8864943.

A. Behnamian, K. Millard, S. N. Banks, L. White, M. Richardson, and J. Pasher. A systematic approach for variable selection with random forests: Achieving stable variable importance values. *IEEE Geoscience and Remote Sensing Letters*, 14 (11):1988–1992, 2017. doi: 10.1109/LGRS.2017.2745049.

M. Berlingerio, F. Calabrese, G. Lorenzo, R. Nair, F. Pinelli, and M. Sbodio. Allaboard: A system for exploring urban mobility and optimizing public transport using cellphone data. In *Machine Learning and Knowledge Discovery in Databases*, pages 663–666, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-38708-1. doi: 10.1007/978-3-642-40994-3_50.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324.

H. Chabbar and M. Chami. Indoor localization using wi-fi method based on fingerprinting technique. In *2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, pages 1–5, 2017. doi: 10.1109/WITS.2017.7934613.

R.-C. Chen, C. Dewi, S.-W. Huang, and R. E. Caraka. Selecting critical features for data classification based on machine learning methods. *Journal of Big Data*, 7(1), July 2020. doi: 10.1186/s40537-020-00327-4.

Y.-K. Cheng and R. Y. Chang. Device-free indoor people counting using wi-fi channel state information for internet of things. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pages 1–6, 2017. doi: 10.1109/GLOCOM.2017.8254522.

A. I. Covaci. Wi-fi mac address randomization vs crowd monitoring, July 2022. URL `http://essay.utwente.nl/91744/`.

M. Cunche. I know your mac address: targeted tracking of individual using wi-fi. *Journal of Computer Virology and Hacking Techniques*, 10, 2014. doi: 10.1007/s11416-013-0196-1.

M. Cunche, M. A. Kaafar, and R. Boreli. I know who you will meet this evening! linking wireless devices using wi-fi probe requests. In *2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–9, 2012. doi: 10.1109/WoWMoM.2012.6263700.

A. Dagelić, T. Perković, and M. Čagalj. Location privacy and changes in wifi probe request based connection protocols usage through years. In *2019 4th International Conference on Smart and Sustainable Technologies (SpliTech)*, pages 1–5, 2019. doi: 10.23919/SpliTech.2019.8783167.

G. Delzanno, L. Caputo, D. D'Agostino, D. Grosso, A. Mustajab, L. Bixio, and M. Rulli. Automatic passenger counting on the edge via unsupervised clustering. *Sensors*, 23:5210, 05 2023. doi: 10.3390/s23115210.

J.-F. Determe, S. Azzagnuni, U. Singh, F. Horlin, and P. De Doncker. Monitoring large crowds with wifi: A privacy-preserving approach. *IEEE Systems Journal*, 16(2):2148–2159, 2022. doi: 10.1109/JSYST.2021.3139756.

A. Di Luzio, A. Mei, and J. Stefa. Mind your probes: De-anonymization of large crowds through smartphone wifi probe requests. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, 2016. doi: 10.1109/INFOCOM.2016.7524459.

M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996.

General Data Protection Regulation (GDPR). European Union, 2018. URL `https://gdpr.eu/article-4-definitions/`.

E. Fenske, D. Brown, J. Martin, T. Mayberry, P. Ryan, and E. Rye. Three years later: A study of MAC address randomization in mobile devices and when it succeeds. *Proceedings on Privacy Enhancing Technologies*, 2021(3):164–181, Apr. 2021. doi: 10.2478/popets-2021-0042.

E. Ferrara, M. Uras, and R. Cossu. Probe requests of 24 devices in a semianechoic chamber, 2020. URL `https://zenodo.org/record/3928500`.

Y. Furuya, H. ASAHINA, M. YOSHIDA, and I. SASASE. Indoor crowd estimation scheme using the number of wi-fi probe requests under mac address randomization. *IEICE Transactions on Information and Systems*, E104.D(9):1420–1426, 2021. doi: 10.1587/transinf.2020EDP7228.

A. Günter, S. Böker, M. König, and M. Hoffmann. Privacy-preserving people detection enabled by solid state lidar. In *2020 16th International Conference on Intelligent Environments (IE)*, pages 1–4, 2020. doi: 10.1109/IE49459.2020.9154970.

L. Hao, B. Huang, B. Jia, G. Xu, and G. Mao. Toward accurate crowd counting in large surveillance areas based on passive wifi sensing. *IEEE Transactions on Intelligent Transportation Systems*, 24(12):14086–14096, 2023. doi: 10.1109/TITS.2023.3303700.

S. He and S.-H. G. Chan. Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Communications Surveys & Tutorials*, 18(1):466–490, 2016. doi: 10.1109/COMST.2015.2464084.

M. Hesse, M. Mailand, H.-J. Jentschel, L. Deneire, and J. Lebrun. Semi-blind cancellation of iq-imbalances. In *2008 IEEE International Conference on Communications*, pages 5023–5027, 2008.

IEEE. IEEE standard for information technology–telecommunications and information exchange between systems - local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, pages 1–4379, 2021. doi: 10.1109/IEEESTD.2021.9363693.

N. Ilyas, A. Shahzad, and K. Kim. Convolutional-neural network-based image crowd counting: Review, categorization, analysis, and performance evaluation. *Sensors*, 20(1):43, 2019.

ISO/IEC JTC 1. Information technology — open systems interconnection basic reference model. *International Standard confirmed*, page 23, 1994. doi: https://www.iso.org/standard/18824.html.

A. Jalalvand, B. Vandersmissen, W. De Neve, and E. Mannens. Radar signal processing for human identification by means of reservoir computing networks. In *2019 IEEE Radar Conference (RadarConf)*, pages 1–6, 2019. doi: 10.1109/RADAR.2019.8835753.

A. Janecek, D. Valerio, K. A. Hummel, F. Ricciato, and H. Hlavacs. The cellular network as a sensor: From mobile phone data to real-time road traffic monitoring. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2551–2572, 2015. doi: 10.1109/TITS.2015.2413215.

T. Jian, B. C. Rendon, E. Ojuba, N. Soltani, Z. Wang, K. Sankhe, A. Gritsenko, J. Dy, K. Chowdhury, and S. Ioannidis. Deep learning for rf fingerprinting: A massive experimental study. *IEEE Internet of Things Magazine*, 3(1):50–57, 03 2020. ISSN 2576-3199. doi: 10.1109/IOTM.0001.1900065.

Z. Jianwu and Z. Lu. Research on distance measurement based on rssi of zigbee. In *2009 ISECS International Colloquium on Computing, Communication, Control, and Management*, volume 3, pages 210–212, 2009. doi: 10.1109/CCCM.2009.5267883.

A. Ketkhaw and S. Thipchaksurar. Hidden rogue access point detection technique for wireless local area networks. In *2017 21st International Computer Science and Engineering Conference (ICSEC)*, pages 1–5, 2017.

K. Li, C. Yuen, S. S. Kanhere, K. Hu, W. Zhang, F. Jiang, and X. Liu. An experimental study for tracking crowd in smart cities. *IEEE Systems Journal*, 13(3): 2966–2977, 2019. doi: 10.1109/JSYST.2018.2880028.

Y. Li, J. Barthelemy, S. Sun, P. Perez, and B. Moran. A case study of wifi sniffing performance evaluation. *IEEE Access*, 8:129224–129235, 2020. doi: 10.1109/ACCESS.2020.3008533.

U. C. London. Local data company - smartstreetsensor footfall data, 2019. URL `https://data.cdrc.ac.uk/dataset/local-data-company-smartstreetsensor-footfall-data`.

R. C. Luo and T.-J. Hsiao. Indoor localization system based on hybrid wi-fi/ble and hierarchical topological fingerprinting approach. *IEEE Transactions on Vehicular Technology*, 68(11):10791–10806, 2019. doi: 10.1109/TVT.2019.2938893.

J. Martin, T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E. C. Rye, and D. Brown. A study of mac address randomization in mobile devices and when it fails, 2017.

C. Matte, M. Cunche, F. Rousseau, and M. Vanhoef. Defeating mac address randomization through timing attacks. In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, WiSec '16, page 15–20, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342704. doi: 10.1145/2939918.2939930.

B. H. Menze, B. M. Kelm, R. Masuch, U. Himmelreich, P. Bachert, W. Petrich, and F. A. Hamprecht. A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC Bioinformatics*, 10(1):213, 2009. doi: 10.1186/1471-2105-10-213.

A. Mohammadian and C. Tellambura. Rf impairments in wireless transceivers: Phase noise, cfo, and iq imbalance – a survey. *IEEE Access*, 9:111718–111791, 2021. ISSN 2169-3536.

M. Mohorčič, A. Simončič, M. Mohorčič, and A. Hrovat. Dataset of ieee 802.11 probe requests from an uncontrolled urban environment, 2023.

A. Mukhopadhyay, P. S. Rajput, and S. Srirangarajan. A smartphone-based indoor localisation system using fm and wi-fi signals. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 2473–2477, 2017. doi: 10.23919/EUSIPCO.2017.8081655.

M. Nitti, F. Pinna, L. Pintor, V. Pilloni, and B. Barabino. iabacus: A wi-fi-based automatic bus passenger counting system. *Energies*, 13(6), 2020. doi: 10.3390/en13061446.

L. Oliveira, D. Schneider, J. De Souza, and W. Shen. Mobile device detection through wifi probe request analysis. *IEEE Access*, 7:98579–98588, 2019. doi: 10.1109/ACCESS.2019.2925406.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

L. Peng, J. Zhang, M. Liu, and A. Hu. Deep learning based rf fingerprint identification using differential constellation trace figure. *IEEE Transactions on Vehicular Technology*, 69(1):1091–1095, 1 2020. ISSN 1939-9359. doi: 10.1109/TVT.2019.2950670.

L. Pintor and L. Atzori. A dataset of labelled device wi-fi probe requests for mac address de-randomization - 2021, 2021.

L. Pintor and L. Atzori. A dataset of labelled device wi-fi probe requests for mac address de-randomization. *Computer Networks*, 205:108783, 2022a. doi: 10.1016/j.comnet.2022.108783.

L. Pintor and L. Atzori. Analysis of wi-fi probe requests towards information element fingerprinting. In *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pages 3857–3862, 2022b. doi: 10.1109/GLOBECOM48099.2022.10001618.

L. Pintor, M. Uras, G. Colistra, and L. Atzori. *Monitoring People's Mobility in the Cities: A Review of Advanced Technologies*, pages 25–42. Springer Nature Switzerland, Cham, 2024. ISBN 978-3-031-39446-1. doi: 10.1007/978-3-031-39446-1_3.

F. Potortì, A. Crivello, M. Girolami, E. Traficante, and P. Barsocchi. Wi-fi probes as digital crumbs for crowd localisation. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, 2016. doi: 10.1109/IPIN.2016.7743599.

U. M. Qureshi, Z. Umair, and G. P. Hancke. Indoor localization using wireless fidelity (wifi) and bluetooth low energy (ble) signals. In *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*, pages 2232–2237, 2019. doi: 10.1109/ISIE.2019.8781189.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

P. Robyns, B. Bonné, P. Quax, and W. Lamotte. Non-cooperative 802.11 mac layer fingerprinting and tracking of mobile devices, 2017. URL `https://zenodo.org/record/545970`.

A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure., 01 2007.

P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. ISSN 0377-0427. doi: 10.1016/0377-0427(87)90125-7. URL `https://www.sciencedirect.com/science/article/pii/0377042787901257`.

R. Rusca, F. Sansoldo, C. Casetti, and P. Giaccone. What wifi probe requests can tell you. In *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*, pages 1086–1091, 2023. doi: 10.1109/CCNC51644.2023.10060447.

K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury. Oracle: Optimized radio classification through convolutional neural networks. In

IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, pages 370–378, 2019. doi: 10.1109/INFOCOM.2019.8737463.

K. Sankhe, M. Belgiovine, F. Zhou, L. Angioloni, F. Restuccia, S. D'Oro, T. Melodia, S. Ioannidis, and K. Chowdhury. No radio left behind: Radio fingerprinting through deep learning of physical-layer hardware impairments. *IEEE Transactions on Cognitive Communications and Networking*, 6(1):165–178, 3 2020. ISSN 2332-7731. doi: 10.1109/TCCN.2019.2949308.

J. Shackleton, B. VanVoorst, and J. Hesch. Tracking people with a 360-degree lidar. In *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 420–426, 2010. doi: 10.1109/AVSS.2010.52.

L. Shu, Y. Chen, Z. Huo, N. Bergmann, and L. Wang. When mobile crowd sensing meets traditional industry. *IEEE Access*, 5:15300–15307, 2017. doi: 10.1109/ACCESS.2017.2657820.

A. Simončič, M. Mohorčič, M. Mohorcic, and A. Hrovat. Non-intrusive privacy-preserving approach for presence monitoring based on wifi probe requests. *Sensors*, 23:2588, 02 2023a. doi: 10.3390/s23052588.

A. Simončič, M. Mohorčič, M. Mohorčič, and A. Hrovat. Labeled dataset of ieee 802.11 probe requests, 2023b.

V. A. Sindagi and V. M. Patel. A survey of recent advances in cnn-based single image crowd counting and density estimation. *Pattern RecogniFtion Letters*, 107: 3–16, 2018.

U. Singh, J.-F. Determe, F. Horlin, and P. Doncker. Crowd monitoring: State-of-the-art and future directions. *IETE Technical Review*, 38, 08 2020. doi: 10.1080/02564602.2020.1803152.

M. Skolnik. Radar handbook. 2008.

W. C. Suski II, M. A. Temple, M. J. Mendenhall, and R. F. Mills. Using spectral fingerprints to improve wireless network security. In *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, pages 1–5, 2008. doi: 10.1109/GLOCOM.2008.ECP.421.

J. Tan and S.-H. Gary Chan. Efficient association of wi-fi probe requests under mac address randomization. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pages 1–10, 2021. doi: 10.1109/INFOCOM42981.2021.9488769.

C. Tang, T. Yan, and Y. An. Radio frequency fingerprint recognition based on deep learning. In *2021 International Conference on Intelligent Transportation,*

Big Data & Smart City (ICITBS), pages 708–711, 3 2021. doi: 10.1109/ICITBS 53129.2021.00177.

A. Thaljaoui, T. Val, N. Nasri, and D. Brulin. Ble localization using rssi measurements and iringla. In *2015 IEEE International Conference on Industrial Technology (ICIT)*, pages 2178–2183, 2015. doi: 10.1109/ICIT.2015.7125418.

H. Togashi, H. Furukawa, Y. Yamaguchi, R. Abe, and J. Shimamura. Network-based positioning and pedestrian flow measurement system utilizing densely placed wireless access points. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, 2016. doi: 10.1109/IPIN.2016.7743648.

T. Tonggoed and S. Panjan. Autonomous guided vehicles with wi-fi localization for smart factory. In *2022 7th International Conference on Robotics and Automation Engineering (ICRAE)*, pages 70–74, 2022. doi: 10.1109/ICRAE56463.2022.1005 6172.

T. Trasberg, B. Soundararaj, and J. Cheshire. Using wi-fi probe requests from mobile phones to quantify the impact of pedestrian flows on retail turnover. *Computers, Environment and Urban Systems*, 87:101601, 2021. ISSN 0198-9715. doi: 10.101 6/j.compenvurbsys.2021.101601.

M. W. Traunmueller, N. Johnson, A. Malik, and C. E. Kontokosta. Digital footprints: Using wifi probe and locational data to analyze human mobility trajectories in cities. *Computers, Environment and Urban Systems*, 72:4 – 12, 2018. doi: 10.1016/j.compenvurbsys.2018.07.006.

M. Uras, R. Cossu, E. Ferrara, O. Bagdasar, A. Liotta, and L. Atzori. Wifi probes sniffing: an artificial intelligence based approach for mac addresses derandomization. In *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 1–6, 2020a. doi: 10.1109/CAMAD50429.2020.9209257.

M. Uras, R. Cossu, E. Ferrara, A. Liotta, and L. Atzori. Pma: A real-world system for people mobility monitoring and analysis based on wi-fi probes. *Journal of Cleaner Production*, 270:122084, 2020b. ISSN 0959-6526. doi: 10.1016/j.jclepro. 2020.122084. URL `https://www.sciencedirect.com/science/article/pii/S0959652620321314`.

M. Uras, E. Ferrara, R. Cossu, A. Liotta, and L. Atzori. Mac address derandomization for wifi device counting: Combining temporal- and content-based fingerprints. *Computer Networks*, 218:109393, 2022. ISSN 1389-1286. doi: 10.1016/j.comnet.2022.109393.

M. V. Barbera, A. Epasto, A. Mei, S. Kosta, V. C. Perta, and J. Stefa. Crawdad dataset sapienza/probe-requests (v. 2013-09-10), 2013. URL `https://ieee-dataport.org/open-access/crawdad-sapienzaprobe-requests`.

M. Valkama, M. Renfors, and V. Koivunen. Advanced methods for i/q imbalance compensation in communication receivers. *IEEE Transactions on Signal Processing*, 49(10):2335–2344, 2001. ISSN 1941-0476. doi: 10.1109/78.950789.

M. Valles Coral, L. Salazar-Ramírez, R. Injante, E. Hernández Torres, J. juarez diaz, J. Navarro-Cabrera, L. Pinedo, and P. Vidaurre-Rojas. Density-based unsupervised learning algorithm to categorize college students into dropout risk levels. *Data*, 7:165, 11 2022. doi: 10.3390/data7110165.

M. Vanhoef, C. Matte, M. Cunche, L. Cardoso, and F. Piessens. Why mac address randomization is not enough: An analysis of wi-fi network discovery mechanisms. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, pages 413–424, 05 2016. doi: 10.1145/2897845.2897883.

E. Vattapparamban, B. S. Ciftler, I. Guvenc, K. Akkaya, and A. Kadri. Indoor occupancy tracking in smart buildings using passive sniffing of probe requests. In *2016 IEEE International Conference on Communications Workshops (ICC)*, pages 38–44, 2016. doi: 10.1109/ICCW.2016.7503761.

M. Vega-Barbas, M. Álvarez-Campana, D. Rivera, M. Sanz, and J. Berrocal. AFOROS: A low-cost wi-fi-based monitoring system for estimating occupancy of public spaces. *Sensors*, 21(11):3863, June 2021. doi: 10.3390/s21113863. URL `10.3390/s21113863`.

A. Venkataramanan, M. Laviale, C. Figus, P. Usseglio-Polatera, and C. Pradalier. Tackling inter-class similarity and intra-class variance for microscopic image-based classification. In M. Vincze, T. Patten, H. I. Christensen, L. Nalpantidis, and M. Liu, editors, *Computer Vision Systems*, pages 93–103, Cham, 2021. Springer International Publishing.

Z. Wang, H. Li, B. Nie, J. Du, Y. Du, and Y. Chen. Feature selection using different evaluate strategy and random forests. In *2021 International Conference on Computer Engineering and Artificial Intelligence (ICCEAI)*, pages 310–313, 2021. doi: 10.1109/ICCEAI52939.2021.00062.

F. Yang, I. Ahriz, and B. Denby. Tools for ground-truth-free passive client density mapping in mac-randomized outdoor wifi networks. *Sensors (Basel, Switzerland)*, 23, 07 2023. doi: 10.3390/s23136142.

H. Zhang, M. Zhou, H. Sun, G. Zhao, J. Qi, J. Wang, and H. Esmaiel. Que-fi: A wi-fi deep-learning-based queuing people counting. *IEEE Systems Journal*, 15(2): 2926–2937, 2021. doi: 10.1109/JSYST.2020.2994062.

P. Zhao, C. X. Lu, J. Wang, C. Chen, W. Wang, N. Trigoni, and A. Markham. mid: Tracking and identifying people with millimeter wave radar. In *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 33–40, 2019. doi: 10.1109/DCOSS.2019.00028.

J. Zhou, Y. Peng, G. Gui, Y. Lin, B. Adebisi, H. Gacanin, and H. Sari. A novel radio frequency fingerprint identification method using incremental learning. In *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*, pages 1–5, 9 2022. doi: 10.1109/VTC2022-Fall57202.2022.10012703.